Modélisation des métadonnées multi sources et hétérogènes pour le filtrage négatif et l'interrogation intelligente de grands volumes de données : application à la vidéosurveillance

Franck Jeveme Panta

▶ **To cite this version:**

Franck Jeveme Panta. Modélisation des métadonnées multi sources et hétérogènes pour le filtrage négatif et l'interrogation intelligente de grands volumes de données : application à la vidéosurveillance. Intelligence artificielle [cs.AI]. Université Paul Sabatier - Toulouse III, 2020. Français. NNT : 2020TOU30098 . tel-03118294

HAL Id: tel-03118294
https://theses.hal.science/tel-03118294

Submitted on 22 Jan 2021

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

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *07/10/2020* **par :**

Franck JEVEME PANTA

Modélisation des métadonnées multi sources et hétérogènes pour le filtrage négatif et l'interrogation intelligente de grands volumes de données : application à la vidéosurveillance

JURY

Elisabeth MURISASCO	Université de Toulon	Rapporteure
Thierry DELOT	Université Polytechnique Hauts-de-France	Rapporteur
Olivier TESTE	Université Toulouse 2 Jean Jaurès	Examinateur
Frédéric BOUCHARA	Université de Toulon	Examinateur
Florence SÈDES	Université Toulouse 3 Paul Sabatier	Directrice de thèse
André PÉNINOU	Université Toulouse 2 Jean Jaurès	Co-directeur de thèse

École doctorale et spécialité :
MITT : Domaine STIC : Intelligence Artificielle
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505 CNRS)
Directeur(s) de Thèse :
Florence SÈDES et *André PÉNINOU*
Rapporteurs :
Elisabeth MURISASCO et *Thierry DELOT*

Modélisation des métadonnées multi sources et hétérogènes pour le filtrage négatif et l'interrogation intelligente de grands volumes de données : application à la vidéosurveillance

Franck JEVEME PANTA

Octobre 2020

Remerciements

Je tiens à remercier très sincèrement Madame Florence Sèdes, Professeure à l'Université Toulouse 3 Paul Sabatier, pour sa constance dans sa bienveillance à mon égard, ainsi que pour ma famille : d'abord l'encadrement à distance de mon mémoire de Master (Cameroun) puis ces années de thèse passées sous sa direction. C'est grâce à son implication directe dans la concrétisation de mon contrat sur le projet national ANR FILTER2 qu'un financement a été possible. Je la remercie pour ses critiques (nourries. . .) et ses commentaires (incisifs) qui m'ont été précieux tout au long de ce travail. Outre son soutien scientifique et son côté humain, ses conseils m'ont permis d'avancer sereinement dans mes travaux, et ses actions d'avoir un financement permanent (ANR, ATER, . . .). Je tiens à lui exprimer toute ma gratitude et ma reconnaissance.

Je remercie mon co-directeur de thèse, Monsieur André Péninou, maître de conférences à l'université Toulouse 2 Jean Jaurès, pour son encadrement, sa disponibilité, son soutien continu, et ses compétences scientifiques qui ont fortement contribué à l'aboutissement de ce travail.

Madame le Pr. Elisabeth Murisasco, Monsieur le Pr. Thierry Delot, vous êtes souvent dans les références que Mme Sèdes fait à des travaux ou des équipes "proches". Que ces travaux soient une nouvelle occasion de relancer cette collaboration : la gestion des données est au centre de tout, dorénavant, avec son cortège de problèmes de sécurité, de mobilité, etc. Votre recul sur ce domaine fait de vous des références pour nous, et pour la "cause" de la modélisation des métadonnées, chère à Mme Sèdes.

Monsieur Frédéric Bouchara, vous avez accepté d'examiner mon travail : s'il est question de vidéo dans le titre, vous avez certainement été déçu de ne pas trouver matière à votre spécialité dans les développements, hormis quelques critères de qualité à intégrer à nos requêtes de filtrage négatif. . . . J'espère vous avoir un peu accompagné sur le chemin vers cette "cause" de la modélisation des métadonnées.

Je vous remercie tou.tes pour m'avoir fait l'honneur d'être rapporteurs et examinateur de cette thèse. Je vous suis reconnaissant d'avoir accepté cette lourde tâche dans ces circonstances si particulières du confinement.

Je remercie Monsieur Olivier Teste, professeur à l'université Toulouse 2 Jean Jaurès et responsable de l'équipe Systèmes d'Informations Généralisées (SIG) de l'IRIT, pour sa présidence du jury de cette thèse. Je saisis cette occasion pour le remercier de m'avoir accepté au sein de son équipe.

Je remercie également tous les membres de l'équipe SIG qui m'ont accueilli pendant ces années de thèse, ainsi que tout le personnel du laboratoire pour leur gentillesse et

les services et solutions trouvées dans des conditions précaires.

J'adresse un remerciement particulier à Monsieur Jean-François Sulzer, émérite ingénieur Thalès Security, désormais consultant indépendant dans le domaine de la vidéosurveillance, pour sa collaboration précieuse, et lui manifeste mon intérêt pour de futures perspectives de collaboration. Il a été converti il y a de ça quelques années et deux ou trois projets européens déjà par Mme Sèdes... et continue à œuvrer auprès de la police en ce sens.

A mes amis et collègues Mahmoud Qodseya, Wafa Abdelghani, Geoffrey Roman Jimenez et Mahdi Washha, nous avons passé de très bons moments ensemble (collaborations, pauses café, balades, ...). Je vous remercie pour tout. Mais tout cela ne fait que commencer....

Je tiens à remercier toute ma famille pour m'avoir soutenu et avoir cru en moi pendant de nombreuses années d'études. Je suis particulièrement reconnaissant envers ma mère désormais grand-mère pour son amour inconditionnel et pour son soutien en toutes circonstances, envers mes frères et sœurs pour leur affection indéfectible. A Lydia, tu partages ma vie, merci pour tout ton soutien et tes encouragements. Enfin, je remercie ma fille Chloé, née il y a 5 mois, pour la joie et le bonheur qu'elle me procure chaque jour.

Résumé

En raison du déploiement massif et progressif des systèmes de vidéosurveillance dans les grandes métropoles, l'analyse a posteriori des vidéos issues de ces systèmes est confrontée à de nombreux problèmes parmi lesquels : (i) l'interopérabilité, due aux différents formats de données (vidéos) et aux spécifications des caméras propres à chaque système ; (ii) le grand temps d'analyse lié à l'énorme quantité de données et métadonnées générées ; et (iii) la difficulté à interpréter les vidéos qui sont parfois à caractère incomplet. Face à ces problèmes, la nécessité de proposer un format commun d'échange des données et métadonnées de vidéosurveillance, de rendre le filtrage et l'interrogation des contenus vidéo plus efficaces, et de faciliter l'interprétation des contenus grâce aux informations exogènes (contextuelles) est une préoccupation incontournable.

De ce fait, cette thèse se focalise sur la modélisation des métadonnées multi sources et hétérogènes afin de proposer un filtrage négatif et une interrogation intelligente des données, applicables aux systèmes de vidéosurveillance en particulier et adaptables aux systèmes traitant de grands volumes de données en général. L'objectif dans le cadre applicatif de cette thèse est de fournir aux opérateurs humains de vidéosurveillance des outils pour les aider à réduire le grand volume de vidéo à traiter ou à visionner et implicitement le temps de recherche.

Nous proposons donc dans un premier temps une méthode de filtrage dit "négatif", qui permet d'éliminer parmi la masse de vidéos disponibles celles dont on sait au préalable en se basant sur un ensemble de critères, que le traitement n'aboutira à aucun résultat. Les critères utilisés pour l'approche de filtrage négatif proposé sont basés sur une modélisation des métadonnées décrivant la qualité et l'utilisabilité/utilité des vidéos.

Ensuite, nous proposons un processus d'enrichissement contextuel basé sur les métadonnées issues du contexte, et permettant une interrogation intelligente des vidéos. Le processus d'enrichissement contextuel proposé est soutenu par un modèle de métadonnées extensible qui intègre des informations contextuelles de sources variées, et un mécanisme de requêtage multiniveaux avec une capacité de raisonnement spatio-temporel robuste aux requêtes floues.

Enfin, nous proposons une modélisation générique des métadonnées de vidéosurveillance intégrant les métadonnées décrivant le mouvement et le champ de vue des caméras, les métadonnées issues des algorithmes d'analyse des contenus, et les métadonnées issues des informations contextuelles, afin de compléter le dictionnaire des

métadonnées de la norme ISO 22311/IEC 79 qui vise à fournir un format commun d'export des données extraites des systèmes de vidéosurveillance.

Les expérimentations menées à partir du framework développé dans cette thèse ont permis de démontrer la faisabilité de notre approche dans un cas réel et de valider nos propositions.

MOTS-CLÉS : métadonnées, informations contextuelles, données ouvertes, interopérabilité, systèmes de vidéosurveillance.

Abstract

Due to the massive and progressive deployment of video surveillance systems in major cities, a posteriori analysis of videos coming from these systems is facing many problems, including the following : (i) interoperability, due to the different data (video) formats and camera specifications associated to each system ; (ii) time-consuming nature of analysis due to the huge amount of data and metadata generated ; and (iii) difficulty to interpret videos which are sometimes incomplete. To address these issues, the need to propose a common format to exchange video surveillance data and metadata, to make video content filtering and querying more efficient, and to facilitate the interpretation of content using external (contextual) information is an unavoidable concern.

Therefore, this thesis focuses on heterogeneous and multi-source metadata modeling in order to propose negative filtering and intelligent data querying, which are applicable to video surveillance systems in particular and adaptable to systems dealing with large volumes of data in general. In the applicative context of this thesis, the goal is to provide human CCTV operators with tools that help them to reduce the large volume of video to be processed or viewed and implicitly reduce search time.

We therefore initially propose a so-called "negative" filtering method, which enables the elimination from the mass of available videos those that it is know in advance, based on a set of criteria, that the processing will not lead to any result. The criteria used for the proposed negative filtering approach are based on metadata modeling describing video quality and usability/usefulness.

Then, we propose a contextual enrichment process based on metadata from the context, enabling intelligent querying of the videos. The proposed contextual enrichment process is supported by a scalable metadata model that integrates contextual information from a variety of sources, and a multi-level query mechanism with a spatio-temporal reasoning ability that is robust to fuzzy queries.

Finally, we propose a generic metadata modeling of video surveillance metadata integrating metadata describing the movement and field of view of cameras, metadata from content analysis algorithms, and metadata from contextual information, in order to complete the metadata dictionary of the ISO 22311/IEC 79 standard, which aims to provide a common format to export data extracted from video surveillance systems. The experiments performed using the framework developed in this thesis showed the reliability of our approach in a real case and enabled the validation of our proposals.

KEYWORDS : metadata, contextual information, open data, interoperability, video surveillance systems.

Publications

Articles de revues internationales avec comité de lecture

1. **Franck Jeveme Panta**, Jean-François Sulzer, Florence Sèdes. Forensics Usage of Video-Surveillance and Associated (Meta) Data. *Global Journal of Forensic Science & Medicine (GJFSM.MS.ID.000504), Volume 1, issue 1*, 2018.
2. Florence Sèdes, **Franck Jeveme Panta**. (Meta-)Data Modelling : Gathering Spatio-Temporal Data for Indoor-Outdoor Queries. *ACM SIGSPATIAL (rank A), Special issue, volume 9, issue 1*, pages 42-49, 2017, doi : 10.1145/3124104. 3124111.

Conférences internationales avec actes et comité de lecture

1. **Franck Jeveme Panta**, André Péninou, Florence Sèdes : Negative Filtering of CCTV Content - Forensic Video Analysis Framework. *15th International Conference on Availability, Reliability and Security, ARES2020 (rank B)*, pages 1-10, 2020.
2. **Franck Jeveme Panta**, André Péninou, Florence Sèdes : An Approach for CCTV Contents Filtering Based on Contextual Enrichment via Spatial and Temporal Metadata : Relevant Video Segments Recommended for CCTV Operators. *17th International Conference on Advances in Mobile Computing and Multimedia, MoMM2019 (rank B)*, pages 195-199, 2019.
3. Mahmoud Qodseya, **Franck Jeveme Panta**, and Florence Sèdes. Visual-based eye contact detection in multi-person interactions. *In 2019 International Conference on Content-Based Multimedia Indexing, CBMI2019*, pages1–6, 2019.
4. **Franck Jeveme Panta**, Mahmoud Qodseya, André Péninou, Florence Sèdes. Management of Mobile Objects Location for Video Content Filtering. *16th International Conference on Advances in Mobile Computing and Multimedia, MoMM 2018 (rank B)*, pages 44-52, 2018, doi : 10.1145/ 3282353.3282368.
5. **Franck Jeveme Panta**, Geoffrey Roman-Jimenez, Florence Sèdes. Modeling metadata of CCTV systems and Indoor Location Sensors for automatic filtering

of relevant video content. *12th International Conference on Research Challenges in Information Science, RCIS 2018 (rank B)*, pages 1-9, 2018, doi : 10.1109/RCIS.2018.8406677.

6. **Franck Jeveme Panta**, Mahmoud Qodseya, Geoffrey Roman-Jimenez, A. Péninou, Florence Sèdes. Spatio-Temporal Metadata Querying for CCTV Video Retrieval : Application in Forensic. *9th ACM SIGSPATIAL (rank A) International Workshop on Indoor Spatial Awareness (ISA)*, pages 7-14, 2018.
7. **Franck Jeveme Panta**, Florence Sèdes. Querying indoor spatio-temporal data by hybrid trajectories. *8th ACM SIGSPATIAL (rank A) International Workshop on Indoor Spatial Awareness (ISA)*, pages 11-18, 2016.
8. **Franck Jeveme Panta**, Florence Sèdes. Mobile objects indoor environment : trajectories reconstruction. *14th International Conference on Advances in Mobile Computing and Multimedia, MoMM 2016 (rank B)*, pages 332-336, 2016.

Workshops internationaux avec actes et comité de lecture

1. **Franck Jeveme Panta**, Mahmoud Qodseya, Geoffrey Roman-Jimenez, A. Péninou, Florence Sèdes. Spatio-Temporal Metadata Querying for CCTV Video Retrieval : Application in Forensic. *9th ACM SIGSPATIAL (rank A) International Workshop on Indoor Spatial Awareness (ISA)*, pages 7-14, 2018.
2. **Franck Jeveme Panta**, Florence Sèdes. Querying indoor spatio-temporal data by hybrid trajectories. *8th ACM SIGSPATIAL (rank A) International Workshop on Indoor Spatial Awareness (ISA)*, pages 11-18, 2016.

Conférences nationales avec actes et comité de lecture

1. **Franck Jeveme Panta**, André Péninou, Florence Sèdes. Modélisation des (méta) données hétérogènes et filtrage des contenus de vidéosurveillance : application au Forensic. *Congrès INFormatique des ORganisations et Systèmes d'Information et de Décision, INFORSID 2018*, pages 1-4, 2018.
2. **Franck Jeveme Panta**, Florence Sèdes. Interrogation des données spatio - temporelles de géolocalisation indoor à partir des trajectoires hybrides. *Conférence internationale francophone Spatial Analytics and GEOmatics, SAGEO2016*, pages 1-22, 2016.

Table des matières

1 Introduction **1**

1.1 Contexte 1
1.2 Problématique 4
1.3 Objectifs 4
1.4 Contributions 5
1.5 Plan du manuscrit 5

2 Etat de l'art **7**

2.1 Systèmes de vidéosurveillance 7
2.1.1 Approches basées sur l'analyse des contenus vidéo 10
2.1.2 Récents projets et travaux dans le domaine de la vidéosurveillance 12
2.2 Informations contextuelles 15
2.2.1 Définition du contexte 15
2.2.2 Modélisation des informations contextuelles 17
2.2.2.1 Modélisation clé-valeur 17
2.2.2.2 Modélisation basée sur le balisage 18
2.2.2.3 Modélisation graphique 18
2.2.2.4 Modélisation orientée objets 19
2.2.2.5 Modélisation logique 19
2.2.2.6 Modélisation basée sur l'ontologie 20
2.3 Intégration des informations contextuelles dans le filtrage des données . 22
2.4 Interrogation basée sur la logique floue 25
2.4.1 Définition de la logique floue 25
2.4.2 Requêtage flou ("fuzzy querying") 27
2.5 Conclusion 29

3 Filtrage négatif via l'exploitation des métadonnées **31**

3.1 Définition du filtrage négatif 32
3.2 Contexte d'application 33
3.3 Modélisation des métadonnées pour le filtrage négatif 35
3.3.1 Métadonnées liées à la qualité de la vidéo 37
3.3.2 Métadonnées liées à l'utilisabilité/utilité de la vidéo 38
3.3.3 Proposition d'un modèle de métadonnées pour la qualité et d'utilisabilité/utilité des vidéos 39

3.4 Mécanisme de filtrage . . . 40
3.4.1 Définition des données . . . 41
3.4.2 Algorithmes de filtrage . . . 41
3.4.2.1 Algorithme de filtrage pour le mode urgent . . . 42
3.4.2.2 Algorithme de filtrage pour le mode approfondi . . . 42
3.4.3 Exemple de filtrage . . . 45
3.4.3.1 Exemple d'analyse urgente . . . 45
3.4.3.2 Exemple d'analyse approfondie . . . 47
3.5 Conclusion . . . 48

4 Enrichissement contextuel 49
4.1 Définition de l'enrichissement contextuel . . . 49
4.2 Étapes génériques pour la mise en œuvre de l'enrichissement contextuel 51
4.3 Enrichissement contextuel : application aux systèmes de vidéosurveillance 52
4.3.1 Analyse des informations contextuelles utiles pour la vidéosurveillance . . . 53
4.3.1.1 Données ouvertes ou Open Data . . . 53
4.3.1.2 Médias sociaux . . . 55
4.3.1.3 Mobilité et géolocalisation . . . 56
4.3.2 Modélisation des informations contextuelles . . . 57
4.3.2.1 Modélisation des métadonnées descriptives . . . 57
4.3.2.2 Modélisation des métadonnées sémantiques . . . 59
4.3.2.3 Modélisation des métadonnées issues de l'open data . . 60
4.3.2.4 Modélisation des métadonnées issues des médias sociaux 61
4.3.2.5 Modélisation des métadonnées issues de la mobilité et la géolocalisation . . . 62
4.3.2.6 Modèle générique de métadonnées . . . 62
4.3.3 Représentation temporelle des évènements dynamiques . . . 63
4.3.4 Mécanisme de requêtage . . . 68
4.4 Conclusion . . . 77

5 Contribution à la norme ISO 22311/IEC 79 79
5.1 Norme ISO 22311/IEC 79 . . . 79
5.2 Proposition d'un modèle générique de métadonnées de vidéosurveillance selon à la norme . . . 81
5.3 Conclusion . . . 84

6 Application 85
6.1 Architecture du framework proposé . . . 85
6.1.1 Module de collecte de métadonnées . . . 85

6.1.2 Module interface utilisateur 87
6.1.3 Module gestion et traitement des métadonnées 88
6.2 Expérimentations et résultats 94
6.2.1 Présentation du dataset 94
6.2.2 Expérience 1 - Filtrage négatif 96
6.2.2.1 Mise en place de l'expérience 96
6.2.2.2 Paramètres de l'expérience 97
6.2.2.3 Résultats et interprétations 99
6.2.2.4 Évaluation 104
6.2.3 Expérience 2 - Enrichissement contextuel 107
6.2.3.1 Mise en place de l'expérience 108
6.2.3.2 Paramètres de l'expérience 109
6.2.3.3 Résultats et interprétations 110
6.2.3.4 Évaluation 116
6.3 Conclusion 118

7 Conclusion générale et perspectives **119**
7.1 Bilan 119
7.2 Perspectives 121

A Bases de données spatiales **123**

Bibliographie **124**

Table des figures

1.1 Exemples de dégradations d'images liées à l'environnement. 3

2.1 Schéma d'un système de vidéosurveillance automatique. 8
2.2 Différentes catégories de contexte et exemples. 15
2.3 Différence entre la logique booléenne et la logique floue 27

3.1 Filtrage de données. 31
3.2 Filtrage négatif. 33
3.3 Paramètres qui influencent la prise de vue et la qualité de l'image (source : FILTER2). 35
3.4 Exemples de scores de qualité d'image obtenus grâce aux métriques sans référence G-BLINDS2 et BIQI. 38
3.5 Modèle générique pour les métadonnées de qualité et d'utilisabilité/utilité des vidéos. 39
3.6 Formalisme possible pour les résultats du filtrage négatif. 40
3.7 Exemple de vidéo en entrée pour le filtrage négatif. 45
3.8 Exemples de seuils de compatibilité aux différents traitements. 45
3.9 Exemple de filtrage négatif en mode urgent. 47
3.10 Exemple de filtrage négatif en mode approfondi. 48

4.1 Étapes génériques pour la mise en œuvre de l'enrichissement contextuel. 52
4.2 Extrait des données ouvertes de DATA Touluse Métropole. 54
4.3 Sources d'informations contextuelles. 58
4.4 Champs de vision d'une caméra. 58
4.5 Métadonnées descriptives de vidéosurveillance. 59
4.6 Métadonnées sémantiques de vidéosurveillance. 60
4.7 Métadonnées issues de données ouvertes. 61
4.8 Métadonnées issues des médias sociaux. 62
4.9 Métadonnées issues de la mobilité et la géolocalisation. 63
4.10 Modèle générique des métadonnées de contexte de la vidéosurveillance. 64
4.11 Exemple de représentation classique d'un évènement dynamique. 66
4.12 Conversion du temps classique en temps relatif. 67
4.13 Requêtage multi-niveaux. 69
4.14 Exemple de représentation floue de la "visibilité". 71
4.15 Processus de calcul du degré de "visibilité". 72

4.16 Croisement temporel et segmentation de la vidéo. 73
4.17 Croisement temporel. 74
4.18 Exemple d'interrogation multi-niveaux. 75
4.19 Mécanisme de requêtage. 77

5.1 Métadonnées liées au capteur. 81
5.2 Métadonnées liées à l'évènement. 81
5.3 Différentes sources de métadonnées. 82
5.4 Modèle générique des métadonnées. 83

6.1 Architecture du framework proposé. 86
6.2 Exemple d'une interface de construction de requête. 87
6.3 Visualisation des extraits vidéo. 88
6.4 Template de requête JSON pour le filtrage négatif. 90
6.5 Exemple de requête JSON pour le filtrage négatif. 91
6.6 Template de requête JSON pour l'enrichissement contextuel. 92
6.7 Exemple de requête JSON pour l'enrichissement contextuel. 93
6.8 Bâtiment principal qui regroupe 17 caméras dont les champs de vision se chevauchent [Malon et al., 2018]. 95
6.9 Positions des caméras disposées sur le campus de l'Université Paul Sabatier. Ces 8 caméras ont des vues disjointes [Malon et al., 2018]. La zone rouge correspond à la Figure 6.8 95
6.10 Dégradation des images 98
6.11 Extrait de la base de données des métadonnées de qualité. 99
6.12 Seuils de compatibilité définis pour l'expérience. 99
6.13 Requête JSON pour le filtrage négatif. 100
6.14 Détection de visage pour la vidéo V_1. 101
6.15 Détection de véhicule pour la vidéo V_1. 102
6.16 Détection et lecture automatique des plaques pour la vidéo V_1. 103
6.17 Durée de vidéo à exploiter après filtrage pour chaque traitement dans le mode urgent. 104
6.18 Durée de vidéo à exploiter (meilleur des cas) après filtrage pour chaque traitement dans le mode approfondi. 105
6.19 Durée de vidéo à exploiter (pire des cas) après filtrage pour chaque traitement dans le mode approfondi. 105
6.20 Gain de temps pour le traitement détection de visage. 106
6.21 Gain de temps pour le traitement détection de véhicule. 107
6.22 Gain de temps pour le traitement détection détection et lecture automatique des plaques. 107
6.23 Requêtage à 3 niveaux. 108

6.24 Données synthétiques de météo. 109
6.25 Fonction d'appartenance pour la visibilité des vidéos. 110
6.26 Requête JSON pour l'enrichissement contextuel. 111
6.27 Exemples de frames sélectionnées au premier niveau de requêtage. . . . 112
6.28 Exemples de frames éliminées au premier niveau de requêtage. 112
6.29 Nombre de frame à visionner après le premier niveau de requâtage. . . 112
6.30 Exemples de frames contenant de véhicules presque en permanence mais pas toujours de mouvement. 113
6.31 Exemples de frames sélectionnées au deuxième niveau de requêtage. . . 114
6.32 Nombre de frame à visionner après le deuxième niveau de requâtage. . 114
6.33 Conversion temps classique - temps relatif des intensités maximales de pluie. 115
6.34 Comparaison des résultats de l'enrichissement contextuel à la vérité terrain. 117

Liste des tableaux

2.1 Comparaison des techniques de modélisation des informations contextuelles. 20

3.1 Valeurs globales des descripteurs pour chaque traitement. 46

4.1 Intensités maximales de pluie exprimées en temps classique. 65
4.2 Intensités maximales de pluie exprimées en temps relatif. 67
4.3 Exemple de données météo (intensités maximales de pluie). 70
4.4 Conversion des données météo en temps relatif. 71
4.5 Résultats du calcul des degrés de visibilité. 72
4.6 Résultat de la requête décrite à l'exemple 1. 73
4.7 Exemple de métadonnées décrivant la présence des objets dans chaque frame. 73
4.8 Résultat de la requête décrite à l'exemple 2. 74

6.1 Exemple de résultats au premier niveau de requêtage. 111
6.2 Exemple de résultats au deuxième niveau de requêtage. 113
6.3 Degrés de visibilité associés aux intensités maximales de pluie. 115
6.4 Degré de visibilité de chaque segment de la vidéo *V1*. 116
6.5 Exemple de résultats au troisième niveau de requêtage. 116

A.1 Comparaison des SGBD spatiales. 123

Chapitre 1

Introduction

1.1 Contexte

Les caméras de surveillance destinées à la sécurité des citoyens sont désormais largement répandues dans la société, tant dans les espaces publics que privés. Ces caméras diversifiées (objectif fixe, focale variable) et appartenant à différents systèmes de vidéosurveillance (ex : RATP, SNCF), génèrent une quantité importante de vidéos car elles filment en continu (24 heures sur 24). En parallèle, les systèmes de vidéosurveillance sont en constante évolution grâce au progrès des algorithmes d'analyse des contenus vidéo. Cette évolution est particulièrement observable dans le cadre des applications temps réel qui permettent l'identification et la reconnaissance des objets, la détection des situations anormales et le déclenchement d'alarmes, l'analyse des mouvements et des comportements en temps réel. Un exemple d'application très récent est celui de la vidéosurveillance pour lutter contre les piétons imprudents en Chine (Pékin), où des dispositifs faisant appel aux caméras de surveillance sont installés et filment les piétons qui traversent quand les feux tricolores sont rouges, ensuite, transmettent sur un écran géant leurs images et les informations telles que leurs noms, obtenues grâce à un "matching" entre la reconnaissance faciale (métadonnées générées) et les fichiers de la police. Les métadonnées désignent un terme très générique qui définit les données de support, complémentaires aux *"données d'intérêt"* et qui sont fournies pour aider à l'interprétation ou à l'exploitation des *"données d'intérêt"*.

Contrairement aux applications temps réel, les données (vidéos et métadonnées) générées par les systèmes de vidéosurveillance peuvent aussi être exploitées a posteriori, c'est le cas étudié dans cette thèse. Il s'agit de traiter de grands volumes de vidéos provenant de sources diverses qui sont le plus souvent hétérogènes (réseaux de vidéosurveillance urbains, réseaux privés, réseaux mobiles, etc.). Par conséquent, le filtrage, l'interrogation, l'interprétation des vidéos et l'interopérabilité des systèmes sont des préoccupations majeures dans diverses applications. L'utilisation des vidéos dans les enquêtes est un cas d'application qui montre l'utilité du filtrage et/ou de l'interrogation des grands volumes de vidéo. Dans le cadre des investigations, les enquêteurs font souvent appel aux vidéos stockées sur des serveurs afin d'en tirer des éléments de preuve. Un exemple récent est celui de l'enquête suite à l'attentat (attaque à la bombe)

survenu en France à Lyon le 21 mai 2019, enquête dans laquelle les images prises par les caméras de vidéosurveillance de la ville ont permis d'identifier et de suivre les traces du suspect.

Les affaires récentes (terrorisme, enlèvement, homicide) ont nécessité l'analyse de plusieurs dizaines de milliers d'heures de vidéo. Le gain de temps fourni par les outils d'analyse actuels reste insuffisant dans un contexte opérationnel. Beaucoup de vidéos analysées pour une enquête s'avèrent inexploitables du fait par exemple de leur mauvaise qualité ou de leur caractère inadapté au contexte de l'enquête. Écarter ces séquences inéligibles ce que nous appelons filtrage « négatif » peut permettre d'optimiser le temps d'exploitation en n'exploitant que les vidéos qui sont réellement exploitables. A titre d'exemple, les problèmes liés à la luminosité de la scène, au calibrage ou à l'optique de la caméra peuvent entraîner des dégradations de la qualité d'image. Une image dégradée peut avoir une influence négative sur son interprétation par l'œil humain ou par les algorithmes de traitement automatique. Du coup, visionner ou traiter de telles séquences s'avère une perte de temps et de ressource. Dans ce cas, le filtrage négatif permettra de réduire le temps d'exploitation des vidéos, soit en écartant les séquences vidéo qui contiennent les images inexploitables, soit en proposant des séquences vidéo avec des pourcentages (niveaux de confiance) d'exploitabilité tout en laissant le choix final d'exploitation à l'opérateur de vidéosurveillance.

D'autres problèmes liés à l'environnement tels que la présence des particules atmosphériques (pluie, brouillard, nuage, poussière, pollution, etc.) peuvent altérer la qualité des images, entraîner des dégradations ou empêcher la visibilité. La Figure 1.1 présente des exemples de dégradations de la qualité d'image liées aux phénomènes environnementaux : l'image (a) est dégradée par la pluie, l'image (b) est dégradée par le brouillard et l'image (c) est dégradée par la pollution. Afin de rechercher et d'éliminer ces séquences vidéo inexploitables (filtrage) ou de les annoter comme optionnel pour l'exploitation (ranking), il est possible de recourir aux informations exogènes (contextuelles) telles que les données ouvertes (Open Data) qui mettent à disposition de nombreuses informations ou métadonnées comme celles de la météo (ex : intensité de brouillard, de pollution, et quantité de précipitation enregistrée périodiquement dans différentes localités). Un tel filtrage ou "ranking" consiste à effectuer un rapprochement spatial et temporel entre les vidéos et les données ouvertes, d'où la nécessité d'un raisonnement spatio-temporel (modélisation spatio-temporelle et la définition d'opérateurs spatiaux et temporels).

Les informations exogènes constituent également une source d'information complémentaire pouvant pallier à certaines limites inhérentes aux données de la vidéosurveillance telles que leur caractère incomplet, ou aider à l'interprétation et la compréhension de ces données. Par exemple les réseaux sociaux comme Twitter et Facebook constituent des sources d'information capitales pour les forces de l'ordre, la presse

(a) (b) (c)

FIGURE 1.1 – Exemples de dégradations d'images liées à l'environnement.

et les internautes concernés par un même signalement, notamment en cas d'incident, qu'il soit anodin ou grave. Une fois les autorités de la ville alertées d'un évènement au travers des mots-clés qui se dégagent des réseaux sociaux, la vidéosurveillance peut permettre de recueillir des images, de localiser le problème, de suivre la situation et d'évaluer son ampleur. En recoupant le contenu publié sur les réseaux sociaux avec les données recueillies par les caméras de vidéosurveillance, les capteurs intelligents et d'autres applications, les autorités peuvent mieux cerner les situations au quotidien et les réponses à y apporter. Recouper toutes ces informations provenant de sources différentes (réseaux sociaux, vidéosurveillance, capteurs, etc.) pose un problème de modélisation spatio-temporelle à résoudre.

De plus, les caméras sont installées dans des contextes différents (indoor, outdoor, fixe, mobile), fabriquées et gérées par des organismes indépendants qui ont chacun leurs spécifications, formats de vidéo et métadonnées, ce qui rend très difficile l'interopérabilité des différents systèmes de vidéosurveillance auxquels elles appartiennent. Par exemple, si un incident se produit dans un centre commercial, il y a une forte probabilité que les enquêteurs fassent appel aux vidéos provenant des différents systèmes de vidéosurveillance présents dans le centre commercial (magasins) et aux alentours (station de métro, distributeurs de billets, station essence, etc.). Or, les différents systèmes ne sont pas toujours conçus pour interopérer. Afin d'adresser le besoin d'interopérabilité des systèmes de vidéosurveillance, les principaux acteurs de ce domaine en France ont développé la norme ISO 22311/IEC 79 qui propose un format d'export (structure et dictionnaire) des données et métadonnées issues des systèmes de vidéosurveillance. Cette norme propose des dictionnaires de métadonnées concernant les capteurs (caméras) et la scène observée. Dans son état actuel, les dictionnaires de métadonnées de la norme ne couvrent pas toutes les spécifications d'où la nécessité de les enrichir afin de rendre la norme plus complète en prenant en compte les métadonnées issues des algorithmes automatiques de traitement d'images et celles issues du contexte.

Cette thèse se situe dans un contexte général d'interrogation des grands volumes de données avec une application aux systèmes de vidéosurveillance, qui sont des systèmes

ayant les particularités suivantes : (i) la diversité des contextes d'acquisition : type de caméra (ex : PTZ), installation des caméras (fixe, mobile, indoor, outdoor) ; (ii) le grand volume de vidéos générées entrainant un très grand temps d'exploitation (recherche et visionnage) ; (iii) le grand nombre de systèmes gérés par des entités différentes ayant leur propre format de données et métadonnées ce qui entraîne un manque d'interopérabilité.

1.2 Problématique

Dans un contexte d'application comme celui défini à la section précédente, la question qui se pose est : quels outils fournir aux opérateurs de vidéosurveillance pour les aider à réduire le grand volume de vidéo à visionner et implicitement le temps de recherche, tout en résolvant les problèmes d'interopérabilité liés aux systèmes de vidéosurveillance ? Nous abordons ce problème dans une approche de gestion de données : interrogation des données (vidéos dans notre cas) grâce à la modélisation et l'intégration des métadonnées associées permettant leur exploitation. Nous ne faisons pas d'analyse de contenu vidéo, ni de traitement d'image, mais nous proposons une *modélisation des métadonnées* caractérisant les vidéos et une *interrogation intelligente de ces métadonnées* afin de filtrer les vidéos et proposer aux opérateurs des séquences vidéo qui satisfont certains prérequis (ex : qualité).

1.3 Objectifs

L'objectif principal de ce travail est de proposer une approche permettant de faciliter l'interrogation de grands volumes de données en s'appuyant sur les métadonnées collectées et/ou extraites des données traitées par un système et des données exogènes ou de contexte. Il s'agit dans le cas des systèmes de vidéosurveillance traités dans cette thèse, de :

— résoudre les problèmes d'interopérabilité des données issues de différents systèmes de vidéosurveillance ;

— proposer une modélisation des métadonnées pour la mise en œuvre du filtrage négatif qui consiste à éliminer parmi les données existantes celles qui ne sont pas exploitables selon un ensemble de métriques définies ;

— lever les verrous relatifs à l'intégration et à la collaboration des différents niveaux de description de métadonnées utiles qui peuvent provenir des informations de contexte ;

— effectuer le requêtage de grands volumes de données à partir des informations contextuelles et des métadonnées.

1.4 Contributions

Afin de répondre à la problématique posée dans le cadre de ce travail, notre contribution s'articule autour des propositions suivantes :

— **La modélisation des (méta)données de vidéosurveillance** : cette proposition est propre aux systèmes de vidéosurveillance. Elle consiste à élaborer un format de métadonnées conforme à la norme ISO 22331/IEC 79 (qui a comme objectif la facilitation de l'interopérabilité des systèmes de vidéosurveillance), encapsulant les métadonnées liées aux capteurs et les métadonnées issues des algorithmes d'analyse des contenus vidéo.

— **Contribution à la norme ISO 22331/IEC 79** : elle a pour but d'enrichir les dictionnaires de la norme et d'en extraire des spécifications pour des outils logiciels qui pourraient venir comme complément de la norme (ex : des outils d'assistance à la recherche d'informations dans les vidéos).

— **Le filtrage négatif basé sur les métadonnées** : cette proposition est adaptable au filtrage de grands volumes de données issues de différents systèmes ou domaines. Elle a pour but de réduire le volume de données à exploiter, afin de réduire ainsi le temps d'analyse. La définition des métriques pour le filtrage négatif est basée sur la modélisation des métadonnées utiles pour le domaine d'application (vidéosurveillance, imagerie médicale, télédétection, etc.) et la définition de seuils paramétrables.

— **L'enrichissement contextuel** : cette proposition consiste à intégrer différentes sources d'informations contextuelles telles que les médias sociaux, la mobilité, l'open data, la géolocalisation, le crowdsourcing, etc. afin d'enrichir et d'améliorer le processus d'interrogation des données et/ou métadonnées.

— **Un mécanisme de requêtage basé sur les métadonnées** : cette proposition a pour but de réduire la dimension de recherche. Elle consiste à mettre en œuvre un mécanisme de requêtage floue robuste aux informations incomplètes ou manquantes à partir des métadonnées disponibles.

1.5 Plan du manuscrit

La suite de manuscrit est structurée en 6 chapitres. Les contenus de ces chapitres sont résumés comme suit :

Le **Chapitre 2** commence par introduire des concepts importants qui seront utilisés dans la suite du manuscrit : La vidéosurveillance "intelligente" et l'information contextuelle. Nous présentons quelques travaux et projets visant à développer les systèmes de vidéosurveillance intelligente, et nous faisons une étude comparative des techniques

de modélisation des informations contextuelles. Ensuite nous faisons un rappel des notions relatives à notre problématique telles que le filtrage des données et le requêtage flou.

Le **Chapitre 3** présente le concept de filtrage négatif dont une application aux systèmes de vidéosurveillance est développée dans le cadre du projet ANR FILTER2 [1]. Premièrement nous définissons le filtrage négatif. Ensuite, nous proposons une analyse et une modélisation des métadonnées, et les algorithmes afférents pour le filtrage négatif. Puis, nous illustrons les différentes étapes des algorithmes développés à travers des exemples d'application.

Le **Chapitre 4** développe la notion d'enrichissement contextuel. Nous commençons par définir cette notion, ensuite nous décrivons les étapes génériques pour sa mise en œuvre. Nous détaillons les sources d'informations contextuelles utiles dans notre cas d'étude (vidéosurveillance) et nous proposons une modélisation de ces informations multi sources en tenant compte de leur aspect spatial et temporel. L'interrogation "intelligente" des informations contextuelles (métadonnées) est rendue possible grâce au mécanisme de requêtage proposé qui intègre les notions de préférences floues.

Le **Chapitre 5** présente notre proposition d'enrichissement de la norme ISO 22331/ IEC 79. Nous proposons un modèle générique de métadonnées de vidéosurveillance aussi bien utilisable pour des besoins d'interopérabilité que pour la recherche et le filtrage des contenus vidéo.

Le **Chapitre 6** est consacré à l'implémentation des modèles de métadonnées, des algorithmes et du mécanisme de requêtage proposés. Nous commençons par une description de l'architecture du framework proposé, puis une présentation des jeux de données utilisés. Ensuite, nous proposons des évaluations pour les expériences menées.

Le **Chapitre 7** présente dans la première partie une conclusion générale qui fait un rappel des objectifs de la thèse et des contributions réalisées. Dans la seconde partie, nous évoquons les perspectives envisagées.

1. https ://anr.fr/Projet-ANR-16-CE39-0013

Chapitre 2

Etat de l'art

Ce chapitre présente une revue de la littérature des approches visant à améliorer la recherche dans les contenus de vidéosurveillance. Nous commençons par une présentation des systèmes de vidéosurveillance automatique, puis une discussion des approches d'analyse des contenus vidéo, des récents projets et travaux dans le domaine de la vidéosurveillance. Ensuite nous introduisons la notion d'information contextuelle qui sont des informations utilisables dans le processus de requêtage des vidéos, puis nous présentons une discussion des techniques de modélisations de ces informations. Enfin, nous examinons les approches d'interrogation basées sur la logique floue qui sont prometteuses pour les systèmes caractérisés par l'exploitation de nombreuses sources d'information et nécessitant une interrogation efficace.

2.1 Systèmes de vidéosurveillance

La vidéosurveillance consiste à surveiller à distance des espaces publics ou privés à l'aide de caméras de surveillance fréquemment déployées dans des zones sensibles ou critiques (aéroports, stations de métro, centres commerciaux, parcs, intersections routières, etc.). Les images filmées par ces caméras sont généralement transmises à un centre de contrôle pour être exploitées en fonction des besoins et selon deux modes : (i) l'exploitation en temps réel qui consiste à visualiser immédiatement les images par les opérateurs pour des besoins ponctuels tels que les alarmes, la surveillance du trafic, la détection et le suivi d'objets, etc. et (ii) l'exploitation *a posteriori* qui consiste à enregistrer les images puis les analyser afin de résoudre des enquêtes et/ou de collecter des preuves suite à un évènement particulier (agression, homicide, enlèvement, terrorisme, etc.). Les travaux menés dans cette étude s'appliquent principalement à l'exploitation *a posteriori* de grands volumes de vidéos issues des systèmes de vidéosurveillance.

Traditionnellement, les systèmes de vidéosurveillance se définissent comme un ensemble de caméras de surveillance qui enregistrent des images, et des opérateurs qui les visionnent et attendent qu'un évènement anormal survienne. Dans de tels systèmes, la plupart des tâches de surveillance reposent sur l'observation humaine. Cependant, à mesure que le nombre de caméras augmente, la surveillance d'évènements par des opérateurs humains devient de plus en plus difficile, et sujette à l'erreur et à la fatigue cognitive de ceux-ci. Les grands volumes de vidéos générés par les caméras rendent

leur exploitation quasi impossible et leur exploitation *a posteriori* très longue et très coûteuse en terme de ressources humaines. Dans le but d'augmenter la robustesse de la vidéosurveillance et de réduire la charge de travail des opérateurs humains, de nombreux systèmes de surveillance dits automatiques, et capables par exemple de détecter des évènements anormaux [Lim et al., 2014], [Mathur and Bundele, 2016], de déclencher des alertes [Lee et al., 2013], de faciliter la recherche dans les séquences enregistrées [Klontz and Jain, 2013], etc. ont vu le jour.

Un système de vidéosurveillance automatique tel qu'illustré à la Figure 2.1 se compose de cinq principaux éléments [Amer and Regazzoni, 2005] : les caméras de surveillance, le réseau, des écrans de surveillance ou une salle de contrôle, une base de données vidéo, et une unité de traitement des vidéos. Les caméras de surveillance capturent et transmettent les vidéos à l'unité de traitement vidéo via le réseau. L'unité de traitement vidéo peut être un ordinateur à usage général ou un équipement informatique dédié tel qu'un serveur. Ainsi, l'unité de traitement vidéo transmettra les informations pertinentes extraites au centre de contrôle, par exemple, une alarme en réponse à un intrus qui s'est introduit dans une zone particulière, la détection d'un objet cible (personnes, véhicules), des séquences vidéo d'intérêt, etc. La base de données vidéo est utilisée pour stocker les vidéos et les données associées aux contenus (métadonnées) en vue d'un éventuel traitement ultérieur. En résumé, un système de surveillance automatique déclenche une alarme chaque fois qu'un évènement particulier est détecté. Cette automatisation des traitements vise à mettre en place la vidéosurveillance intelligente.

La "vidéosurveillance intelligente" consiste à faire analyser automatiquement les vidéos par des algorithmes capables de détecter et de suivre des objets d'intérêt au cours du temps, et capables de détecter des activités, évènements, ou comportements suspects particuliers, le but étant d'alerter les opérateurs en cas d'évènements spéci-

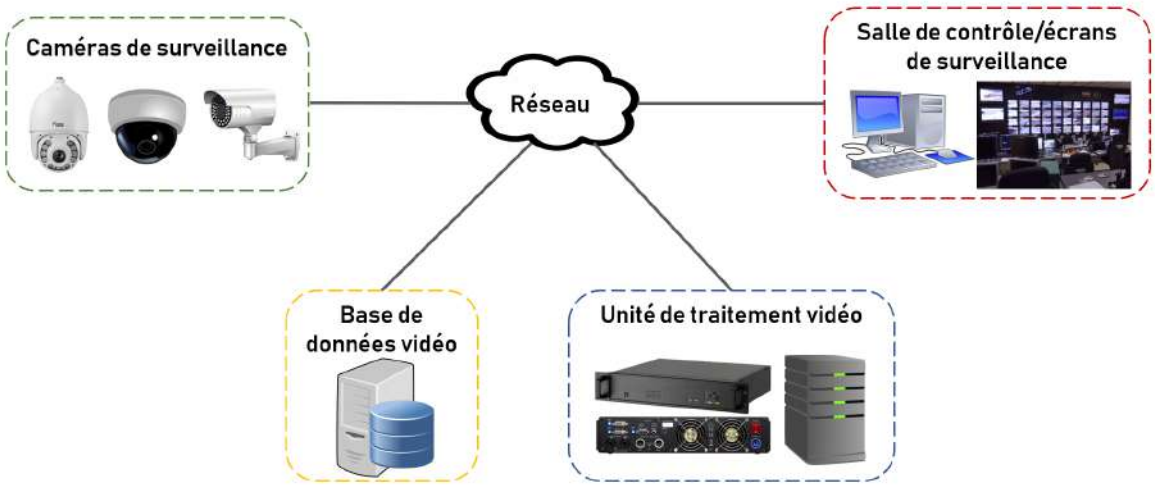


FIGURE 2.1 – Schéma d'un système de vidéosurveillance automatique.

fiques, de se focaliser uniquement ou en priorité sur les données pertinentes pour la surveillance, et d'améliorer les capacités de recherche dans les séquences enregistrées. La vidéosurveillance "intelligente" peut donc pré-filtrer de grandes quantités de données de surveillance et, si les données (vidéos) contiennent un évènement inhabituel ou significatif, alerter l'opérateur. La vidéosurveillance "intelligente" peut également fournir un niveau de détail qui permet de détecter et d'identifier plus précisément les objets et d'analyser leurs mouvements en temps réel.

La vidéosurveillance intelligente fait l'objet d'une attention accrue en raison de la demande grandissante en matière de sécurité et de sûreté. En France, les besoins en outils d'assistance à l'investigation exprimés par les services de police s'articulent autour de trois missions de sécurité intérieure telles que définies par le ST(SI)^2 (*Service des Technologies et des Systèmes d'Information de la Sécurité Intérieure*) [Sèdes et al., 2012] :

— **la prévention et la sécurisation :** cette mission s'inscrit dans le cadre d'une surveillance vidéo en temps réel et son objectif est de sécuriser des lieux et des évènements (manifestations, sites critiques, etc.) par l'intermédiaire des outils d'aide à la recherche en temps réel permettant à l'opérateur de focaliser son attention sur des évènements pertinents grâce au déclenchement automatique d'une alarme par exemple ;

— **le renseignement :** l'objectif de cette mission consiste à recueillir tout type d'informations ciblées sur des individus ou des sociétés par exemple. Le besoin en outils d'assistance dans ce cadre concerne la génération d'alerte quand des évènements prédéfinis surviennent (ex. : la détection d'une personne), l'extraction d'informations dans les flux de données recueillies (ex. : l'écoute ou la lecture labiale), et l'aide au recoupement d'informations entre toutes les données acquises (ex. : la reconnaissance d'une personne). Il est important de distinguer détection et reconnaissance. La détection de personne signifie qu'un système est capable de repérer la présence d'une personne dans une image ou une vidéo, mais pas de l'identifier. La reconnaissance peut confirmer l'identité de la personne.

— **l'enquête :** pour cette mission de sécurité interne, il s'agit de traiter *a posteriori* de grands volumes de vidéos collectées pour une enquête et provenant de diverses sources qui sont le plus souvent hétérogènes : réseaux de vidéosurveillance urbains, réseaux privés (magasins, banques, etc.), réseaux mobiles, etc. Cette mission est sans doute la plus complexe, car elle nécessite dans un premier temps l'ingestion de grands volumes de données hétérogènes afin de faciliter leur exploitation (ex. : visualisation, recoupement d'information), puis, une structure matérielle et applicative pour implémenter et exécuter un ensemble d'opérations (filtrages des contenus, traitement d'images, etc.), et permettre des recherches sémantiques et contextuelles sur les vidéos.

De nombreux travaux de recherche et commerciaux se sont focalisés sur les deux premières missions de sécurité et ont donné naissance à une large gamme d'applications dans le domaine du contrôle d'accès [Semertzidis et al., 2010], [Kitchin, 2014], de l'identification des personnes [Dadgostar et al., 2011], de la détection des anomalies [Wang et al., 2011] et des alarmes [Ahmed et al., 2010]. Par contre, les travaux concernant la recherche des preuves vidéos ou numériques *a posteriori* (Forensic) sont moins nombreux, moins matures, et de nombreuses (méta)données restent inexploitées de nos jours.

Bien que l'exploitation *a posteriori* des vidéos dans le cadre d'une enquête puisse également traiter les mêmes scénarios que l'exploitation traditionnelle en temps réel (par exemple centres commerciaux ou stations de métro), la nature de l'analyse est différente : la première est axée sur le traitement de très grandes quantités de données issues de plusieurs systèmes de vidéosurveillance, appartenant à des organismes différents et utilisant des technologies différentes. Dès lors, il est difficile d'exploiter sans outils appropriés ces masses de données hétérogènes.

L'intérêt pour le traitement *a posteriori* des vidéos a poussé de nombreux chercheurs à proposer des solutions au problème de la recherche des extraits vidéo "preuves" dans les collections de vidéosurveillance. La majorité des travaux proposés se focalisent sur le développement d'outils d'*analyse des contenus vidéo* afin de détecter et/ou suivre des objets [Edelman and Bijhold, 2010], [Chen et al., 2013], de reconnaître des actions [Niebles et al., 2008], évènements [Gerónimo and Kjellström, 2014] ou scènes [Lee and Pagliaro, 2013], [van den Eeden et al., 2016], d'analyser le comportement des foules humaines [Yogameena and Priya, 2015], etc.

2.1.1 Approches basées sur l'analyse des contenus vidéo

L'un des principaux problèmes liés à *l'analyse des contenus vidéo* est le fossé sémantique ("gap") entre les éléments visuels de bas niveau et la sémantique du contenu de haut niveau. Les humains ont tendance à utiliser des concepts de haut niveau dans la vie de tous les jours. Cependant, ce que les techniques actuelles de vision par ordinateur peuvent automatiquement extraire de l'image sont surtout des caractéristiques de bas niveau. Actuellement, la plupart des recherches se concentrent sur l'extraction de caractéristiques visuelles de niveau intermédiaire qui, d'une part, pourraient être dérivées de caractéristiques de bas niveau comme la couleur et le mouvement ; d'autre part, elles pourraient être utilisées pour révéler partiellement la sémantique vidéo sous-jacente. La majorité des solutions proposées pour l'analyse des contenus vidéo prennent en compte l'analyse de la structure de la vidéo. En général, les vidéos sont structurées selon une hiérarchie descendante de clips vidéo, de scènes, de séquences et d'images. L'analyse de contenus vidéo vise à segmenter une vidéo en un

certain nombre d'éléments structuraux qui ont une unité sémantique. Parmi les solutions d'analyse de contenus vidéo, les plus répandues proposent des approches basées sur : la détection de limites des séquences [Ling et al., 2008], l'extraction d'images clés [Nasreen and Shobha, 2013] et la segmentation des scènes [Hu et al., 2011].

Les méthodes de détection de limites des séquences consistent généralement à extraire d'abord les éléments visuels de chaque image, puis à mesurer les similitudes entre les images à l'aide des éléments extraits et, enfin, à détecter les limites des séquences entre les images qui ne sont pas identiques. Les approches de détection des limites des séquences peuvent être basées soit sur des seuillages [Wu et al., 2008], [Xia et al., 2007], soit sur l'apprentissage statistique [Chang et al., 2007]. La principale limite des approches basée sur les seuils est que la détection de limites des séquences dépend totalement du seuil qui est difficile à déterminer. Les approches statistiques basées sur l'apprentissage sont limitées par le fait qu'elles s'appuient fortement sur un ensemble de données d'apprentissage bien choisi.

Les images clés sont celles sélectionnées parmi l'ensemble des images redondantes d'une même séquence et qui reflètent le mieux le contenu de la séquence. Les approches d'extraction des images clés les plus utilisées actuellement sont basées sur le clustering, la simplification des contours, et la détection des objets/évènements. Ces approches ont pour principales limites : leur dépendance aux résultats du clustering qui est très difficile à mettre en place, surtout pour les grands volumes de données ; la grande complexité de calcul liée à l'obtention de la meilleure représentation des contours d'images ; la forte dépendance des algorithmes de détection d'objets/évènements aux règles heuristiques spécifiées en fonction de l'application, qui par conséquent, rend ces algorithmes efficaces uniquement dans les cas où les paramètres expérimentaux sont soigneusement choisis.

La segmentation des scènes est encore appelée segmentation par unités d'histoire. En général, une scène est un groupe de séquences contiguës qui sont cohérentes avec un sujet ou un thème donné. Les scènes ont une sémantique de plus haut niveau que les séquences. Les scènes sont identifiées ou segmentées en regroupant des séquences successives ayant un contenu similaire dans une unité sémantique significative. Le regroupement peut être basé sur des informations provenant des textes, d'images ou de la piste audio de la vidéo. Les approches de segmentation des scènes sont en général basées sur la détection des images clés. Dans ces approches, les séquences sont représentées par un ensemble d'images clés sélectionnées, qui souvent ne représentent pas efficacement le contenu dynamique des séquences car les séquences d'une scène sont corrélées par le contenu dynamique de la scène plutôt que par les similitudes entre les séquences d'images clés. Par conséquent, deux prises de vue sont considérées comme similaires si leurs images clés se trouvent dans le même environnement plutôt que si elles sont visuellement similaires.

En général, l'analyse automatique des contenus vidéo issus des systèmes de vidéosurveillance est confrontée aux problèmes suivants : (i) le grand volume de données qui entraîne des coûts d'exécution élevés, (ii) la difficulté à développer des algorithmes d'analyse de contenus vidéo assez génériques (la plupart des algorithmes d'analyse vidéo ont été développés pour des applications spécifiques et entraînés avec des données bien choisies, par conséquent il reste difficile d'utiliser ces algorithmes hors des domaines d'applications pour lesquels ils ont été entraînés), et (iii) le manque de robustesse des systèmes aux variations de l'environnement et à la complexité des scènes urbaines.

De nombreux travaux menés pour le développement des systèmes de vidéosurveillance intelligents se focalisent sur le développement d'outils d'analyse de contenu, mais les différents problèmes mentionnés dans les paragraphes précédents rendent inefficaces les algorithmes d'analyse de contenu dans certains environnements ou applications, d'où la réduction de la précision des résultats obtenus. Par conséquent, il n'y a pas de solution d'analyse de contenu utilisable dans l'ensemble des systèmes de vidéosurveillance. Par ailleurs, de nouvelles approches basées sur les métadonnées (informations contextuelles), permettant le filtrage des contenus et l'interrogation intelligente des vidéos sans avoir recours à une indexation exhaustive basée sur le contenu, pourraient accélérer l'essor de la vidéosurveillance intelligente.

2.1.2 Récents projets et travaux dans le domaine de la vidéosurveillance

Au cours des dernières années, l'analyse automatique des vidéos a suscité beaucoup d'intérêts chez de nombreux chercheurs et entreprises spécialisées dans le développement des logiciels de vidéosurveillance intelligente. De nombreuses solutions ont été proposées et de nombreux projets collaboratifs ont été mis en place tant au niveau national qu'au niveau européen. Nous présentons dans ce qui suit quelques travaux et projets relatifs à l'analyse automatique des vidéos, en évoquant leurs objectifs, et en indiquant s'ils proposent un filtrage avant les traitements automatiques, s'ils prennent en compte les informations contextuelles et s'ils sont applicables aux traitements *a posteriori*.

CARETAKER [Carincotte et al., 2006] est un projet Européen qui s'inscrivait dans le contexte de la surveillance des stations de métro via l'exploitation des flux de nature vidéo et audio. Son objectif était de reconnaître un ensemble d'évènements ou d'identifier d'autres types d'évènements grâce à l'analyse de ces flux. Le projet a permis de développer des techniques d'extraction automatique des métadonnées sémantiques pertinentes à partir des contenus vidéo. Par contre, aucun filtrage n'est fait avant l'extraction des connaissances des flux vidéo.

Le projet VANAHEIM (Video/Audio Networked surveillance system enhAncement through Human-cEntered adaptIve Monitoring) a permis de développer une technique pour le filtrage automatique en temps réel des vidéos grâce à des algorithmes permettant de détecter des activités anormales. Mais l'implémentation des algorithmes d'apprentissage utilisés dans le processus de filtrage semble complexe pour des grands volumes de données.

Le projet SURTRAIN (SURveillance des Transports par Analyse de l'Image et du soN) a pour but de détecter automatiquement des situations critiques à partir des images et du son. Les modèles 3D des objets sont utilisés pour assurer le suivi intra et inter caméra. Dans SURTRAIN, le déclenchement d'une procédure de suivi peut se faire par le mode de perception audio. Plusieurs situations critiques (ex : agressions, altercations) sont caractérisées par la présence des paroles ou cris au niveau sonore très élevé. Le projet a permis de développer des fonctions de détection et de localisation permettant de déclencher la procédure de suivi via l'activation de la caméra la plus proche de la situation critique en cours. Ce projet offre un système de surveillance vidéo et audio intelligent à bord des véhicules, mais n'intègre pas le traitement des requêtes *a posteriori*.

Très récemment (2017 à 2020), notre équipe a collaboré sur le projet européen VICTORIA (Video analysis for Investigation of Criminal and TerrORIst Activities). Face à la masse de vidéos collectées dans le cadre des investigations liées aux actes criminels majeurs et aux attaques terroristes, l'objectif du projet était de développer une plateforme d'analyse vidéo accélérant les tâches de traitement vidéo et d'exploration des données, aujourd'hui, encore effectuées manuellement par les enquêteurs. Les travaux de l'équipe ont permis de développer une modélisation générique des métadonnées attachées à la capture des vidéos ou extraites ultérieurement lors des analyses vidéos, et un mécanisme de requêtes avancé afin d'optimiser la fouille des données et de donner l'accès aux informations pertinentes requises par les enquêteurs.

Dans [Deng et al., 2010], les auteurs présentent un système d'analyse et de récupération d'évènements vidéo utilisant des techniques informatiques géospatiales. A partir du suivi des cibles et de l'analyse des flux vidéo des réseaux de caméras distribuées, le système génère des métadonnées de suivi vidéo pour chaque vidéo, les représente sur une carte et les fusionne en une coordonnée géospatiale uniforme. Les métadonnées combinées sont sauvegardées dans une base de données spatiales où les trajectoires cibles sont représentées en géométrie et en type de données géographiques. La base de données spatiales fournit au système une plateforme stable, rapide et facile à gérer, ce qui est essentiel pour gérer de grandes quantités de données vidéo. L'index spatial fourni par la base de données permet l'interrogation en ligne par l'élagage rapide de données sans rapport et le travail sur les données d'intérêt. Par contre, il n'y a aucun filtrage avant l'étape de génération des métadonnées de suivi vidéo pour chaque vidéo

qui peut être très coûteuse en temps.

Un des travaux récents de notre équipe [Codreanu, 2015] portait sur la modélisation des métadonnées spatio-temporelles associées aux contenus vidéos et l'interrogation de ces métadonnées à partir des trajectoires hybrides. L'objectif était d'effectuer un filtrage spatio-temporel des collections vidéo en se basant sur les métadonnées associées aux contenus vidéos. Ces travaux s'appliquent dans les environnements outdoor. Une extension aux environnements indoor a été proposée dans [Panta and Sèdes, 2016b], [Panta and Sèdes, 2016a], [Sèdes and Panta, 2017], [Panta et al., 2018]. Une des perspectives de ces travaux consistait à s'appuyer sur des métadonnées pertinentes dans le contexte de la vidéosurveillance pour réduire l'espace et implicitement le temps de recherche, car la variété et le grand volume des contenus vidéos rendent impossible leur analyse exhaustive, d'autres mesures de filtrage "négatif" pouvant être développées en se basant sur les métadonnées ou sur les caractéristiques des images (exemple : qualité d'image).

Globalement, beaucoup de travaux menés pour le développement des systèmes de vidéosurveillance intelligents visent à développer des outils d'analyse de contenus. Mais les problèmes tels que le volume de données générées, l'hétérogénéité des systèmes (contextes d'acquisition, formats de données et métadonnées, etc.) et la qualité très variable des enregistrements rendent inutile voire impossible l'utilisation des algorithmes d'analyse automatique de contenus. Face à ces problèmes, la mise en oeuvre d'une approche basée sur la modélisation des informations provenant d'autres sources (informations contextuelles, métadonnées) et permettant le filtrage et l'interrogation intelligente des collections vidéo serait une solution qui exclut le recours à une indexation exhaustive basée sur le contenu. L'idée est de recourir à d'autres sources d'informations telles que les métadonnées techniques (ex : installation, position, champ de vue des caméras), les métadonnées issues des algorithmes d'analyse du contenu (ex : qualité), et les informations contextuelles (ex : open data, médias sociaux), pour définir des mesures de filtrage et faciliter l'interrogation des vidéos. Parmi les potentielles sources d'informations sur lesquelles peut s'appuyer une telle approche, nous avons la norme ISO 22311 renommée en IEC 79.

La norme ISO 22311 a été publiée grâce à la participation commune des principaux acteurs du domaine de la vidéosurveillance en France et à l'international. Cette norme est destinée à des fins de sécurité sociétale et définit un format commun pour les données qui peuvent être extraites des systèmes de collecte de vidéosurveillance, par exemple à des fins d'enquête. Les dictionnaires de métadonnées définis dans la norme détaillent les informations relatives à la localisation des capteurs (caméras), la description des scènes observées, les caractéristiques des capteurs (ex : caméras), etc. Un enrichissement des dictionnaires de la norme est envisagée grâce à l'apport des métadonnées issues des algorithmes d'analyse automatique de contenu et d'autres

informations contextuelles (données ouvertes et liées, médias sociaux, mobilité, etc.), afin de généraliser la mise en œuvre de la norme.

2.2 Informations contextuelles

2.2.1 Définition du contexte

L'information contextuelle ou le contexte est un terme imprécis qui possède plusieurs significations. Diverses définitions du contexte ont été proposées dans la littérature, mais elles offrent peu d'indices sur les propriétés qui sont intéressantes pour la modélisation du contexte. Les auteurs de [Dey et al., 2001] ont évalué et souligné les faiblesses de ces définitions. Ils ont affirmé que les définitions fournies par [Ward et al., 1997], [Rodden et al., 1998], [Franklin and Flaschbart, 1998] utilisent des termes tels que l'environnement et la situation pour décrire le contexte. Mais, ces définitions ne peuvent pas être utilisées pour identifier un nouveau contexte. Dans [Abowd and Mynatt, 2000], cinq termes (qui, quoi, où, quand, pourquoi) ont été identifiés comme informations minimales nécessaires pour comprendre le contexte. Pas satisfaits d'une définition générale, de nombreux chercheurs ont tenté de définir le contexte en énumérant des exemples de contextes. Certains travaux [Schilit et al., 1994], [Pascoe, 1998] ont divisé le contexte en trois catégories (Figure 2.2) : contexte informatique (ex : connectivité réseau, coûts de communication et la bande passante de communication), contexte utilisateur (ex : profil utilisateur, son emplacement, les personnes à proximité, situation sociale), et contexte physique (ex : éclairage, niveaux de bruit, conditions de circulation, température). [Dey et al., 2001] a prétendu que ces définitions étaient aussi

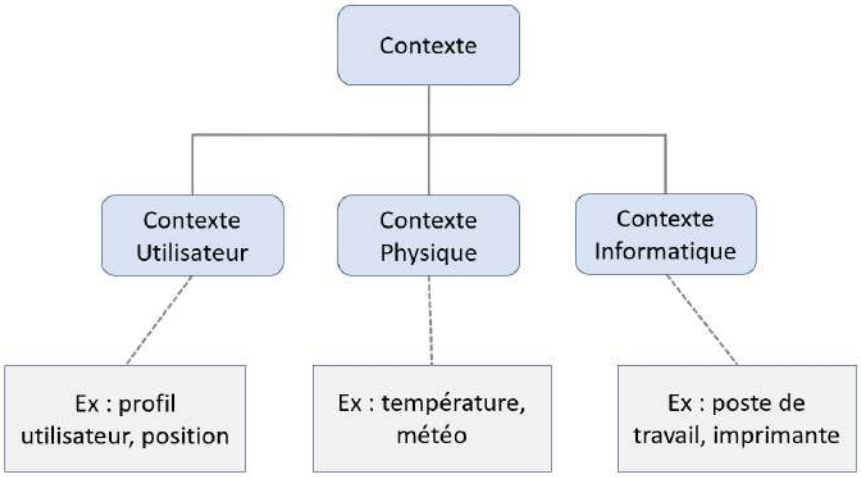


FIGURE 2.2 – Différentes catégories de contexte et exemples.

spécifiques et ne pouvaient être utilisées pour identifier le contexte dans un sens plus large et a fourni une définition du contexte comme suit : *"le contexte est toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité est*

une personne, un lieu ou un objet qui est considéré comme pertinent pour l'interaction entre un utilisateur et une application, y compris l'utilisateur et les applications elles-mêmes".

Une application contextuelle ou un système sensible au contexte est un système qui utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur, ou qui considère l'information contextuelle pour la prise de décision. Les applications contextuelles sont généralement développées selon l'une des trois approches suivantes [Hu et al., 2008] :

— Pas de modèle contextuel au niveau de l'application : chaque application communique directement avec les capteurs et les autres sources d'information contextuelle, pré-traite les données brutes au niveau requis et évalue l'information pour prendre des décisions sur la façon de l'adapter.

— Modèle contextuel implicite : les applications sont développées à l'aide de bibliothèques/outils réutilisables pour le traitement des informations de contexte qui aident à la collecte et au pré-traitement des données. Cependant, le contexte est toujours étroitement lié à l'application.

— Modèle contextuel explicite : les applications ont leur propre contexte bien défini et utilisent une infrastructure de gestion de contexte partagée pour alimenter les modèles au moment de l'exécution à l'aide de sources contextuelles. La gestion du contexte et l'application sont clairement séparées et peuvent être développées et étendues indépendamment.

Dans les deux premières approches, il revient aux applications contextuelles de traiter les erreurs dans la collecte et le pré-traitement des informations contextuelles, ce qui augmente la taille et la complexité des applications et la difficulté de leur mise en œuvre. La troisième approche est basée sur des modèles formels explicites d'informations contextuelles (types d'informations contextuelles et qualité de ces informations) utilisés par des applications particulières et permet donc à plusieurs applications contextuelles de partager un ensemble de sources contextuelles communes et de composants de pré-traitement des informations contextuelles, limitant la charge des sources contextuelles contraintes par les ressources. De plus, cette approche permet de transférer les problèmes de tolérance aux pannes, de reconfiguration et une auto-réparation vers le système de gestion de contexte, libérant ainsi les développeurs des applications et leur permettant de se consacrer aux fonctionnalités et aux métiers souhaités logique de l'application.

Un système ou application de vidéosurveillance sensible au contexte doit pouvoir intégrer les données et métadonnées de vidéosurveillance, et les informations contextuelles afin de proposer des applications ou services utiles aux opérateurs de videosurveillance selon leurs besoins. Nous entendons par informations contextuelles, des

informations provenant de différentes sources telles que : les open data (informations sur les évènements survenus à savoir date, localisation, description, etc.), les médias sociaux (contenus partagés, liens entre profils, etc.), la mobilité (informations de localisations issues des capteurs attachés aux véhicules et personnes, informations issues des applications de mobilité, etc.) et le crowdsourcing (informations collectées du grand public). Une application de vidéosurveillance sensible au contexte doit permettre l'intégration entre des sources d'information multiples et hétérogènes (interopérabilité). Notre étude s'intéressera à la modélisation et l'intégration de ces informations contextuelles multi sources et hétérogènes.

2.2.2 Modélisation des informations contextuelles

Il existe de nombreux types d'informations contextuelles. Leurs différentes propriétés conduisent à différentes façons de les exprimer et de les modéliser. A notre connaissance, tous les systèmes actuels utilisent leur propre méthode pour modéliser l'information contextuelle. Les techniques de modélisation de contexte les plus populaires sont étudiées dans [Chen and Kotz, 2000], [Strang and Linnhoff-Popien, 2004]. La mise en œuvre réelle de ces techniques peut varier considérablement selon le domaine d'application (par exemple, les détails de mise en œuvre peuvent différer d'un environnement embarqué à un autre, des environnements mobiles vers les environnements basés sur le cloud). Par conséquent, nous nous concentrons sur la perspective conceptuelle de chaque modélisation technique et non sur l'implémentation spécifique. Notre discussion est basée sur les six techniques de modélisation de contexte les plus populaires : clé-valeur, schémas de balisage, graphique, orientée objet, basée sur la logique et basée sur l'ontologie. Une comparaison de toutes ces techniques est présentée dans le tableau 2.1.

2.2.2.1 Modélisation clé-valeur

Les informations contextuelles sont modélisées sous forme de paires clé-valeur dans différents formats tels que les fichiers texte et les fichiers binaires. Le modèle de paires clé-valeur est la structure de données la plus simple pour modéliser des informations contextuelles. Cependant, cette modélisation n'est pas évolutive et ne convient pas au stockage de structures de données complexes. En outre, les structures hiérarchiques ou les relations ne peuvent pas être modélisées à l'aide de paires clé-valeur. Par conséquent, le manque de capacité de structuration des données rend difficile la récupération efficace de l'information modélisée. De plus, il n'est pas possible de joindre des méta-informations. La technique clé-valeur est une technique orientée application et adaptée aux besoins de stockage temporaire tels que des configurations d'application et des préférences utilisateur moins complexes.

2.2.2.2 Modélisation basée sur le balisage

Cette modélisation utilise une structure de données hiérarchique composée de balises avec attributs et contenus. En particulier, le contenu des balises est généralement défini de façon récursive par d'autres balises. Cette technique constitue une amélioration par rapport à la technique de modélisation clé-valeur. L'avantage d'utiliser des balises est qu'elles permettent une récupération efficace des données. En outre, la validation est prise en charge par les définitions du schéma. Des outils de validation sophistiqués sont disponibles pour les techniques de balisage populaires telles que XML. La vérification de l'étendue de mesure est également possible jusqu'à un certain point pour les valeurs numériques. Les schémas de balisage tels que XML sont largement utilisés dans presque tous les domaines d'application pour stocker temporairement des données, transférer des données entre applications et transférer des données entre composantes. En revanche, les langages de balisage n'offrent pas de capacités expressives avancées qui permettent le raisonnement. De plus, en raison de l'absence de spécifications de conception, la modélisation du contexte, la récupération, l'interopérabilité et la réutilisabilité sur différents schémas de balisage peuvent être difficiles. Une application courante de la modélisation basée sur le balisage est la modélisation des profils. Les profils sont généralement élaborés à l'aide de langages tels que XML. Cependant, le concept de langage de balisage n'est pas limité uniquement au XML. Tout langage ou mécanisme (ex : JSON) qui prend en charge le stockage basé sur des balises permet la modélisation basée sur le balisage. Un exemple de modélisation d'un schéma de balisage populaire est CC/PP [Klyne et al., 2004]. Il existe un nombre important d'applications similaires telles que ContextML [Knappmeyer et al., 2010] dans l'informatique contextuelle. Les tuples sont également utilisés pour modéliser le contexte [Yanwei et al., 2011].

2.2.2.3 Modélisation graphique

Cette technique modélise le contexte avec les relations. Un outil de modélisation général très bien connu est le langage de modélisation unifié (UML) qui a une forte composante graphique (diagrammes UML). En raison de sa structure générique, UML est également approprié pour modéliser le contexte. C'est ce que montre par exemple [Bauer et al., 2003], où les aspects contextuels pertinents pour la gestion du trafic aérien sont modélisés comme des extensions UML. Un autre exemple d'outil est le modèle objet-rôle (ORM) dont une extension est proposée dans [Henricksen et al., 2003].

En termes de richesse expressive, la modélisation graphique est meilleure que la modélisation basée sur le balisage et clé-valeur car elle permet de saisir les relations dans le modèle contextuel. La représentation réelle des bas niveaux de la technique de modélisation graphique pourrait varier. Par exemple, il peut s'agir d'une base de

données SQL, noSQL, XML, etc. Les bases de données peuvent contenir des quantités massives de données et fournir des opérations simples d'extraction de données, qui peuvent être effectuées rapidement. En revanche, le nombre d'implémentations différentes (c'est-à-dire différentes bases de données et autres solutions) rend difficile l'interopérabilité. De plus, il y a des limites aux mécanismes d'extraction de données tels que SQL. En outre, des exigences sophistiquées de récupération de contexte peuvent exiger l'utilisation de requêtes SQL très complexes. Les requêtes peuvent être difficiles à créer, à utiliser et à gérer, même avec les outils sophistiqués qui existent aujourd'hui. L'ajout d'informations contextuelles et la modification de la structure des données sont également difficiles dans les étapes ultérieures. Cependant, certaines des tendances et solutions récentes du mouvement noSQL [Han et al., 2011a] permettent de surmonter ces problèmes d'altération structurelle. Par conséquent, la modélisation graphique peut être utilisée comme stockage persistant du contexte.

2.2.2.4 Modélisation orientée objets

Les concepts basés sur les objets (ou orientés objet) sont utilisés pour modéliser les données à l'aide des hiérarchies de classes et de relations. Le paradigme orienté objet favorise l'encapsulation et la réutilisation. Comme la plupart des langages de programmation de haut niveau prennent en charge les concepts orientés objet, la modélisation s'intègre facilement dans des systèmes contextuels. Ainsi, la modélisation basée sur les objets peut être utilisée comme un mécanisme interne, non partagé, basé sur le code, de modélisation du contexte d'exécution, de manipulation et de stockage. Par contre, il ne fournit pas de capacités de raisonnement intégrées. La validation des conceptions orientées objet est également difficile en raison de l'absence des normes et des spécifications.

2.2.2.5 Modélisation logique

Les informations contextuelles sont représentées par les faits, expressions et règles. Les règles sont également utilisées par d'autres techniques de modélisation, comme les ontologies. Les règles sont principalement utilisées pour exprimer les principes, les contraintes et les préférences. Cette modélisation offre une richesse expressive beaucoup plus grande que les autres techniques de modélisation citées précédemment. Ainsi, le raisonnement est possible jusqu'à un certain niveau. Les structures et langages spécifiques qui peuvent être utilisés pour modéliser le contexte à l'aide des règles sont variés. Par contre, l'absence de normalisation réduit la capacité de la réutilisabilité et l'applicabilité. De plus, des techniques graphiques hautement sophistiquées et interactives peuvent être utilisées pour développer des représentations logiques ou basées sur des règles.

2.2.2.6 Modélisation basée sur l'ontologie

Les ontologies représentent une description des concepts et des relations. Par conséquent, les ontologies sont un instrument très prometteur pour la modélisation de l'information contextuelle en raison de leur expressivité élevée et formelle, et des possibilités d'application des techniques du raisonnement ontologique. Un certain nombre de normes différentes (RDF, RDFS, OWL) et de capacités de raisonnement sont disponibles en fonction des besoins. Une large gamme d'outils de développement et de moteurs de raisonnement sont également disponibles. Par contre, l'extraction du contexte peut nécessiter beaucoup de calculs et de temps lorsque la quantité de données croît.

TABLE 2.1 – Comparaison des techniques de modélisation des informations contextuelles.

Techniques	**Avantages**	**Inconvénients**	**Applications**
Clé-valeur	- Simple - Flexible - Facile à gérer pour des petits volumes de données	- Fortement couplée avec les applications - Non évolutif - Aucune structure ou schéma - Difficile de récupérer l'information - Impossible de représenter les relations - Pas de support de validation - Aucun outil de traitement standard n'est disponible	Peut être utilisée pour modéliser une quantité limitée de données telles que les préférences de l'utilisateur et les configurations d'application. Il s'agit surtout d'informations indépendantes et non liées entre elles. Cela convient également au transfert de données limitées et à toute autre exigence de modélisation temporaire moins complexe.
Schéma de balisage	- Flexible - Plus structurée - Validation possible par le schéma - Disponibilité des outils de traitement.	- L'application est dépendante du fait qu'il n'y a pas de normes pour les structures - Peut-être complexe lorsque plusieurs niveaux d'information sont impliqués - Moins difficile de récupérer l'information.	Peut être utilisée comme format intermédiaire d'organisation des données ainsi que comme mode de transfert des données sur le réseau. Peut être utilisée pour dissocier les structures de données utilisées par deux composants d'un système (par exemple JSON comme format pour le transfert de données sur le réseau).

Graphique	- Permet la modélisation des relations - Facilite la recherche d'information - Disponibilité des normes et implémentations - Validation possible par des contraintes	- L'interrogation peut être complexe - La configuration peut être nécessaire - L'interopérabilité entre les différentes implémentations est difficile - Pas de normes, mais régie par des principes de conception.	Peut être utilisée pour l'archivage permanent à long terme et pour de grands volumes de données. Le contexte historique peut être stocké dans des bases de données.
Orientée objets	- Permet la modélisation des relations - Peut-être bien intégrée à l'aide de langages de programmation - Disponibilité des outils de traitement	- Difficile de récupérer l'information - Pas de normes, mais régie par des principes de conception - Manque de validation	Peut être utilisée pour représenter le contexte au niveau du code de programmation. Permet la manipulation du contexte d'exécution. Très court terme, temporaire, et la plupart du temps stockée dans la mémoire de l'ordinateur. Supporte également le transfert de données sur le réseau.
Basée sur la logique	- Permet de générer un contexte de haut niveau en utilisant un contexte de bas niveau - Simple à modéliser et à utiliser - Supporte le raisonnement logique - Disponibilité des outils de traitement	- Pas de normes - Manque de validation - Fortement couplée avec les applications	Peut être utilisée pour générer un contexte de haut niveau en utilisant un contexte de bas niveau (c.-à-d. générer de nouvelles connaissances), modéliser des évènements et des actions (c.-à-d. détection d'évènements), et définir des contraintes et des restrictions.

Basée sur l'ontologie	- Supporte le raisonnement sémantique - Permet une représentation plus expressive du contexte - Validation solide - Indépendant de l'application et permet le partage - Appui solide de la part des normes - Disponibilité des outils assez sophistiqués	- La représentation peut être complexe - La recherche d'information peut être complexe et exiger beaucoup de ressources.	Peut être utilisée pour modéliser la connaissance du domaine et structurer le contexte en fonction des relations définies par l'ontologie. Plutôt que de stocker des données sur les ontologies, les données peuvent être stockées dans des sources de données appropriées (c'est-à-dire des bases de données) tandis que la structure est fournie par les ontologies.

Le choix d'une technique de modélisation adaptée aux informations contextuelles traitées dans cette étude est un verrou à lever.

2.3 Intégration des informations contextuelles dans le filtrage des données

Le filtrage est un ensemble de processus par lesquels des données/informations sont fournies aux utilisateurs d'un système en fonction de leurs besoins (utilité, intérêt). Un système de filtrage surmonte le problème de surcharge d'information en fournissant aux utilisateurs les informations/contenus les plus pertinents. Lorsque l'information fournie à l'utilisateur est présentée sous forme de suggestions, le système de filtrage s'appelle système de recommandation. Les systèmes de filtrage ou de recommandation ont donc pour but d'aider les utilisateurs à trouver des données d'intérêt (séquences vidéo, pages Web, etc.) parmi les masses de données disponibles.

Trois grandes approches sont généralement utilisées pour le filtrage de l'information [Adomavicius and Tuzhilin, 2005] : le filtrage collaboratif, le filtrage basé sur le contenu et le filtrage hybride. Un système de filtrage basé sur le contenu sélectionne les éléments en fonction de la corrélation entre le contenu des éléments et les préférences de l'utilisateur, tandis qu'un système de filtrage collaboratif choisit les éléments en fonction de la corrélation entre les personnes ayant des préférences similaires. Les systèmes de filtrage hybrides combinent les deux approches précédentes et sont basés sur l'idée que l'intégration du contenu et de l'information sociale pourrait conduire à une meilleure technique de filtrage. Récemment, les systèmes de filtrage ou de recommandation contextuels, qui intègrent des informations contextuelles dans le filtrage, sont devenus l'un des sujets les plus abordés dans le domaine des systèmes de fil-

trage/recommandation [Yujie and Licai, 2010]. L'importance de l'information contextuelle a été étudiée par des chercheurs et des professionnels dans de nombreuses disciplines telles que la fouille de données (data mining), la personnalisation du commerce électronique, les systèmes ubiquitaires et mobiles, les bases de données, la recherche d'information, le marketing et la gestion [Adomavicius and Ricci, 2009].

Fouille de données (data mining). Dans le domaine du data mining, le contexte est parfois défini comme des évènements qui caractérisent le cycle de vie d'un client et qui peuvent déterminer un changement dans ses préférences, son statut et sa valeur pour une entreprise [Linoff and Berry, 2011]. Par exemple, la connaissance des informations de contexte telles que : la naissance d'un enfant, un nouvel emploi, le mariage, le divorce ou la retraite aide les modèles de fouilles relatifs à ces contextes à se concentrer uniquement sur les données pertinentes ou à sélectionner uniquement les résultats pertinents.

Personnalisation du commerce électronique. [Palmisano et al., 2008] utilisent le motif d'un achat effectué par un client dans une application de commerce électronique comme information contextuelle. Des motifs d'achat différents peuvent conduire à des comportements différents. Par exemple, une même cliente peut acheter sur le même compte en ligne différents produits pour différentes raisons : un livre pour améliorer ses compétences professionnelles personnelles, un livre comme cadeau ou un appareil électronique pour son passe-temps. Pour faire face aux différents motifs d'achat, [Palmisano et al., 2008] établissent un profil distinct d'un client pour chaque contexte d'achat, et ces profils distincts sont utilisés pour construire des modèles distincts permettant de prédire le comportement du client dans des contextes spécifiques et pour des segments spécifiques de clients. Une telle segmentation contextuelle de la clientèle est utile, car elle permet d'obtenir de meilleurs résultats des modèles prédictifs pour différentes applications de commerce électronique. Les systèmes de recommandation sont également liés à la personnalisation du commerce électronique, puisque des recommandations personnalisées de divers produits et services sont fournies aux clients. L'importance d'inclure et d'utiliser l'information contextuelle dans les systèmes de recommandation a été démontrée dans [Adomavicius et al., 2005], où les auteurs ont présenté une approche multidimensionnelle qui peut fournir des recommandations fondées sur l'information contextuelle en plus de l'information typique sur les utilisateurs et les éléments utilisés dans plusieurs applications de recommandation.

Systèmes ubiquitaires et mobiles. Les informations contextuelles sont capitales pour la fourniture des services basés sur la localisation aux clients mobiles [Schiller and Voisard, 2004]. Par exemple, le théâtre du Capitole peut vouloir recommander aux visiteurs du Capitole à Toulouse, trente minutes avant le début du spectacle, des billets de théâtre à prix fortement réduits (puisque ces billets seront de toute façon perdus après le début du spectacle) et envoyer ces informations aux smart-

phones ou autres dispositifs de communication des visiteurs. L'heure, l'emplacement et le type de dispositif de communication (ex : smart-phones) constituent des informations contextuelles dans cette application. L'importance de la prise en compte du contexte dans le paradigme de l'Internet des objets (IoT) a été démontrée dans [Perera et al., 2013].

Bases de données. Des capacités contextuelles ont été ajoutées à certains Systèmes de Gestion de Base de Données (SGBD) en incorporant les préférences des utilisateurs et en renvoyant différentes réponses aux requêtes de base de données selon le contexte dans lequel les requêtes ont été exprimées et selon les préférences particulières des utilisateurs correspondants aux contextes spécifiques. Plus précisément, dans [Mansoor et al., 2007], un ensemble de paramètres contextuels est introduit et des préférences sont définies pour chaque combinaison d'attributs relationnels réguliers et ces paramètres contextuels. Puis [Mansoor et al., 2007] présentent une extension contextuelle de SQL pour accommoder de telles préférences et informations contextuelles. Agrawal et ses collaborateurs [Agrawal et al., 2006] présentent une autre méthode pour intégrer le contexte et les préférences de l'utilisateur dans les langages d'interrogation et élaborent des méthodes de rapprochement et de classement des différentes préférences afin de fournir rapidement des réponses classées aux requêtes contextuelles. [Mokbel and Levandoski, 2009] décrivent le serveur de base de données CoreDB qui prend en compte le contexte et l'emplacement et discutent de plusieurs questions liées à sa mise en œuvre, y compris les défis liés aux opérateurs de requêtes contextuelles et aux requêtes continues, le traitement multi-objectif des requêtes et l'optimisation des requêtes.

Recherche d'information. L'information contextuelle s'est avérée utile pour la recherche et l'accès à l'information [Jones, 2005], bien que la plupart des systèmes existants basent leurs décisions de recherche uniquement sur les requêtes et les collections de documents, alors que l'information sur le contexte de recherche est souvent ignorée [Akrivas et al., 2002]. Dans la recherche sur le Web, le contexte est considéré comme l'ensemble des sujets potentiellement liés au terme de recherche. Par exemple, [Lawrence, 2000] décrit comment l'information contextuelle peut être utilisée et propose plusieurs moteurs de recherche contextuels spécialisés dans le domaine. L'intégration du contexte dans la composition des services Web est suggérée dans [Maamar et al., 2006]. La plupart des techniques actuelles d'accès et de recherche d'information, qui prennent en compte le contexte, se concentrent sur les problèmes à court terme, et les intérêts et demandes immédiats des utilisateurs (ex : " trouver tous les fichiers créés lors d'une réunion de printemps un jour ensoleillé devant un restaurant italien à Paris"). Elles ne sont pas conçues pour modéliser les préférences des utilisateurs à long terme.

Marketing et gestion. Les chercheurs en marketing ont soutenu que le proces-

sus d'achat dépend du contexte dans lequel la transaction a lieu, puisque le même client peut adopter différentes stratégies de décision et préférer différents produits ou marques selon le contexte [Panniello and Gorgoglione, 2012]. Les consommateurs varient dans leurs règles de décision en raison de la situation d'utilisation, de l'utilisation du bien ou du service (pour la famille, pour le cadeau, pour soi) et de la situation d'achat (vente par catalogue, sélection en magasin, et achat assisté par un vendeur). Par conséquent, les prévisions exactes des préférences des consommateurs devraient dépendre de la mesure dans laquelle l'information contextuelle pertinente a été intégrée. Dans la littérature marketing, le contexte a également été étudié dans le domaine de la théorie de la décision comportementale.

Contrairement à toutes ces applications (IoT, services de localisation, commerce électronique, etc.) pour lesquelles l'importance des informations contextuelles a été démontrée, il existe peu de travaux axés sur l'apport du contexte dans les **systèmes de vidéosurveillance**. Beaucoup d'applications de vidéosurveillance sensibles au contexte telles que [An and Kim, 2012], [Fragkiadaki et al., 2012], [Huang et al., 2012], [Nam et al., 2012] utilisent des approches dans lesquelles les informations contextuelles sont extraites des contenus afin d'effectuer le filtrage des vidéos. De telles approches utilisent des algorithmes d'analyse de contenu qui sont souvent très coûteux en temps. D'autres sources d'informations contextuelles (ex : données ouvertes) pourraient permettre de filtrer les vidéos sans recourir aux algorithmes d'analyse des contenus. Dès lors il convient de proposer une approche adéquate pour l'interrogation de ces informations contextuelles.

2.4 Interrogation basée sur la logique floue

2.4.1 Définition de la logique floue

Les méthodes et techniques basées sur des ensembles flous sont largement utilisées dans de nombreux domaines actuels de la science et de la technologie, au sens large dans l'informatique et en particulier dans l'aide à la décision. La raison est que les ensembles flous sont des outils adéquats et puissants lorsqu'il s'agit de manipuler des informations importantes et incontournables, mais aussi complexes, telles que des informations imprécises/incertaines. L'imprécision de l'information, son caractère flou (vague) est un phénomène omniprésent et inévitable. L'être humain a une capacité remarquable à comprendre des informations imprécises, à raisonner et à prendre les bonnes décisions sur cette base. Cette capacité semble faire partie des caractéristiques fondamentales de l'intelligence humaine. Prenons des exemples simples :

— *Prévisions météo : vent et généralement nuageux, bien que certaines ruptures de nuages soient probables dans le sud. Pluie, plus ou moins abondante, dans le*

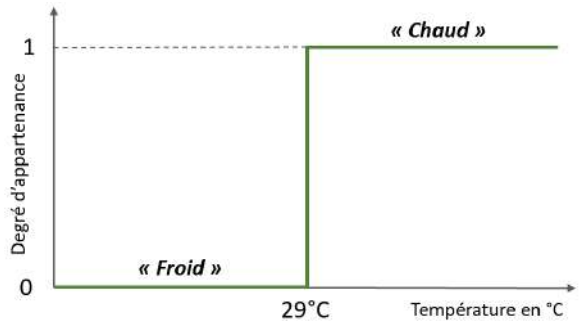
nord et le nord-ouest, et ailleurs majoritairement sec.

— *Description d'un agresseur donnée par un témoin oculaire : homme blanc ; environ 185 cm ; entre 35 - 40 ans ; athlétique ; cheveux foncés et courts ; yeux foncés.*

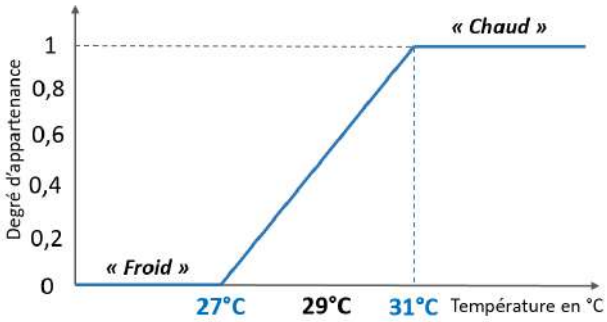
L'imprécision de ces déclarations n'est pas causée par le manque de soin ou la mauvaise volonté d'un individu, mais elle résulte de la nature même des notions et des questions qui y sont évoquées. Malgré cette imprécision, l'être humain est capable de reconnaître dans une foule une personne semblable à celle de la description, d'adapter ses activités et le type de vêtements aux prévisions météorologiques. Par contre, les applications informatiques nécessitent des aptitudes pour faire face aux difficultés liées à l'imprécision de ces informations. Une réponse efficace à ce besoin est l'utilisation de la logique floue qui a pour but de formaliser ou structurer l'information imprécise [Dubois and Prade, 1997], [Wygralak, 2013].

La logique floue, introduite par [Zadeh, 1965], est une approche de traitement d'information basée sur les **degrés de vérité** et non pas sur la logique classique ou logique booléenne habituelle **"vraie ou fausse"** (1 ou 0) sur laquelle l'ordinateur est basé. La logique floue permet de représenter et de manipuler les informations imprécises/incertaines et se rapproche de la flexibilité du raisonnement humain. Dans la logique classique, chaque fait ou proposition, tel que : *"il pleuvra demain"*, doit être vrai ou faux. Pourtant, de nombreuses informations utilisées par les humains comportent un certain degré d'incertitude. Comme la théorie des probabilités, la logique floue attache des valeurs numériques entre 0 et 1 à chaque proposition afin de représenter l'incertitude. Tandis que la théorie des probabilités mesure la probabilité que la proposition soit correcte, la logique floue mesure le degré auquel la proposition est correcte. Par exemple, la proposition *"le président français est jeune"* peut avoir un degré de certitude/justesse de 0,9. La distinction importante entre information probabiliste et logique floue est qu'il n'y a pas d'incertitude sur l'âge du président mais plutôt sur le degré auquel il correspond à la catégorie jeune. De nombreux termes tels que : *grand, riche, célèbre, ou sombre* sont valides uniquement à un certain degré lorsqu'ils sont appliqués à une personne ou une situation particulière. La logique floue essaie de mesurer ce degré et permet aux ordinateurs de manipuler ces informations.

La Figure 2.3 montrent la différence entre les descriptions du concept vague "chaud" par la logique booléenne classique (ou binaire) et par la logique floue. Selon la logique binaire (Figure 2.3a), pour toute température inférieure à 29°C, *"il fait froid"*. Cependant, dans la vie quotidienne, la façon de penser est complètement différente et similaire à la logique de la Figure 2.3b. Selon cette dernière logique, pour toute température inférieure à 27°C, *"il fait froid"*, tandis que toutes les températures supérieures à 31°C impliquent qu'"il fait chaud". En revanche, les températures comprises entre 27°C et 31°C appartiennent à la fois aux ensembles "froid" et "chaud" avec des valeurs



(a) Description possible par la logique booléenne



(b) Description possible par la logique floue

FIGURE 2.3 – Différence entre la logique booléenne et la logique floue

d'appartenance spécifiques.

2.4.2 Requêtage flou ("fuzzy querying")

Il y a quelques années, on a assisté à un intérêt croissant pour l'expression des préférences dans les requêtes de base de données. Par ailleurs, les premiers travaux de recherche sur ce sujet remontent à la fin des années 80 (ex : [Ichikawa and Hirakawa, 1986]). Les motifs d'une telle préoccupation sont multiples. Premièrement, il est devenu souhaitable d'utiliser des langages d'interrogation plus expressifs et plus fidèles à ce qu'un utilisateur a l'intention de demander. Deuxièmement, l'introduction de préférences dans les requêtes fournit une base pour classer les éléments récupérés, ce qui est particulièrement utile dans le cas d'un grand nombre d'éléments satisfaisant une requête. Troisièmement, une requête classique peut aussi avoir un ensemble de réponses vides, tandis qu'une version flexible (et donc moins restrictive) de la requête pourrait correspondre à certains critères. Dans de nombreux cas, les utilisateurs ne sont pas certains des caractéristiques des éléments qu'ils essaient d'extraire d'une base de données. Ils ne peuvent pas fournir de termes ou de valeurs précises pour effectuer la requête. Dans ces cas, un système qui permet aux utilisateurs d'exécuter une requête d'une manière flexible peut résoudre leurs besoins de manière satisfaisante.

L'interrogation floue des données permet à l'utilisateur de formuler des requêtes flexibles afin d'obtenir des résultats satisfaisants. Plus précisément, les requêtes flexibles se présentent sous la forme :

"Quels objets de la base de données sont p ?",

où p désigne une propriété arbitraire, généralement imprécise, exprimée dans une langue naturelle. Exemples de requêtes flexibles :

(R1) "Quelles sont les vidéos de qualité ***acceptable***, filmées à ***proximité*** du Capitole ?",

(R2) "Quels sont les hôtels ***moins chers*** situés aux ***environs*** du centre de Paris ?".

L'idée clé de l'interrogation flexible est d'introduire des préférences à l'intérieur des requêtes. Les préférences sont définies à deux niveaux [Zadrozny et al., 2009] dans la requête : à l'intérieur des conditions et entre les conditions. Dans le premier cas, l'objectif est d'exprimer que certaines valeurs sont plus adéquates que d'autres (de la satisfaction totale au rejet complet), tandis que dans le second cas, les préférences sont destinées à des niveaux d'importance associés aux conditions. Cette approche généralise les interrogations booléennes lorsqu'une condition est satisfaite ou non et que toutes les conditions sont d'importance égale. L'intérêt majeur des requêtes flexibles réside dans le fait que leurs réponses ne sont plus un ensemble fixe, mais sont plutôt discriminées, reflétant ainsi le respect des préférences énoncées dans la requête. Il devient donc possible d'obtenir les meilleures k réponses ou celles ayant un score de correspondance supérieur à un seuil donné. Il est donc important de noter qu'une requête flexible n'est pas une simple présentation d'une requête booléenne.

Plusieurs domaines actifs de la recherche exploitent ou utilisent directement les bases de données et les notions d'ensemble flou. En rapport avec nos travaux, nous évoquons ici les exemples de la fouille de données, les Systèmes d'Informations Géographiques (SIG)/données spatiales et la recherche d'information.

De nombreuses approches de fouille de données sont basées sur des algorithmes de génération des règles d'association ("Apriori algorithm") [Han et al., 2011b], qui incluent des approches floues sur les hiérarchies ou les classifications [Beniwal and Arora, 2012] [Muyeba et al., 2008] sur les mesures de soutien/support [Au and Chan, 2005], et sur les transactions [Mangalampalli and Pudi, 2009].

Plusieurs travaux dans le domaine des SIG et des bases de données spatiales ont proposé des modèles de données spatiales utilisant des approches d'ensembles flous [de Caluwe et al., 2013]. Des efforts ont été faits pour utiliser les ensembles flous dans les bases de données spatiales, notamment : dans la définition des relations spatiales [Bai et al., 2013], l'interrogation de l'information spatiale [Vert et al., 2002], et la modélisation orientée objet [Ma, 2005].

La notion de flou a été reconnue dans la recherche d'information depuis longtemps lorsque [Tahani, 1976], puis [Sachs, 1976] ont émis l'idée d'appliquer la théorie des ensembles flous à l'imprécision inhérente à la recherche d'information. L'utilisation de la logique floue pour généraliser les requêtes booléennes, la préservation de l'homomorphisme entre les termes et les requêtes généralisées, et la préservation de la sémantique de la requête ont été développées plus tard [Bezdek et al., 2012]. Les travaux modernes sur la recherche de l'information sont pour la plupart liés à l'imprécision et abordent les questions relatives au traitement du langage naturel, à la détection de similitude, au filtrage (recommandation ou élimination d'information), à la recherche sur le Web, aux données multimédias, etc.

Dans ces différents domaines (fouille de données, SIG, recherche d'information, etc.), généralement caractérisés par l'exploitation de nombreuses sources d'informations, la nécessité de systèmes d'interrogation efficaces se fait de plus en plus sentir. Le but de ces systèmes d'interrogation est d'aider les utilisateurs à trouver l'information recherchée parmi les masses de données disponibles. Pour cela, les utilisateurs doivent formuler leurs besoins sous forme de requêtes. Cependant, la définition des requêtes reste une tâche pénible, due au fait qu'elles doivent être compréhensibles par les systèmes informatiques malgré leur imprécision, et aussi parce qu'il arrive que les utilisateurs ne sachent pas formuler clairement leurs besoins. L'interrogation floue offre aux utilisateurs la possibilité d'exprimer leurs besoins de manière flexible (via des requêtes flexibles), afin de prendre en compte les préférences des utilisateurs dans les résultats.

Notre travail vise à améliorer l'interrogation des métadonnées multi sources et hétérogènes en introduisant les notions de préférences floues.

2.5 Conclusion

L'analyse de l'état de l'art a révélé qu'il reste des lacunes dans la recherche en ce qui concerne le défi que représente le traitement *a posteriori* de grands volumes de données numériques en général et des vidéos issues des systèmes de vidéosurveillance en particulier. Par exemple, il demeure nécessaire d'entreprendre des recherches sur des techniques de réduction et d'interrogation des grands volumes de données, et l'utilisation des informations décrivant le contexte. La plupart des approches de traitement des vidéos existantes ne prennent pas en compte d'autres sources d'informations exogènes, mais se focalisent sur le développement des outils d'analyse des contenus, qui trouvent des limites face aux problèmes liés à l'hétérogénéité des systèmes et la qualité variable des vidéos.

Dans une démarche d'utilisation des informations contextuelles, la modélisation et l'intégration des multiples sources d'informations existantes font partie des verrous à lever. L'hétérogénéité de ces sources d'information et leur utilisation rendent complexe le choix du modèle approprié pour la définition du contexte. Comme le souligne l'état de l'art, un système sensible au contexte doit disposer d'un mécanisme intelligent pour l'interrogation des informations contextuelles.

Chapitre 3

Filtrage négatif via l'exploitation des métadonnées

Une définition générique du filtrage est "l'action de ne laisser passer que certaines choses et pas les autres". Dans le domaine des systèmes d'information (SI) et de la gestion des données, le filtrage peut être défini comme un ensemble de processus permettant, parmi les masses de données disponibles, de fournir à l'utilisateur des données en fonction de ses besoins, sans inclure les données non pertinentes. La constante croissance du volume de données demeure un problème majeur pour les systèmes ou applications qui gèrent des masses importantes de données, en particulier pour les systèmes dont le traitement rapide des données est une priorité. Une solution possible à ce problème consiste à effectuer le filtrage (Figure 3.1) afin de réduire la quantité de données à traiter. Dans ce chapitre, nous proposons une approche de réduction du volume de données appelée **"filtrage négatif"**, basée sur l'exploitation des métadonnées et applicable aux systèmes traitant des grands volumes de données. Nous justifions notre approche par une application aux systèmes de vidésurveillance proposée dans le cadre du projet FILTER2 [1].

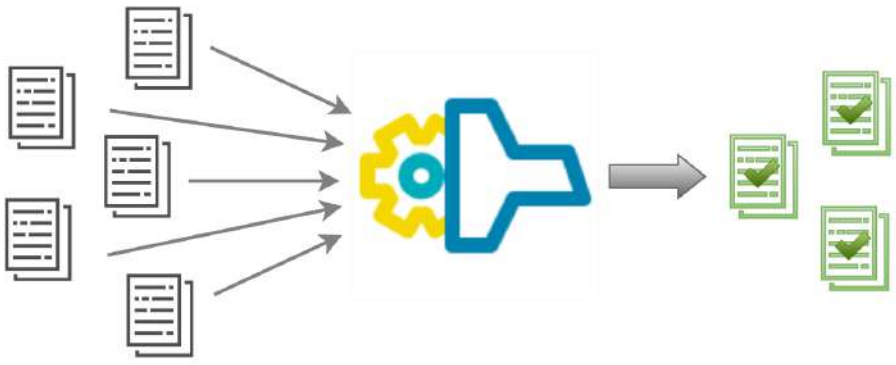


Figure 3.1 – Filtrage de données.

La section 3.1 de ce chapitre présente la définition du filtrage négatif dont un contexte applicatif est présenté à la section 3.2. L'approche de filtrage négatif proposée dans cette thèse est développée dans les sections 3.3 et 3.4.

1. Projet ANR FILTER2 - FILtrage negaTif des contEnus de vidéopRotection

3.1 Définition du filtrage négatif

Nous allons définir la notion de filtrage négatif grâce au contexte applicatif de notre travail qui est l'utilisation de la vidéosurveillance à des fins d'enquête, ensuite nous allons généraliser cette définition.

La vidéosurveillance est devenue une mesure de sécurité très importante dans la prévention du crime [Gill and Spriggs, 2005]. Des études récentes [Fletcher, 2011], [Welsh and Farrington, 2008] illustrent l'importance de la vidéosurveillance qui permet aux propriétaires de magasins, aux chefs d'entreprise et à la police de dissuader et de réagir aux incidents criminels signalés par cette technologie. L'une des principales fonctions de la vidéosurveillance est de stocker des images d'incidents criminels et de comportements antisociaux afin de faciliter l'analyse post-incident pendant les enquêtes [Gill and Spriggs, 2005]. Le processus actuellement utilisé par les enquêteurs (en France), consiste à appliquer des traitements vidéo (détection de visages, détection de véhicules, détection et lecture des plaques d'immatriculation...) sur l'ensemble des séquences vidéos récupérées en amont et préalablement indexées (conversion au bon format, géolocalisation des caméras, gestion des horodatages...). Les traitements sont appliqués sur l'ensemble des vidéos ou uniquement sur un sous-ensemble géographique ou temporel. Après ces traitements et une extraction des données, les enquêteurs effectuent des recherches (véhicules, plaques, personnes, visages...), visionnent les séquences d'intérêt et lancent de nouveaux traitements. Les récentes enquêtes (vol, terrorisme, incivisme, homicide) ont nécessité l'analyse de plusieurs dizaines de téraoctects de données vidéos, correspondant à plusieurs dizaines de milliers d'heures de vidéo. L'utilisation des traitements automatiques dans le cadre de ces enquêtes a permis d'obtenir un gain de temps (encore trop faible dans un contexte opérationnel) d'un facteur 3 en moyenne par rapport à une analyse manuelle. Néanmoins, beaucoup de vidéos étaient inexploitables pour les traitements automatiques (luminosité due à la nuit, conditions optiques, ...). Le temps d'exécution des traitements aurait pu être réduit si les séquences vidéo inadaptées aux traitements automatiques avaient pu être écartées (filtrage négatif). Dans ce contexte, le **filtrage négatif** est donc défini comme un ensemble de processus permettant d'éliminer parmi une masse de vidéos, les séquences qui ne sont pas compatibles aux traitements automatiques donnés. L'objectif est d'améliorer la rapidité d'exécution des traitements en exploitant uniquement les séquences vidéo exploitables. A titre d'exemple, considérons un algorithme de détection et lecture automatique de plaque d'immatriculation qui est capable de lire une plaque d'immatriculation lorsque la valeur (comprise entre 0 et 1) de la qualité d'image est d'au minimum 0.7. En deçà de ce seuil, le traitement ne retourne aucun résultat. Dans l'hypothèse où il est possible de déterminer à l'avance que la qualité d'une séquence vidéo ne peut être supérieure à 0.6, il devient certain que l'analyse de cette séquence

ne donnera aucun résultat. On peut alors exclure cette séquence pour le traitement «détection et lecture des plaques d'immatriculation».

Pour généraliser, le **filtrage négatif** (Figure 3.2) définit des critères de filtrage permettant d'éliminer parmi les masses de données disponibles celles dont on sait au préalable que le traitement n'aboutira à aucun résultat.

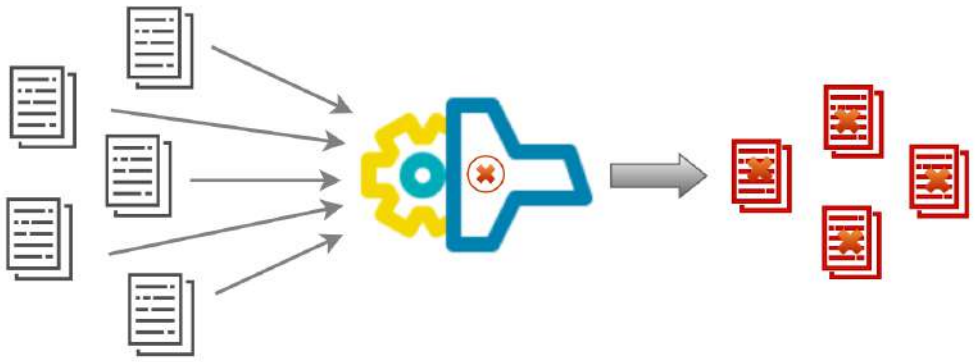


FIGURE 3.2 – Filtrage négatif.

La section suivante présente les besoins exprimés dans le projet FILTER2 permettant de justifier notre approche de filtrage négatif.

3.2 Contexte d'application

FILTER2 pour FILtrage negaTif des contEnus de vidéopRotection, est un projet ANR (https ://anr.fr/Projet-ANR-16-CE39-0013) sur lequel nous travaillons en collaboration avec quatre autres partenaires qui sont : la Police Technique et Scientifique (PTS), Thales Communications & Security SA (spécialiste mondial des systèmes de sécurité en particulier pour la Police Technique et Scientifique), le laboratoire XLIM-SIC de l'Université de Poitiers (qui dispose d'une expertise reconnue dans l'analyse des images couleur et l'évaluation de la qualité vidéo, notamment dans le cadre d'applications comme la vidéoprotection) et le laboratoire GREyC-IRF de l'Université de Caen (qui dispose d'une expertise reconnue dans le domaine de l'analyse des images et de la biométrie et plus spécifiquement dans le domaine de la qualité des images/vidéos dans un cadre applicatif tel que l'authentification des personnes et la vidéoprotection).

Le besoin exprimé dans le projet FILTER2 consiste à améliorer la rapidité des traitements automatiques des vidéos en écartant les séquences vidéo qui ne valident pas leurs pré-requis (traitements automatiques). Donc il s'agit de déterminer si une séquence vidéo est susceptible de donner des résultats par rapport à un traitement donné, auquel cas le traitement sera appliqué à cette séquence. Dans le cas inverse, il est inutile d'exécuter le traitement, s'il est déjà certain qu'aucun résultat n'en découlera. Les traitements retenus dans le cadre du projet sont :(i) la détection de visages, (ii)

la détection des véhicules et (iii) la détection et lecture automatique des plaques. Ces traitements constituent les filtres les plus couramment employés dans les analyses par la PTS. On distingue deux modes d'analyse : urgent et approfondi. Le premier nécessite une analyse très rapide des vidéos. Elle intervient lorsqu'un ou plusieurs individus dangereux et recherchés sont en fuite. Le besoin de résultats immédiats conduit à ne privilégier que les sources de très bonne qualité sur la zone géographique et temporelle où les chances de trouver la cible sont élevées. On préférera écarter des vidéos dont les résultats sont approximatifs par rapport à des vidéos de très bonne qualité susceptibles d'apporter des éléments tangibles à l'enquête. Le deuxième mode permet une analyse approfondie et moins urgente. Il s'agit dans ce cas d'obtenir des résultats plus précis et le plus exhaustif possible. La recherche peut porter sur des séquences de moins bonne qualité, mais dont les traitements sont quand-même susceptibles de donner des résultats.

Le filtrage négatif est basé sur des indices permettant de quantifier la qualité et l'utilisabilité/utilité des vidéos. Il convient alors de définir pour chaque traitement les critères de qualité et d'utilisabilité/utilité vidéo qui permettront de filtrer les séquences vidéo en fonction des deux modes d'analyse (urgent / approfondi). Plus concrètement, il s'agit de développer des métriques reflétant la qualité et l'utilisabilité/utilité des vidéos en se basant sur une combinaison des métadonnées techniques, décrivant le mouvement et le champ de vue de la caméra (ex : vitesse de la caméra, orientation par rapport à des objets qui pourraient obstruer le champ de vue) et issues des algorithmes d'analyse du contenu (ex : décrivant le mouvement ou le nombre de personnes dans la scène). La Figure 3.3 illustre un ensemble de paramètres qui influencent la prise de vue et implicitement la qualité des enregistrements vidéo issus d'un système de vidéoprotection. Cette figure met en évidence des exemples de paramètres liés à l'environnement tels que l'éclairage jour/nuit, l'emplacement intérieur ou extérieur (indoor/outdor) de la scène, les mauvaises conditions météorologiques (forte pluie, neige, brouillard) et les paramètres liés à la caméra (résolution, densité de pixels). Cette figure montre également que les niveaux de qualité requis dépendent des objets recherchés et des applications (inspection, identification, reconnaissance, détection). Par exemple, nous nous intéressons dans cette étude aux applications de détection (véhicules, visages) et de reconnaissance (plaque d'immatriculation) dont certains paramètres pris en compte dans la qualité d'image sont la position de la caméra (exprimée par rapport à un système de référence), la distance de l'objet par rapport à la caméra et la taille de l'objet.

Dans ce projet, les tâches relatives au traitement d'image telles que la définition des métriques de qualité et d'utilisabilité/utilité des vidéos sont réalisées respectivement par les partenaires des laboratoires XLIM-SIC et GREyC-IRF. Les métriques développées sont ensuite mises à disposition sous forme de métadonnées de qualité et d'utilisabilité/utilité vidéo. Notre tâche consiste à intégrer et à modéliser ces mé-

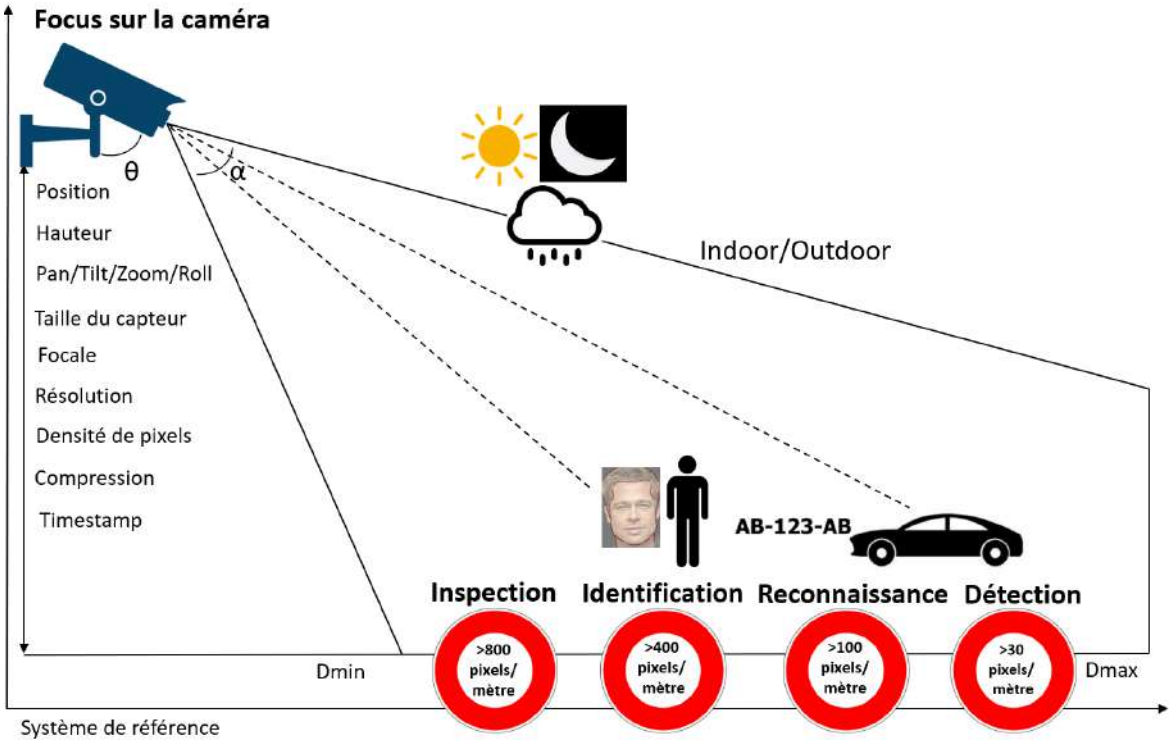


FIGURE 3.3 – Paramètres qui influencent la prise de vue et la qualité de l'image (source : FILTER2).

tadonnées qui peuvent être représentées selon différents niveaux de sémantique et de granularité, tout en assurant la collaboration entre les différents niveaux de métadonnées. Ensuite, de proposer un mécanisme de filtrage négatif basé sur ces métadonnées.

La section suivante présente une modélisation des métadonnées pour le filtrage négatif des vidéos.

3.3 Modélisation des métadonnées pour le filtrage négatif

La valeur de l'information numérique dépend de la facilité avec laquelle elle peut être localisée, recherchée et retrouvée. Les métadonnées décrivant le contenu de l'information numérique sont indispensables pour ces tâches, et sans elles, certaines informations numériques sont considérées comme dénuées de sens, inutilisables ou sans valeur. Par exemple, les bases de données de vidéos pourraient être utilisées simplement comme dépôts ou archives d'images ou de vidéos. Toutefois, la valeur des vidéos contenues dans ces bases de données est grandement accrue si elles peuvent également stocker des métadonnées précises sur les vidéos, de sorte que des requêtes (spatio-temporelles par exemple) puissent ultérieurement être exécutées pour récupérer des vidéos contenant certains types d'objets ou d'évènements. Il est donc facile de comprendre que dans les systèmes de vidéosurveillance, qui gèrent d'énormes volumes de

données recueillies dans des conditions variées et sur de longues périodes de temps, les métadonnées jouent un rôle clé.

Généralement, les métadonnées de vidéosurveillance peuvent être divisées en quatre grandes catégories :

— **Métadonnées descriptives (ou statiques) :** ce sont les données fixes décrivant la caméra de prise de la vidéo, son installation et sa configuration, son emplacement et son orientation, etc. Ces métadonnées peuvent être partagées par plusieurs vidéos issues d'une même caméra.

— **Métadonnées dynamiques :** ce sont des données associées à une ou plusieurs vidéos, produites en permanence et potentiellement évolutives dans le temps, telles que les informations relatives à la géolocalisation, aux évènements, aux alarmes et capteurs divers, etc.

— **Résultats analytiques (métadonnées intrinsèques) :** ce sont des métadonnées issues des algorithmes d'analyse de contenus, notamment des détections et des caractéristiques, associés à des niveaux de confiance estimés.

— **Annotations et commentaires :** généralement du texte.

Une description détaillée des métadonnées descriptives ou statiques et des métadonnées dynamiques est présentée dans [Codreanu, 2015]. Les propositions faites à ce niveau concernent les métadonnées issues des algorithmes d'analyse automatique des vidéos, qui sont encore appelées *métadonnées intrinsèques*, car elles sont directement liées aux informations de contenu des vidéos. Les métadonnées intrinsèques peuvent être divisées en *métadonnées structurelles* (traitées dans ce chapitre) et *métadonnées sémantiques* (traitées dans le chapitre suivant).

Les métadonnées structurelles donnent des caractéristiques descriptives des images en tant que représentation numérique (et non le contenu visuel). Leur création implique le calcul des caractéristiques de bas niveau qui encapsulent de façon concise l'apparence des images (frames) vidéo en termes numériques simples, obtenus en analysant la couleur, la forme, la texture, la structure et le mouvement qui les composent. Comme exemple de métadonnées structurelles, nous avons la composition des couleurs d'une image qui peut être représentée par un vecteur de dimension n. Les métadonnées structurelles ont trois applications principales [Gupta and Jain, 1997] : *(i)* pour la segmentation automatique de la vidéo et la reconnaissance des changements de scène, déterminées par des changements brusques dans l'apparence visuelle ; *(ii)* pour créer un index précis à l'image (frame) près, offrant un accès non linéaire à n'importe quel segment de la vidéo ; *(iii)* et surtout du point de vue de l'interrogation par contenu, pour permettre une recherche floue rapide, précise et efficace d'images ou de séquences vidéo.

Les métadonnées sémantiques se situent au niveau des connaissances géné-

rales et résultent de l'analyse des propriétés des objets et évènements contenus dans les vidéos et visibles ou reconnaissable aux utilisateurs de la vidéo. Les métadonnées sémantiques ont une très grande valeur informative, car elles se rapportent directement aux caractéristiques d'une vidéo qui sont d'une pertinence immédiate pour la compréhension humaine du contenu vidéo. Par exemple, les métadonnées décrivant le mouvement d'un objet (véhicule/personne) dans la vidéo.

Les métadonnées structurelles modélisées dans ce chapitre sont liées à la qualité et à l'utilisabilité/utilité des vidéos. L'objectif de la modélisation est d'intégrer et d'unifier toutes ces métadonnées selon différents niveaux de sémantique et de granularité (frame, segment, vidéo) afin de parvenir à un filtrage négatif basé sur la qualité et l'utilisabilité/utilité des vidéos. La qualité d'une image peut désigner soit le degré/niveau de précision de l'image (vue comme un ensemble de signaux) lors de l'acquisition, du traitement, du stockage et de la restitution, soit un ensemble d'attributs visuellement significatifs de l'image. L'utilisabilité/utilité de la vidéo quant à elle désigne un ensemble de caractéristiques qui déterminent l'aptitude d'une vidéo à être utilisée dans une situation donnée.

3.3.1 Métadonnées liées à la qualité de la vidéo

La qualité d'une image dépend de l'optique du capteur, de l'électronique (amplification, quantification et échantillonnage), ainsi que de l'environnement dans lequel l'image a été capturée et des conditions d'éclairage. Afin d'évaluer la qualité d'une vidéo, il est nécessaire de définir un ensemble de métriques de qualité en utilisant ou en développant des descripteurs spatiaux de qualité d'images de manière à caractériser les dégradations visibles et prépondérantes dans le rendu visuel d'une image. La définition des métriques de qualité est généralement basée sur deux types approches [Saad et al., 2010] : (i) les approches de mesure de la qualité avec référence dont l'objectif est d'évaluer le degré de conformité de l'image cible (dégradée) par rapport à une image originale ou de référence, et (ii) les approches de mesure de la qualité sans référence basées sur l'apprentissage statistique, sur les statistiques des scènes naturelles, ou sur des distorsions spécifiques. Les métadonnées de qualité vidéo exploitables dans ce contexte d'application sont relatives aux approches de mesure sans référence. Selon ces approches, les métadonnées de qualité vidéo sont fournies sous forme de vecteurs de caractéristiques extraits des images.

Des exemples des métriques sans références G-BLIINDS2 et BIQI [Charrier, 2011] ont été développées par nos partenaires dans le projet FILTER2. Comme présenté à la Figure 3.4, ces métriques sont utilisées pour évaluer la qualité de trois images extraites de la base TID2008 [Ponomarenko et al., 2008], et dégradées de gauche à droite respectivement par une compression JPEG, une erreur de transmission JPEG2000 et

un bruit du à une insertion de blocs de couleur et d'intensité différente. Les valeurs MOS (score moyen des observateurs ou « Mean Opinion Score ») sont obtenues par les observateurs humains et représentent la vérité terrain de l'évaluation de la qualité d'image. Ces valeurs sont calculées sur une échelle de 0 (très mauvaise qualité) à 9 (qualité excellente).

G-BLINDS2 = 0,786 BIQI = 0,756 MOS =3,65	G-BLINDS2 = 0,662 BIQI = 0,714 MOS =2,67	G-BLINDS2 = 0,803 BIQI = 0,759 MOS =3,69

FIGURE 3.4 – Exemples de scores de qualité d'image obtenus grâce aux métriques sans référence G-BLINDS2 et BIQI.

3.3.2 Métadonnées liées à l'utilisabilité/utilité de la vidéo

L'utilisabilité/utilité des contenus vidéo peut être définie en se basant sur des caractéristiques liées aux informations bas-niveau de l'image. Les caractéristiques liées aux différents défauts d'acquisition comme le flou, l'éblouissement, le bruit de capture, etc. sont des paramètres pris en compte dans la définition des critères d'utilisabilité/utilité de la vidéo.

L'utilisabilité/utilité de la vidéo tient compte de l'habilitée à détecter, reconnaître ou identifier les objets dans les vidéos. L'exploitation du "*critère de Johnson*" est une base pour la définition des métriques permettant d'évaluer l'utilisabilité/utilité des vidéos. Johnson a défini des seuils, connus sous le nom de "*critères Johnson*", comme étant les résolutions effectives pour détecter, reconnaître ou identifier les cibles capturées par les caméras. Rappelons que *détecter* c'est la capacité à distinguer un objet de l'arrière-plan, *reconnaître* c'est la capacité à classer les objets (personnes, véhicule, etc.), *identifier* c'est la capacité à décrire l'objet en détail (personne avec un chapeau, lecture d'une plaque, etc.). Les seuils définis par le critère de Johnson peuvent être influencés par des facteurs tels que le champ de vision, la résolution spatiale, l'occultation de la scène, etc. Les partenaires du projet FILTER2 ont proposé de faire une évaluation subjective de détection, reconnaissance et identification, ensuite d'introduire les scores obtenus dans un algorithme d'apprentissage automatique afin de proposer des métadonnées d'utilisabilité/utilité des vidéos.

3.3.3 Proposition d'un modèle de métadonnées pour la qualité et d'utilisabilité/utilité des vidéos

La Figure 3.5 représente un modèle générique des métadonnées de qualité et d'utilisabilité/utilité vidéo proposé pour le filtrage négatif des contenus de vidéoprotection. Ce modèle de métadonnées met en évidence l'ensemble des entités prises en compte dans le système modélisé, ainsi que les relations entre les différentes entités.

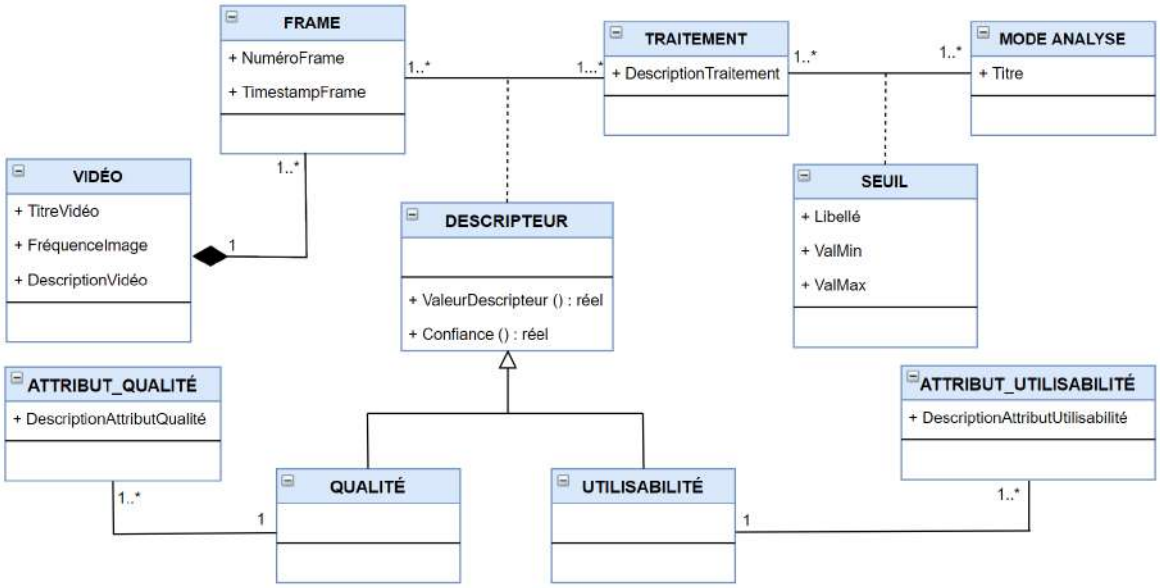


FIGURE 3.5 – Modèle générique pour les métadonnées de qualité et d'utilisabilité/utilité des vidéos.

La définition des classes **VIDÉO** et **FRAME** est essentielle pour la définition ultérieure des autres classes. Dans l'analyse vidéo conventionnelle, les vidéos sont divisées en scènes, chacune se rapportant à un aspect différent de l'ensemble de la vidéo, et les scènes sont subdivisées en plans, chacun d'entre eux étant une seule série contiguë de frames dérivées d'une prise de vue. Les caractéristiques des vidéos sont calculées par frame ou groupe de frame, d'où une division de la vidéo directement en frames sans avoir à définir des segments vidéo. Les caractéristiques vidéo pour chaque frame sont représentées par la classe **DESCRIPTEUR** et chaque descripteur est lié à un traitement spécifique (classe **TRAITEMENT**). Les fonctions ***getFeatureValue()*** et ***Confidence()*** sont respectivement utilisées pour calculer la valeur globale et la confiance globale en se basant sur les attributs des métadonnées de qualité (classe **QUALITÉ**) ou d'utilisabilité (classe **UTILISABILITÉ**) des vidéos. De nouveaux attributs (**ATTRIBUT_QUALITÉ, ATTRIBUT_UTILISABILITÉ**) peuvent être définis à tout moment pour les métadonnées. Des seuils (classe **SEUIL**) sont définis pour chaque mode d'analyse (classe **MODE ANALYSE**), afin de déterminer la compatibilité d'un traitement au mode d'analyse choisi. Bien que les métadonnées

requises pour les différents types d'analyse vidéo puissent varier, le modèle de métadonnées proposé est conçu pour être générique, permettant d'intégrer facilement de nouvelles métadonnées (héritant de **DESCRIPTEUR**), de sorte que de nouveaux besoins puissent être pris en compte.

Ce modèle générique de métadonnées est ensuite utilisé dans le mécanisme de filtrage négatif que nous définissons dans la section suivante.

3.4 Mécanisme de filtrage

Cette section présente le mécanisme proposé pour la mise en œuvre du filtrage négatif des contenus vidéo.

Le filtrage négatif constitue un module de pré-analyse qui viendra s'intégrer en amont dans le processus global d'analyse massive. A l'issue de ses calculs, le module de filtrage négatif fournit deux types d'information :

— Analyse urgente : le résultat du filtrage indiquera pour chaque segment vidéo et pour chaque traitement si une analyse dans l'urgence est pertinente ou non. Le résultat peut être affiché, par exemple, à l'aide d'un code de couleur : vert si la séquence est compatible avec le traitement en mode analyse urgente, rouge si non.

— Analyse approfondie : le résultat du filtrage indiquera pour chaque séquence vidéo et chaque traitement un score de compatibilité avec l'analyse approfondie. Le résultat peut être présenté par exemple sous la forme d'un code de couleur à trois niveaux en fonction des seuils préalablement définis. Le vert peut être défini pour une parfaite compatibilité, orange pour une compatibilité moyenne et rouge pour une incompatibilité. L'utilisateur choisira alors de traiter les segments en orange ou non en fonction de ses besoins et de ses ressources en temps.

Exemple : la Figure 3.6 présente les résultats du filtrage négatif pour un traitement donné (ex : détection des véhicules) sur la vidéo "*fichier_001.mp4*" : le code de couleur montre que le traitement sélectionné peut s'appliquer aux segments vidéo "U_2" et "U_4" dans le mode d'analyse urgent, ainsi qu'aux segments vidéo "A_2", "A_4", et "A_6" dans le mode d'analyse approfondi. En fonction de ses besoins et de ses ressources en temps, l'enquêteur peut analyser le segment vidéo "A_3".

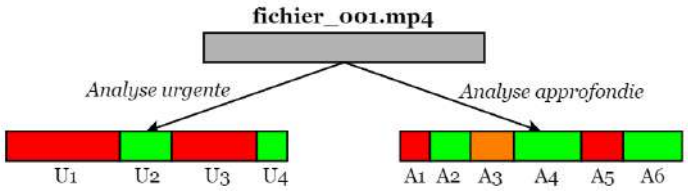


FIGURE 3.6 – Formalisme possible pour les résultats du filtrage négatif.

Les segments en rouge ("U_1", "U_3", "A_1" et "A_5") sont affichés à titre indicatif, ils ne font pas partie des résultats et sont à éliminer.

3.4.1 Définition des données

Définition 1 : un segment vidéo u est une séquence de frames successives f_1, f_2, ..., f_i appartenant à une vidéo v, *debSeg(u)* donne la première frame du segment et $finSeg(u)$ donne la dernière frame du segment. Le segment vidéo $u \in v$ est défini par $u = [f_{start}, f_{end}]$ où f_{start} représente la frame marquant le début du segment et f_{end} la frame marquant la fin du segment. Donc pour le segment vidéo u, $debSeg(u)$ et $finSeg(u)$ retournent respectivement f_{start} et f_{end} ;

Définition 2 : une frame est compatible avec un traitement donné pour un mode d'analyse donné si la valeur globale des descripteurs associées est comprise dans un intervalle s appelé seuil de compatibilité. Le seuil s est défini par $s = [ValMin, ValMax]$, avec $0 \leq ValMin < ValMax \leq 1$.

Définition 3 : Un traitement t désigne un algorithme d'analyse automatique de vidéo et possède un seuil de compatibilité pour chaque mode d'analyse des vidéos. Le traitement t est défini par $t = \{s(m)\}$ où chaque s représente le seuil de compatibilité du mode d'analyse m.

3.4.2 Algorithmes de filtrage

Le filtrage négatif proposé s'appuie sur la modélisation des métadonnées présentée à la section 3.3.3. L'objectif est de définir des algorithmes de requêtage basés métadonnées afin d'écarter de manière automatique les plages inexploitables des vidéos selon des critères de qualité et d'utilité/utilisabilité vidéo. Les algorithmes de filtrage ont pour paramètres une liste de vidéo, une liste de traitement et des seuils paramétrables. Les seuils sont définis pour chaque mode d'analyse (urgent ou approfondi) afin de déterminer la compatibilité des séquences vidéo aux différents traitements (détection de visages, détection de véhicules, détection et lecture automatique des plaques d'immatriculation). La compatibilité à un mode d'analyse est déterminée en comparant la valeur globale des descripteurs de chaque frame (ou groupe de frame) de la vidéo aux seuils de compatibilité définis pour les différents traitements. Ensuite, les résultats des comparaisons sont utilisés pour constituer (regroupement des frames) des segments vidéo.

Étant donné un ensemble de vidéos $V = \{v_1, v_2, ..., v_i\}$ (chaque vidéo v_i composée d'un ensemble de frames $F_i = \{f_1^i, f_2^i, ..., f_n^i\}$) et un ensemble de traitement $T = (t_1, t_2, ..., t_j)$, le résultat du filtrage négatif pour chaque mode d'analyse est un ensemble de triplets : $R = \{r = (t_j, v_i, [f_{start}^i, f_{end}^i])\}$, avec $t_j \in T, v_i \in V$, et $f_{start}^i, f_{end}^i \in F_i | f_{start}^i \leq f_{end}^i$. Pour le mode d'analyse **urgent**,chaque $[f_{start}^i, f_{end}^i]$ est un segment

vidéo compatible au traitement t_j. Pour le mode d'analyse **approfondi**, on a des segments vidéo $[f^i_{start}, f^i_{end}]$ compatibles aux traitements t_j et des segments vidéo $[f^i_{start}, f^i_{end}]$ moyennement compatibles (optionnels) aux traitements t_j.

Les algorithmes 1 et 2 permettent d'obtenir les résultats du filtrage négatif pour les deux modes d'analyse. Pour ces deux algorithmes, la fonction $getVideoFrames(v_i)$ récupère dans une liste l'ensemble des frames de la vidéo v_i, et la fonction *getFeatureValue*(f_k, t_j) récupère pour une frame f_k de cette liste la valeur globale des descripteurs correspondants au traitement t_j. Cette valeur globale des descripteurs est ensuite comparée aux différents seuils définis pour chaque mode d'analyse afin de déterminer la compatibilité de la frame pour le traitement. Les algorithmes s'exécutent en deux grandes étapes qui sont le filtrage proprement dit par frame et la composition des segments.

L'étape de filtrage par frame : est celle qui consiste à comparer chaque frame d'une vidéo aux différents seuils définis afin de déterminer son éligibilité à un traitement donné en fonction d'un mode d'analyse.

L'étape de composition de segments vidéo : un segment vidéo étant défini comme une suite consécutive de frames, cette étape regroupe les frames d'une vidéo de manière chronologique et selon l'indexation (éligibilité à un traitement pour un mode d'analyse) faite à l'étape de filtrage, afin de constituer des segments vidéo indexés, c'est-à-dire éligibles ou non à un traitement donné selon un mode d'analyse choisi.

Un exemple concret de filtrage négatif effectué à l'aide de ces algorithmes est présenté à la section 3.4.3.

3.4.2.1 Algorithme de filtrage pour le mode urgent

L'algorithme 1 prend en entrées : un ensemble de traitements et un ensemble de vidéos, et retourne comme résultat pour chaque traitement, un ensemble de segments vidéo compatibles au mode d'analyse urgent.

3.4.2.2 Algorithme de filtrage pour le mode approfondi

L'algorithme 2 prend en entrées : un ensemble de traitements et un ensemble de vidéos, et retourne comme résultat pour chaque traitement, un ensemble de segments vidéo compatibles et un ensemble de segments vidéo moyennement compatibles (ou optionnels) au mode d'analyse approfondi.

Algorithm 1: Negative filtering algorithm for urgent analysis

Input: a set of processing tasks : T and a set of videos : V
Output: a list of compatible video segments per processing tasks : $urgentResult$

```
1  foreach t_j in T do
2      foreach v_i in V do
3          frameList ← getVideoFrames(v_i);
4          k = 0;
5          while k < size(frameList) do
6              notUrgent ← true;
7              while notUrgent and k < size(frameList) do
8                  val ← getFeatureValue(frameList.get(k), t_j);
9                  if val ≥ t_j.s(u).ValMin and val ≤ t_j.s(u).ValMax then
10                     notUrgent ← false;
11                 else
12                     k++;
13                 end if
14             end while
15             urgent ← true;
16             while urgent and k < size(frameList) do
17                 val ← getFeatureValue(frameList.get(k), t_j);
18                 if val ≥ t_j.s(u).ValMin and val ≤ t_j.s(u).ValMax then
19                     add(urgentSegment, frameList.get(k));
20                     k++;
21                 else
22                     urgent ← false;
23                 end if
24             end while
25             if urgentSegment is not empty then
26                 f^i_start ← debSeg(urgentSegment);
27                 f^i_end ← finSeg(urgentSegment);
28                 add(urgentResult, t_j, v_i, [f^i_start, f^i_end]);
29                 clear(urgentSegment);
30             end if
31         end while
32     end foreach
33 end foreach
```

Algorithm 2: Negative filtering algorithm for in-depth analysis

Input: a set of processing tasks : T and a set of videos : V
Output: a list of compatible and optional video segments per processing tasks : $indepthResult, optionalResult$

```
1  foreach t_j in T do
2      foreach v_i in V do
3          frameList ← getVideoFrames(v_i);
4          k = 0;
5          while k < size(frameList) do
6              notCompatible ← true;
7              while notCompatible and k < size(frameList) do
8                  val ← getFeatureValue(frameList.get(k), t_j);
9                  if val ≥ t_j.s(a).ValMin and val ≤ t_j.s(a).ValMax then
10                     notCompatible ← false;
11                 else if val ≥ t_j.s(o).ValMin and val ≤ t_j.s(o).ValMax then
12                     notCompatible ← false;
13                 else
14                     k++;
15                 end if
16             end while
17             indepth ← true;
18             while indepth and k < size(frameList) do
19                 val ← getFeatureValue(frameList.get(k), t_j);
20                 if val ≥ t_j.s(a).ValMin and val ≤ t_j.s(a).ValMax then
21                     add(indepthSegment, frameList.get(k));
22                     k++;
23                 else
24                     indepth ← false;
25                 end if
26             end while
27             if indepthSegment is not empty then
28                 f^i_start ← debSeg(indepthSegment);
29                 f^i_end ← finSeg(indepthSegment);
30                 add(indepthResult, t_j, v_i, [f^i_start, f^i_end]);
31                 clear(indepthSegment);
32             end if
33             optional ← true;
34             while optional and k < size(frameList) do
35                 val ← getFeatureValue(frameList.get(k), t_j);
36                 if val ≥ t_j.s(o).ValMin and val ≤ t_j.s(o).ValMax then
37                     add(optionalSegment, frameList.get(k));
38                     k++;
39                 else
40                     optional ← false;
41                 end if
42             end while
43             if optionalSegment is not empty then
44                 f^i_start ← debSeg(optionalSegment);
45                 f^i_end ← finSeg(optionalSegment);
46                 add(optionalResult, t_j, v_i, [f^i_start, f^i_end]);
47                 clear(optionalSegment);
48             end if
49         end while
50     end foreach
51 end foreach
```

3.4.3 Exemple de filtrage

Supposons comme entrées de nos algorithmes :

— la vidéo "*vidéo_001.mp4*" illustrée à la figure 3.7, qui est composée de 20 frames $(f_1, f_2, f_3, ..., f_{20})$;

— les traitements $t_1, t_2,$ et t_3 dont les seuils (paramétrables) de compatibilité pour chaque mode d'analyse sont définis dans le tableau de la Figure 3.8.

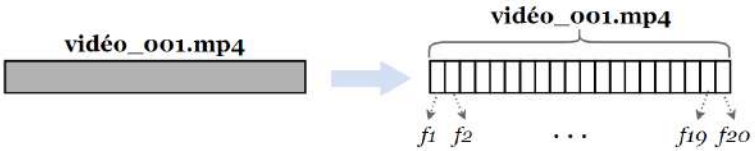


FIGURE 3.7 – Exemple de vidéo en entrée pour le filtrage négatif.

MODE	TRAITEMENTS		
	Traitement 1 (t1)	*Traitement 2 (t2)*	*Traitement 3 (t3)*
Urgent	1 / 0,85 / 0	1 / 0,75 / 0	1 / 0,65 / 0
Approfondi	1 / 0,5 / 0,15 / 0	1 / 0,65 / 0,35 / 0	1 / 0,5 / 0,25 / 0

Seuil compatibilité
Seuil optionnel
Seuil incompatibilité

FIGURE 3.8 – Exemples de seuils de compatibilité aux différents traitements.

Les valeurs globales des descripteurs pour chaque frame de la vidéo et chaque traitement sont présentées dans le tableau 3.1. Ces valeurs proviennent de la base de données conçue grâce au modèle de métadonnées décrit à la section 3.3.3.

3.4.3.1 Exemple d'analyse urgente

Les 2 étapes d'exécution de l'algorithme 1 sont présentées à la Figure 3.9. A l'étape 1, la valeur globale associée à chaque frame pour un traitement est récupérée et comparée au seuil défini pour le traitement.

Exemple : pour le mode urgent, le traitement t_1 est applicable à une frame si la valeur globale des descripteurs de la frame appartient à l'intervalle [0.85, 1] (voir Figure

TABLE 3.1 – Valeurs globales des descripteurs pour chaque traitement.

	TRAITEMENTS		
FRAMES	traitement t_1	traitement t_2	traitement t_3
f_1	0.71	0.76	0.67
f_2	0.68	0.80	0.71
f_3	0.74	0.83	0.74
f_4	0.11	0.79	0.68
f_5	0.12	0.32	0.76
f_6	0.14	0.29	0.82
f_7	0.13	0.30	0.74
f_8	0.87	0.24	0.91
f_9	0.91	0.41	0.89
f_{10}	0.94	0.52	0.21
f_{11}	0.98	0.49	0.23
f_{12}	0.16	0.61	0.19
f_{13}	0.18	0.83	0.24
f_{14}	0.14	0.78	0.53
f_{15}	0.19	0.84	0.61
f_{16}	0.25	0.90	0.51
f_{17}	0.35	0.87	0.55
f_{18}	0.56	0.79	0.62
f_{19}	0.67	0.87	0.59
f_{20}	0.78	0.93	0.64

3.8). La valeur globale des descripteurs de la frame f_1 pour le traitement t_1 est 0.71 (voir tableau 3.1), donc le traitement t_1 n'est pas applicable à la frame f_1 en mode urgent. C'est pourquoi sur la Figure 3.9, la frame f_1 est représentée en couleur rouge pour le traitement t_1. Par contre, le traitement t_2 est applicable à la frame f_1 en mode urgent, car la valeur globale des descripteurs de f_1 pour ce traitement ($valeur = 0.76$) appartient au seuil défini (intervalle $[0.75, 1]$) pour la compatibilité en mode urgent. Cela se justifie sur la Figure 3.9 par la représentation de la frame f_1 en couleur verte pour le traitement t_2.

A l'étape 2, les frames sont regroupées pour former des segments vidéo, tout en distinguant pour chaque traitement les segments vidéo compatibles ou non au mode d'analyse urgent. Par exemple, le traitement t_2 peut s'appliquer aux segments vidéo V_1 (composé des frames f_1, f_2, f_3, f_4) et V_3 (composé des frames f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}) pour une analyse urgente de la vidéo "*vidéo_oo1.mp4*".

Le résultat du filtrage négatif en mode urgent pour notre exemple est l'ensemble : {(t_1, *vidéo_001.mp4*, [f_8, f_{11}]), (t_2, *vidéo_001.mp4*, [f_1, f_4]), (t_2, *vidéo_001.mp4*, [f_{13}, f_{20}]), (t_3, *vidéo_001.mp4*, [f_1, f_9])}.

Rappel : les segments en rouge (affichés à titre indicatif) ne font pas partie des résultats, ils sont à éliminer.

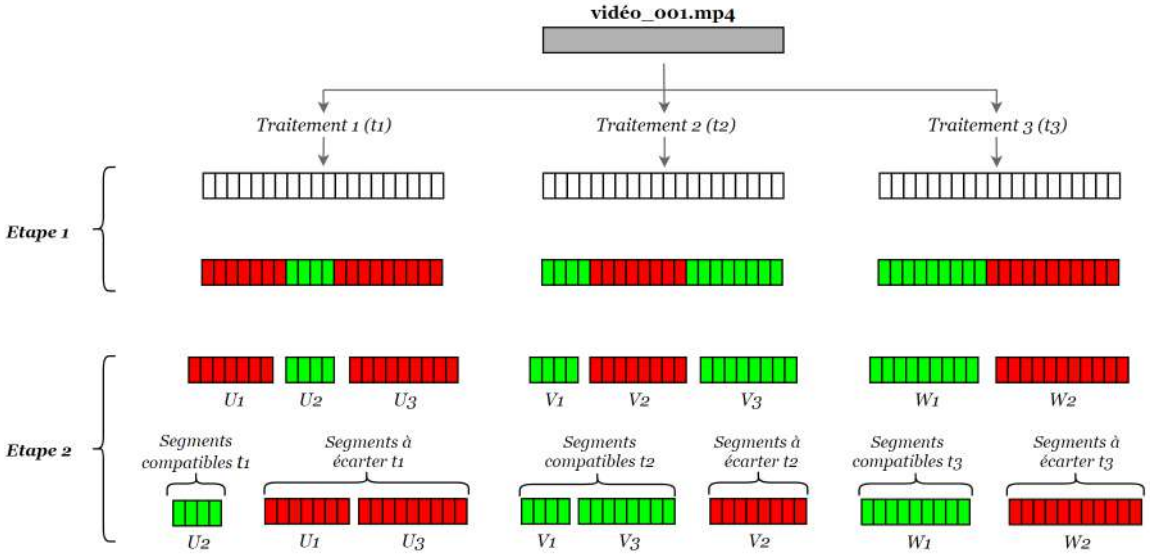


FIGURE 3.9 – Exemple de filtrage négatif en mode urgent.

3.4.3.2 Exemple d'analyse approfondie

La Figure 3.10 présente les 2 étapes d'exécution de l'algorithme 2. A l'étape 1 de cet algorithme, un nouveau seuil est pris en compte dans la comparaison de la valeur globale des descripteurs associée à chaque frame pour un traitement donné. Il s'agit du seuil de compatibilité moyenne. A l'étape 1 du mode d'analyse urgent, une frame pouvait avoir soit l'état "compatible" (représenté en vert), soit l'état "incompatible" (représenté en rouge) à un traitement donné. Avec le mode d'analyse approfondi,un nouvel état "moyennement compatible" ou "optionnel" (représenté en orange) est pris en compte.

Exemple : pour le mode d'analyse approfondi, le traitement t_1 est optionnel pour une frame si la valeur globale des descripteurs de la frame appartient à l'intervalle $[0.15, 0.5]$ (voir Figure 3.8). La valeur globale des descripteurs de la frame f_{12} pour le traitement t_1 est 0.16 (voir tableau 3.1), donc le traitement t_1 est optionnel pour la frame f_{12} dans le mode d'analyse approfondi. Par conséquent, sur la Figure 3.10, la frame f_{12} est représentée en couleur orange pour le traitement t_1.

A l'étape 2, des segments vidéo moyennement compatibles à un traitement peuvent être constitués. Par exemple, le segment vidéo U_4 (composé des frames f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}) est moyennement compatible au traitement t_1 pour une analyse approfondie de la vidéo "*vidéo_oo1.mp4*".

Le résultat du filtrage négatif en mode approfondi pour notre exemple donne 2 grands ensembles :

— l'ensemble constitué des segments vidéo compatibles à l'analyse approfondie : $\{(t_1, \textit{vidéo\_001.mp4}, [f_1, f_3]), (t_1, \textit{vidéo\_001.mp4}, [f_8, f_{11}]), (t_1, \textit{vidéo\_001.mp4}, [f_{18}, f_{20}]), (t_2, \textit{vidéo\_001.mp4}, [f_1, f_4]), (t_2, \textit{vidéo\_001.mp4}, [f_{13}, f_{20}]), (t_3, \textit{vi-}$

déo_001.mp4, $[f_1, f_9]$), (t_3, *vidéo_001.mp4*, $[f_{14}, f_{20}]$)}.

— l'ensemble constitué des segments vidéo moyennement compatibles à l'analyse approfondie : {(t_1, *vidéo_001.mp4*, $[f_{12}, f_{17}]$), (t_2, *vidéo_001.mp4*, $[f_9, f_{12}]$)}.

Rappel : les segments en rouge (affichés à titre indicatif) ne font pas partie des résultats, ils sont à éliminer.

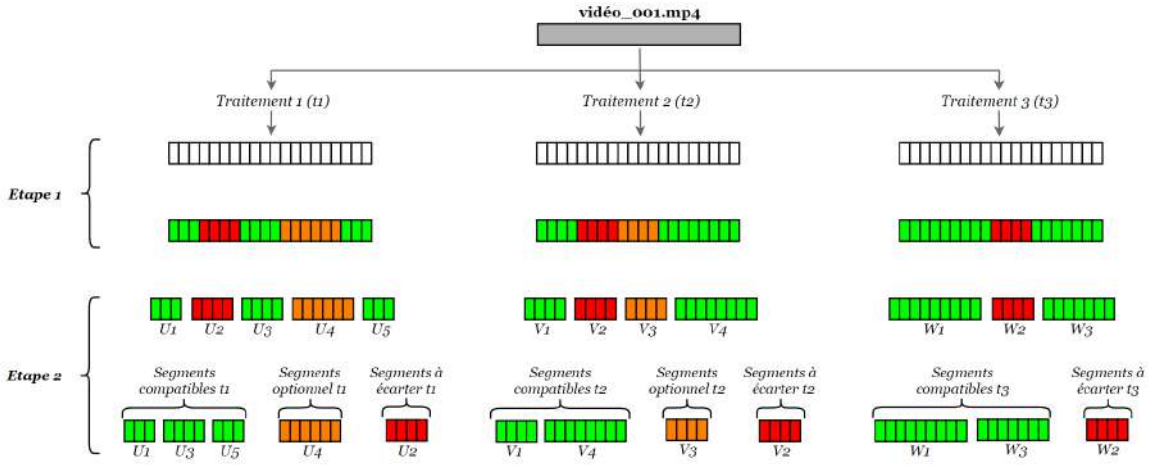


FIGURE 3.10 – Exemple de filtrage négatif en mode approfondi.

3.5 Conclusion

Dans ce chapitre, nous avons proposé la notion de filtrage négatif basé sur les métadonnées Un exemple d'application aux systèmes de vidéosurveillance a été développé pour illustrer l'intérêt d'un tel filtrage. La modélisation des métadonnées est une étape clé pour la mise en œuvre du filtrage négatif, car elle permet de décrire et structurer les métadonnées utiles pour le système, afin de permettre l'interopérabilité de ces métadonnées et de faciliter leur utilisation. Ainsi, nous avons proposé un modèle générique intégrant les métadonnées de qualité et d'utilisabilité/utilité des vidéos issues des systèmes de vidéosurveillance. Le mécanisme de filtrage proposé s'appuie sur le modèle générique de métadonnées afin de réduire les volumes de vidéos à traiter sur des critères de qualité et d'utilisabilité/utilité.

Bien que la modélisation des métadonnées et le mécanisme afférent de filtrage négatif ne soient ici appliqués qu'au cadre de la vidéosurveillance, ils peuvent être adaptés ou reconstruits de la même manière dans un nouveau contexte.

Chapitre 4

Enrichissement contextuel

Comme présenté dans le chapitre précédent, le *filtrage négatif* permet d'éliminer parmi les vidéos provenant de différents systèmes de vidéosurveillance, les séquences vidéo dont les contenus ne satisfont pas un ensemble de critères (ex : qualité et utilisabilité/utilité) définis et indispensables au traitement *a posteriori* (ex : dans la cadre d'une enquête) des dites vidéos. Le filtrage négatif peut donc contribuer à l'essor de la vidéosurveillance intelligente grâce au pré-filtrage. Cependant, demeurent les problèmes liés au caractère incomplet et à l'interprétation des données de vidéosurveillance. L'état de l'art a montré le besoin d'exploiter d'autres sources d'informations afin de faciliter le traitement des données de vidéosurveillance. Les informations contextuelles constituent de nos jours des sources enrichissantes d'informations dans différents domaines et applications. Les solutions actuelles visant au développement des systèmes de vidéosurveillance intelligents sont majoritairement basées sur l'extraction des informations des contenus vidéo, et intègrent peu ou presque pas les informations (contextuelles) externes aux vidéos. Dans ce chapitre, nous proposons une approche d'*enrichissement contextuel* permettant une interrogation "intelligente" des données. L'utilisation de cette approche dans le cadre applicatif (vidéosurveillance) de cette thèse vise à améliorer le processus d'interrogation des grands volumes de vidéos.

La section 4.1 de ce chapitre donne une définition de l'enrichissement contextuel. Des étapes génériques pour la mise en œuvre du processus d'enrichissement contextuel sont présentées dans 4.2. Une application des différentes étapes de l'enrichissement contextuel au cadre de la vidéosurveillance est développée dans 4.3.

4.1 Définition de l'enrichissement contextuel

Le contexte est un concept au sens large, qui n'a pas de définition unique pouvant couvrir tous les aspects auxquels il fait référence. Nous définissons le contexte ou l'information contextuelle dans cette étude comme toute information (implicite ou explicite) qui peut être utilisée pour décrire ou caractériser une entité (personne, objet physique/informatique, tâche, concept, etc.) dans une situation donnée (spatiale et temporelle). Les informations contextuelles peuvent être utilisées pour réduire efficacement les raisonnements nécessaires (par filtrage, agrégation et inférence) à l'interprétation d'un fait, d'une action ou à la prise de décision dans une application spécifique.

La notion *d'enrichissement contextuel* proposée dans ce chapitre est définie comme un processus consistant à identifier et à modéliser les informations contextuelles pertinentes pour une entité d'un système, et appliquer des opérations contextuelles afin de répondre à certaines exigences fonctionnelles ou opérationnelles de ce système. Nous définissons trois principales opérations contextuelles pouvant être mises en œuvre dans le processus d'enrichissement contextuel : le filtrage contextuel (qui est traité dans cette thèse), l'agrégation contextuelle et l'inférence contextuelle.

Définition 1 : *le filtrage contextuel* filtre les données en fonction d'un contexte donné. Par exemple, les informations relatives à la météo et à l'environnement peuvent permettre d'exclure des traitements ultérieurs ou des requêtes, les séquences vidéo dans lesquelles la visibilité est dégradée par la pluie, les nuages, la luminosité ou la nuit.

Définition 2 : *l'agrégation contextuelle* combine des données en fonction des similarités contextuelles et de la pertinence. Par exemple, l'agrégation des données provenant des réseaux sociaux (images, vidéos, etc.) et des données géolocalisées peuvent permettre aux opérateurs de vidéosurveillance de délimiter une zone d'intérêt et s'intéresser uniquement aux séquences vidéo prises dans cette zone d'intérêt (temporelle ou spatiale).

Définition 3 : *l'inférence contextuelle* est le processus de déduction des nouvelles connaissances à partir du contexte. Par exemple, les données de géolocalisation et les données provenant des applications de mobilité peuvent permettre de localiser un objet cible (véhicule, personne, équipement) à un instant t_i, ensuite, d'utiliser ces informations pour exclure la possibilité que cet objet se retrouve à un autre endroit précis à l'instant t_j.

Les informations contextuelles peuvent être considérées comme un ensemble de métadonnées provenant de différentes sources. Les formats, les types et la représentation des données peuvent varier d'une source à l'autre, ce qui rend les données hétérogènes et constitue une limite pour le croisement et l'interrogation de ces données. L'enrichissement contextuel est un processus complexe, car il doit prendre en compte les relations spatio-temporelles entre les vidéos et les informations de contexte. Par exemple, on veut vérifier si des enregistrements vidéo d'une période ont été faits dans des conditions météorologiques (ex : pollution) empêchant toute visibilité, cela implique la prise en compte des relations spatiales et temporelles entre les caméras (ex : leurs positions) qui ont généré les vidéos et les informations météorologiques (lieu, temps, météo) qui sont des informations contextuelles (disponibles en tant que données libres ou Open Data). Autrement dit, un raisonnement spatio-temporel est nécessaire pour répondre à une telle requête. L'enrichissement contextuel pose donc un problème d'intégration des (méta)données hétérogènes avec une capacité de raisonnement spatio-temporel. De plus, l'enrichissement contextuel se doit d'être un processus évolutif afin de faciliter l'intégration de nouvelles sources d'informations contextuelles.

La section suivante décrit une démarche générique permettant de mettre en œuvre l'enrichissement contextuel. Cette démarche sera adaptée à notre domaine d'application (système de vidéosurveillance) et est adaptable à d'autres domaines.

4.2 Étapes génériques pour la mise en œuvre de l'enrichissement contextuel

Nous définissons trois principales étapes permettant de mettre en œuvre l'enrichissement contextuel dans un système ou une application quelconque : (i) l'analyse des informations contextuelles utiles pour le système, (ii) la modélisation des informations contextuelles obtenues grâce à l'analyse et (iii) la mise en œuvre d'un mécanisme de requêtage basé sur la modélisation des informations contextuelles.

Étape 1 **- Analyser les informations contextuelles utiles pour le système** : avant de concevoir un système, il faut savoir quels objectifs le système doit atteindre. Dans le cadre des systèmes gérant de grandes masses de données, un exemple d'objectif peut être la réduction du volume de données à traiter. L'usage des informations contextuelles dans ce cas devrait contribuer à atteindre cet objectif. Connaissant les objectifs du système, l'objectif à cette étape consiste à déterminer les informations contextuelles ou sources d'informations contextuelles susceptibles d'apporter de la valeur aux données du système.

Étape 2 **- Modéliser les informations contextuelles** : les informations contextuelles utiles pour un système peuvent provenir de diverses sources, avoir des formats différents et être représentées différemment. Dès lors, il est nécessaire de proposer une modélisation uniforme de ces informations contextuelles afin de faciliter leur utilisation. Comme présenté dans l'état de l'art (section 2.2.2), le choix des techniques de modélisation des informations contextuelles dépend des exigences du système.

Étape 3 **- Développer un mécanisme de requêtage** : l'idée est de proposer des algorithmes basés sur les informations contextuelles modélisées afin de répondre aux attentes du système. Des algorithmes peuvent être proposés pour chaque opération contextuelle (filtrage, agrégation et inférence) en fonction des besoins du système.

Comme illustré à la Figure 4.1, l'enrichissement contextuel est un processus itératif. Les différentes étapes du processus peuvent être répétées pour assurer l'extensibilité/évolutivité du système.

La section suivante présente l'utilisation de ces étapes génériques d'enrichissement contextuel pour le filtrage et l'interrogation de grands volumes de vidéos.

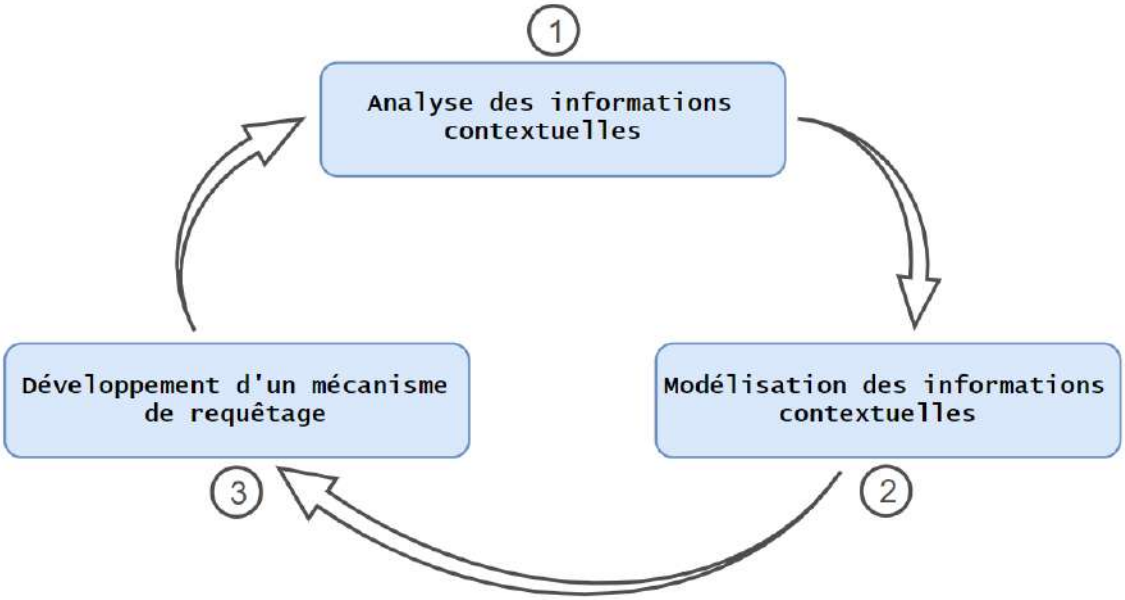


FIGURE 4.1 – Étapes génériques pour la mise en œuvre de l'enrichissement contextuel.

4.3 Enrichissement contextuel : application aux systèmes de vidéosurveillance

Dans un cadre d'application comme le nôtre (vidéosurveillance), sujet à l'exploitation de grands volumes de données, l'enrichissement contextuel peut permettre d'améliorer le processus de filtrage (filtrer à partir d'autres points d'intérêt) et de recherche dans les vidéos afin de réduire le volume de vidéo à exploiter ou à visionner. Comme expliqué dans l'introduction de ce mémoire, les opérateurs de vidéosurveillance sont confrontés à une énorme quantité de vidéos à exploiter/visionner quand il faut rechercher des scènes, des objets, ou des personnes cibles. Exploiter/visionner certaines séquences vidéo s'avère une perte de temps pour plusieurs raisons : contenu inadapté au besoin des opérateurs, conditions de prise de vue inexploitables (ex : présence du brouillard, pollution), etc. Le besoin des opérateurs est de pouvoir traiter uniquement les séquences vidéo d'intérêt. Des nombreuses solutions basées sur les algorithmes d'analyse de contenus vidéo ont été proposées pour répondre à ce besoin. Néanmoins, ces solutions nécessitent des améliorations pour être opérationnelles dans certaines applications. L'approche proposée ici est complémentaire aux approches existantes, et a pour but de réduire considérablement le volume de vidéo à exploiter et de faciliter la recherche dans ces vidéos grâce à l'enrichissement contextuel.

4.3.1 Analyse des informations contextuelles utiles pour la vidéosurveillance

Diverses définitions du contexte ont été proposées dans l'état de l'art de ce travail. En fonction du domaine, de l'application, ou des besoin d'un système, le contexte peut avoir différentes significations et être utilisé à de fins multiples. De nombreux travaux de la littératures proposent des solutions applicables à la vidéosurveillance en exploitant des informations contextuelles issues des flux vidéo (structures spatiales, changements temporelles, actions, etc.). Dès lors, le contexte ou l'information contextuelle peut être définit comme toute information (contenu dans la vidéo) susceptible d'influencer la façon dont une scène est perçue. Par ailleurs, il existe des informations contextuelles exogènes pouvant influencer l'interprétation des vidéos. Une analyse de ces informations est présentée dans cette section. Les sources d'informations contextuelles externes (à la vidéo) sont multiples. Celles prises en compte dans notre étude sont : les données ouvertes (Open Data), les médias sociaux, la mobilité et la géolocalisation.

4.3.1.1 Données ouvertes ou Open Data

Les données ouvertes sont des données pour lesquelles l'accès est totalement public et libre de droit, au même titre que l'exploitation et la réutilisation. Ce sont des données qui offrent de nombreuses opportunités pour étendre le savoir humain et créer de nouveaux produits et services de qualité. Les données ouvertes peuvent être utiles dans de nombreux secteurs, à divers groupes de personnes et organisations. Pour chaque catégorie de données, les avantages peuvent être spécifiques et peuvent varier selon les domaines et les applications. Par exemple : dans le domaine des transports, le partage des données de transports publics et des emplacements des bouchons en temps réel permet aux usagers de mieux se déplacer ; dans le domaine de la santé, le partage des données relatives à la présence des maladies contagieuses et les zones (localités) concernées permet aux populations d'éviter ou de se faire prendre en charge en cas de contaminations. Les Open Data créent de nos jours des valeurs économiques et sociales, et des nouveaux champs d'usage peuvent être développés dans différents domaines.

Nous nous intéressons dans cette étude à l'usage que l'on peut faire des données ouvertes dans le processus de réduction de grands volumes de vidéos issues des systèmes de vidéosurveillance. L'apport des données ouvertes peut varier en fonction des besoins de l'opérateur de vidéosurveillance et des différentes sources de données ouvertes (données géographiques, météo, environnement, finances, ...). Par exemple, si le besoin de l'opérateur consiste à visionner ou à traiter prioritairement/uniquement les vidéos de bonne qualité, les sources d'informations contextuelles Open Data telles que

la météo et/ou de l'environnement peuvent être utiles, car elles influencent la qualité des images lors de leur acquisition par les caméras de vidéosurveillance, ce qui peut rendre difficile ou impossible la détection des objets ou d'évènements par l'œil humain ou par les algorithmes de traitement automatique. Une liste non-exhaustive des métadonnées pertinentes que l'on peut tirer des informations contextuelles Open Data dans ce cas d'application (exploitation d'images de bonne qualité) est :

— Présence des particules atmosphériques : pluie, brouillard, pollution, poussière, etc.

— Faible éclairage de la scène : obscurité, nuit.

— Occultation de la scène.

Toutes ces informations contextuelles peuvent être utilisées dans le processus de réduction de volume de données à traiter via une opération de filtrage contextuel. Les informations contextuelles Open Data utilisées dans cette thèse proviennent de la plateforme Data Toulouse Metropole [1]. Cette plateforme met à disposition des données ouvertes variées parmi lesquelles les données de météo utilisables pour le filtrage des vidéos. Les données liées à la météo sont enregistrées périodiquement (chaque 15 minutes) pour chaque station météo et fournissent les informations présentées à la Figure 4.2 telles que : l'heure d'enregistrement des données (heure_utc), la température moyenne, la valeur de la pression, la quantité de précipitation, l'intensité maximale de pluie, etc.

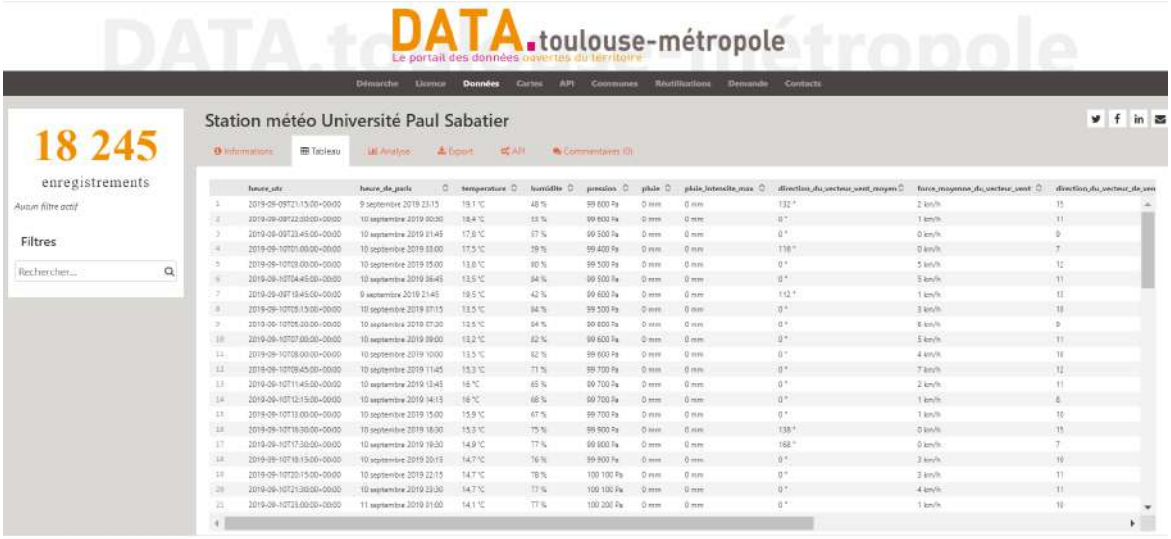


FIGURE 4.2 – Extrait des données ouvertes de DATA Touluse Métropole.

1. https ://data.toulouse-metropole.fr/pages/accueil/

4.3.1.2 Médias sociaux

Les médias sociaux désignent des supports de diffusion massive d'informations utilisant toutes les formes existantes (texte, image, vidéo, audio...) et permettant une interaction sociale. Les technologies ou plateformes de médias sociaux prennent différentes formes telles que le partage de photos et de vidéos, les blogues, les réseaux sociaux (Facebook, Twitter, Instagram, LinkedIn, etc.), les forums, les réseaux d'affaires, les revues et bien d'autres. Ces plateformes connectent de nombreux utilisateurs et génèrent également une quantité importante de données individuelles et relationnelles. Les données disponibles sur les médias sociaux présentent une opportunité de pouvoir pallier les limites des données à caractère incomplet ou de guider l'interprétation qui doit en être faite. Les médias sociaux font de plus en plus partie intégrante de la « vie en ligne » grâce au nombre croissant de sites web (dotés de composantes sociales telles que des champs de commentaires pour les utilisateurs) et applications sociales, et sont déjà utilisés à des fins professionnelles. Par exemple, les médias sociaux sont utilisés pour commercialiser des produits, promouvoir des marques, établir des liens avec les clients actuels et favoriser de nouvelles affaires. L'analyse des médias sociaux permet de sonder l'opinion des clients pour soutenir les activités de marketing et de services à la clientèle.

Les données issues de médias sociaux constituent une source d'information supplémentaire pouvant être utilisée en collaboration avec les métadonnées de vidéosurveillance afin de faciliter ou d'améliorer le filtrage et la recherche dans les grands volumes de vidéos. En effet, les différentes plateformes des médias sociaux (Facebook, Twitter, etc.) sont considérées comme des outils accessibles et contenant des informations utilisables pour la communication, le renseignement ou encore les enquêtes. A titre d'exemple, les vidéos issues des caméras de surveillance ne contiennent pas toujours suffisamment d'informations pour l'analyse d'une situation ou d'un fait (incident, évènement, ...). Exploiter d'autres sources d'informations devient alors une nécessité pour les opérateurs de vidéosurveillance. Les médias sociaux constituent donc une source d'information enrichissante pouvant permettre de faciliter la détection, l'identification, la localisation dans les vidéos et aussi l'interprétation ou la déduction de nouvelles connaissances via l'analyse des profils utilisateurs, les liens (abonnés, amis, groupes, ...) et les contenus partagés (textes, images, vidéos, ...).

D'autres informations contextuelles peuvent provenir du crowdsourcing. Le crowdsourcing est un type d'activité participative en ligne qui regroupe un grand nombre de personnes dans le but de travailler sur une tâche ou un ensemble de tâches spécifiquement définies. Les tâches peuvent s'effectuer de manière collaborative ou en parallèle. Le crowdsourcing est aussi utilisé pour la collecte d'informations provenant du grand public et utiles pour l'accomplissement de certaines tâches. Par exemple,

dans le domaine des sciences environnementales, le crowdsourcing est utilisé pour acquérir un plus grand nombre de données dans des zones géographiquement éloignées et nécessitant la présence des personnes pour la collecte. Les données collectées grâce au crowdsourcing peuvent constituer une source d'information enrichissante pour les contenus vidéo et métadonnées de vidéosurveillance. Par exemple, supposons une plateforme de crowdsourcing mise en place par la police et permettant au public de publier les informations (audios, vidéos, images, etc.) relatives à un incident survenu. Cette plateforme pourra créer de la valeur en agrégeant ou en croisant les contenus avec les vidéos issues de caméras de vidéosurveillance, et offrir des pistes d'investigation par l'intermédiaire du grand public.

4.3.1.3 Mobilité et géolocalisation

La mobilité, aujourd'hui appelée mobilité intelligente (transport + mobilité + numérique) désigne l'ensemble des moyens de transport, des technologies et applications de transport et de communication mis en œuvre pour assurer le déplacement (réservation, calcul d'itinéraire, etc.) des usagers en garantissant l'efficacité (ex : contrôle de vitesse), la sécurité (ex : sécurité des personnes) et le confort (ex : info-trafic, internet à bord). En effet, les systèmes d'information pour la mobilité ne concernent plus seulement les moyens de transport, ils intègrent aussi les technologies informatiques (billettique, télépéage, contrôles, etc.) et télécommunication (sites internet, Smartphone, SMS, etc.). La mobilité intelligente vise donc à fournir aux usagers des solutions adaptées, efficaces, confortables et sécurisées pour leurs déplacements.

Les systèmes d'information de mobilité offrent un ensemble de services parmi lesquels : le calcul d'itinéraires, la présentation des cartes de transport (navigation), la recherche des transports et services à proximité et les services connexes (ex : billettique). Ces services sont utilisés par les usagers et produisent un ensemble de données utilisables à d'autres fins ou dans d'autres domaines. Les données issues des services de mobilité telles que les données de localisation, les trajectoires, les informations spatiales et temporelles, les mouvements des entités (usagers, véhicules, ...), les données de connectivité et d'accès aux différents services peuvent constituer une source d'information utilisable par les opérateurs de vidéosurveillance pour faciliter l'exploitation des vidéos qui manquent parfois des informations nécessaires à l'identification ou au suivi d'une entité.

Les services de mobilité utilisent généralement les technologies de géolocalisation. La géolocalisation désigne un ensemble de moyens permettant de localiser des personnes ou des objets (véhicules, équipements, etc.) sur une carte ou un plan grâce à leurs coordonnées géographiques. Il existe de nos jours de nombreuses techniques ou technologies de géolocalisation : par satellite, par géocodeur, par GSM, par Wifi, par RFID, par adresse IP, etc. Les avancées technologiques (en informatique, réseau,

télécommunication) ont permis de développer de nombreux capteurs et services de géolocalisation et de les intégrer aux équipements (smartphones, tablettes, etc.), et objets utilisés au quotidien. Les applications de géolocalisation sont en plein développement et offrent de nombreux services utiles tels que le suivi temps réel et historique, la localisation (lieu d'incident), la navigation vers les lieux d'intervention, les trajectoires détaillées sur carte, la localisation des objets proches ou dans une zone spécifique, etc. Les informations de localisation générées par les systèmes et applications de géolocalisation peuvent constituer une source externe d'information permettant de localiser, suivre les traces des personnes et objets (véhicules, équipements, ...) dans les vidéos provenant des caméras de surveillance.

Dans la section suivante, nous proposons une modélisation de ces informations contextuelles multisources afin de faciliter ou rendre possible leur interopérabilité.

4.3.2 Modélisation des informations contextuelles

L'idée est de proposer une approche qui s'appuie sur des informations contextuelles afin de faciliter le filtrage et l'interrogation des grands volumes de vidéos. La proposition faite dans ce cadre est une modélisation de métadonnées issues des différentes source d'informations contextuelles décrite à la section 4.3.1. D'autres métadonnées de vidéosurveillance telles que les métadonnées descriptives et les métadonnées sémantiques sont prises en compte dans cette modélisation. Le modèle proposé est constitué de cinq sources d'information (voir Figure 4.3) : (1) métadonnées descriptives, (2) métadonnées sémantiques, (3) open data, (4) médias sociaux, (5) mobilité et géolocalisation. Les liaisons entre les métadonnées des différentes sources du modèle se font grâce aux informations spatio-temporelles. Dans les sections suivantes, nous proposons une modélisation des métadonnées pour chacune des sources d'information contextuelle, et nous présentons les entités prises en compte dans la modélisation ainsi que les relations entre les différentes entités.

4.3.2.1 Modélisation des métadonnées descriptives

Les performances d'un système de vidéosurveillance dépendent en partie des performances des caméras qui le composent et de leur installation (réglages, position, etc.). Les métadonnées descriptives de vidéosurveillance représentent généralement les informations décrivant la caméra, son installation, sa configuration, son emplacement, son orientation, etc. Nous nous focalisons sur la modélisation de toutes ces informations. La position de la caméra et l'information temporelle associée sont des éléments clés de la modélisation, car elles permettent d'établir des liens avec les informations contextuelles. La Figure 4.5 illustre le modèle de données représentant les métadonnées descriptives de vidéosurveillance.

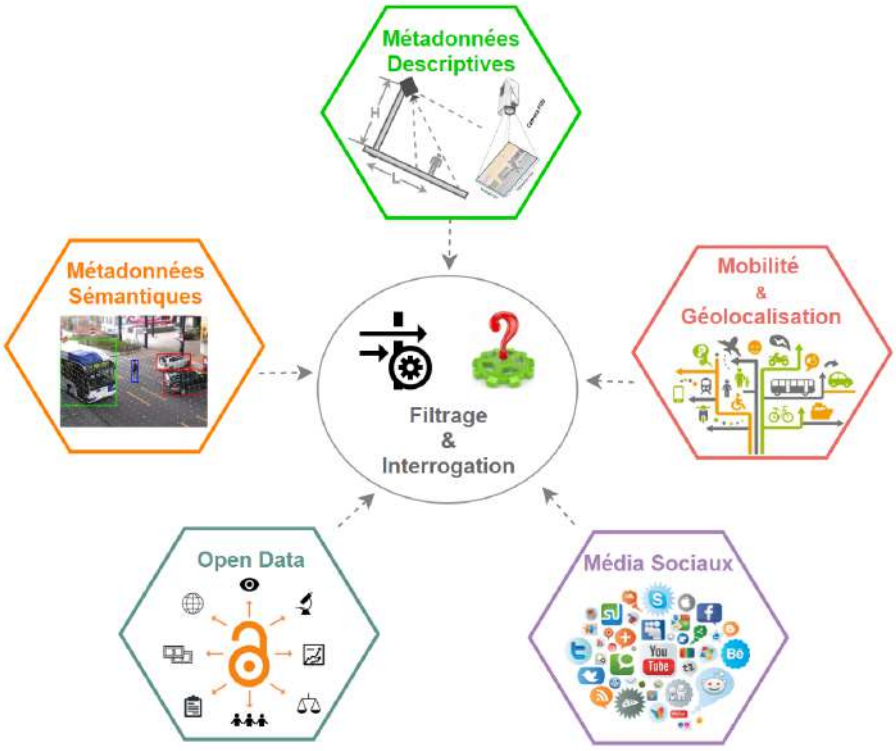


FIGURE 4.3 – Sources d'informations contextuelles.

Conformément à ce modèle de données, une caméra située à une position donnée à un temps donné, avec une orientation et une installation spécifiques, peut filmer une zone précise. Selon qu'une caméra est installée à un endroit fixe (rue, station de métro, commerce, etc.) ou sur un objet mobile (bus, train, etc.) on parle de caméra fixe ou mobile. Les caméras mobiles peuvent avoir plusieurs positions avec des champs de vision variables à différents moments. La Figure 4.4 illustre le champ de vision d'une caméra. Les métadonnées décrivant les paramètres (résolution, contraste, luminosité, etc.) des images filmées par la caméra sont prises en compte dans le modèle. Le modèle de données proposé est évolutif dans ce sens où il peut intégrer de nouveaux capteurs et leurs spécifications (héritage du "CAPTEUR").

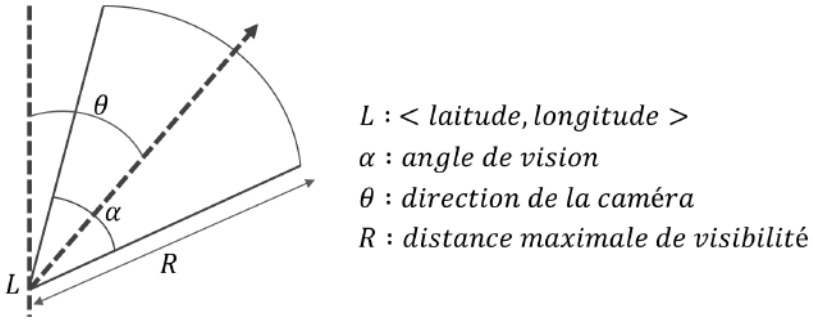


FIGURE 4.4 – Champs de vision d'une caméra.

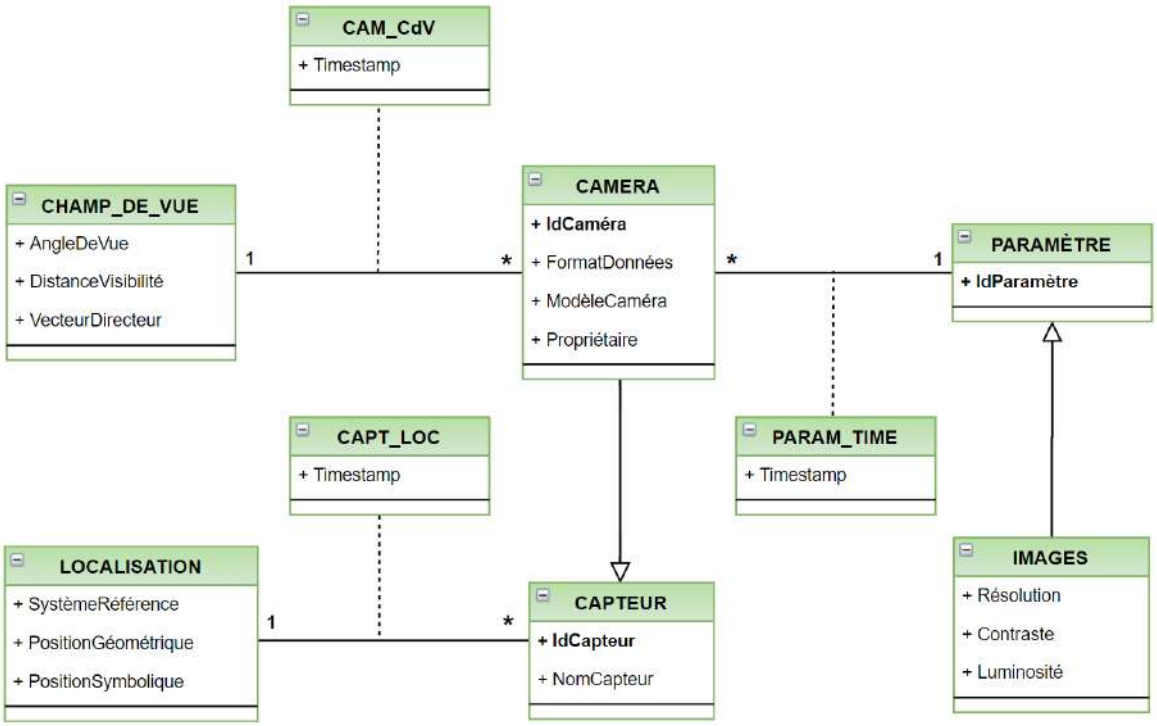


FIGURE 4.5 – Métadonnées descriptives de vidéosurveillance.

4.3.2.2 Modélisation des métadonnées sémantiques

Les métadonnées sémantiques sont des caractéristiques vidéo (haut niveau) obtenues grâce aux algorithmes d'analyse de contenu et qui fournissent une valeur informative pertinente pour la compréhension du contenu vidéo par l'humain. Ces métadonnées fournissent des informations sur les objets (détection, identification, propriétés, mouvements, localisation, etc. des objets) et les évènements (description des évènements) dans les vidéos. Dans cette étude nous nous intéressons aux métadonnées décrivant la détection et le mouvement des objets dans les vidéos et les métadonnées décrivant les évènements. Ces métadonnées peuvent permettre de filtrer les contenus vidéo afin d'éviter aux opérateurs de vidéosurveillance de visionner des séquences vidéo ne contenant pas d'informations pertinentes. A titre d'exemple, visionner des séquences vidéo ne contenant aucun objet (personnes, véhicules) et dans lesquelles il n'y a aucun mouvement constitue une perte de temps pour les opérateurs.

La Figure 4.6 présente le modèle de données décrivant les métadonnées sémantiques prises en compte dans notre approche. D'après ce modèle, les métadonnées sont extraites pour chaque frame de la vidéo. Une vidéo est composée d'au moins une frame et peut contenir plusieurs évènements. Un évènement est une action impliquant des éléments de contenu à un emplacement donné et pendant un intervalle de temps donné (par exemple des personnes suspectes sortant d'un bâtiment, et entrant dans un véhicule stationné à un emplacement non-indiqué). Une vidéo peut contenir plus d'un évènement. La détection des objets est faite par frame, et une frame peut contenir

plusieurs objets. Le modèle définit les métadonnées permettant de spécifier si un objet dans une frame est en mouvement ou non par rapport à la frame précédente.

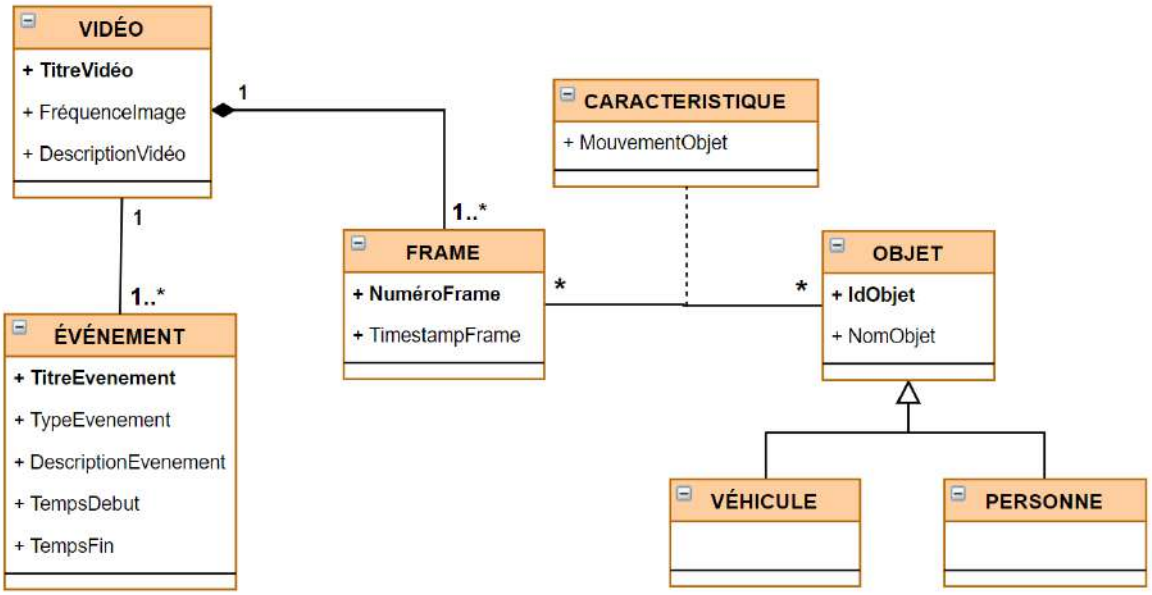


FIGURE 4.6 – Métadonnées sémantiques de vidéosurveillance.

4.3.2.3 Modélisation des métadonnées issues de l'open data

Les sources de données ouvertes sont nombreuses et peuvent être : géographiques (données utiles pour la formulation des cartes), culturelles (données sur les œuvres et objets culturels), scientifiques (données produites dans le cadre de la recherche scientifique dans toutes les catégories), environnementales (données relatives à l'environnement physique telles que la pollution, les rivières, les mers, les montagnes, les volcans, la météo, etc.), gouvernementales (données publiées par le gouvernement afin d'assurer la transparence de leurs plans et politiques auprès du grand public), etc.

Selon les objectifs du cas applicatif traité dans cette thèse (filtrage et interrogation des vidéos issues des systèmes de vidéosurveillance), nous nous sommes concentrés sur les métadonnées provenant des données environnementales ouvertes. Nous prenons l'exemple des métadonnées issues des données de météo et de pollution. Le modèle de données proposé à la Figure 4.7 vise à intégrer les métadonnées issues des données ouvertes environnementales afin de permettre le filtrage des séquences vidéo dégradées par des phénomènes liés à l'environnement. Les métadonnées prises en compte dans ce modèle décrivent des évènements (pluie, vent, brouillard, pollution, etc.) : emplacement, l'intervalle de temps de survenue et autres descriptions propres à chaque évènement. L'intervalle de temps d'un évènement est défini ici par un *temps relatif* (qui sera décrit à la section 4.3.3) qui permet de faciliter la représentation temporelle des évènements dynamiques. Grâce à l'héritage de la classe "ENVIRONNEMENT", le

modèle de données offre la possibilité d'intégrer de nouveaux évènements qui n'ont pas été pris en compte lors de sa conception.

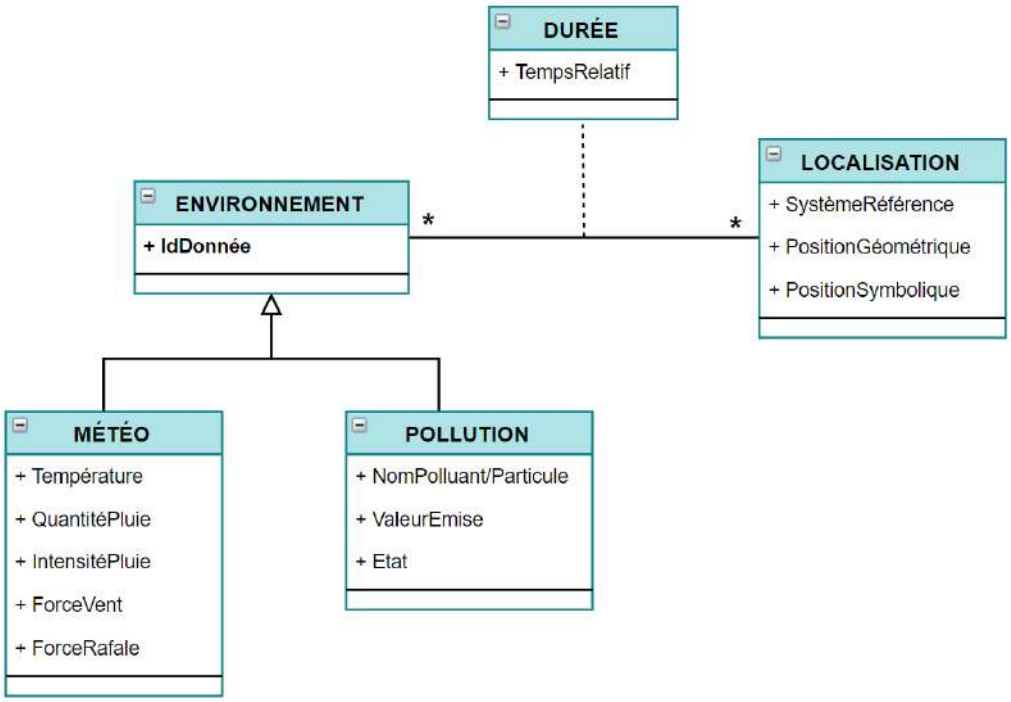


FIGURE 4.7 – Métadonnées issues de données ouvertes.

4.3.2.4 Modélisation des métadonnées issues des médias sociaux

Les métadonnées modélisées à ce niveau représentent les informations décrivant les profils, les liens entre profils (amis, groupes, etc.), les contenus publiés et les évènements sociaux. Toutes ces informations peuvent être utilisées de manière spécifique afin d'apporter de nouvelles connaissances dans le processus d'analyse des vidéos issues des systèmes de vidéosurveillance. Par exemple, collecter les informations des réseaux sociaux d'un individu (profil) ciblé dans une vidéo filmée par les caméras de vidéosurveillance constituerait des renseignements pertinents pour les opérateurs de vidéosurveillance. Un modèle de données intégrant les métadonnées issues des médias sociaux et facilitant l'interrogation de ces métadonnées est présenté à la Figure 4.8. Ce modèle permet de gérer les profils utilisateurs et les différentes relations pouvant exister entre les profils telles que les liens d'amitié et l'appartenance à des groupes. Les différents types de contenu (texte, images, vidéos, etc.) publiés par les utilisateurs ainsi que les dates de publication sont pris en compte dans le modèle. Le modèle de données intègre aussi la gestion des évènements sociaux (ex : anniversaire). La collecte des données relatives aux profils des individus est faite dans le respect du règlement général sur la protection des données (RGPD).

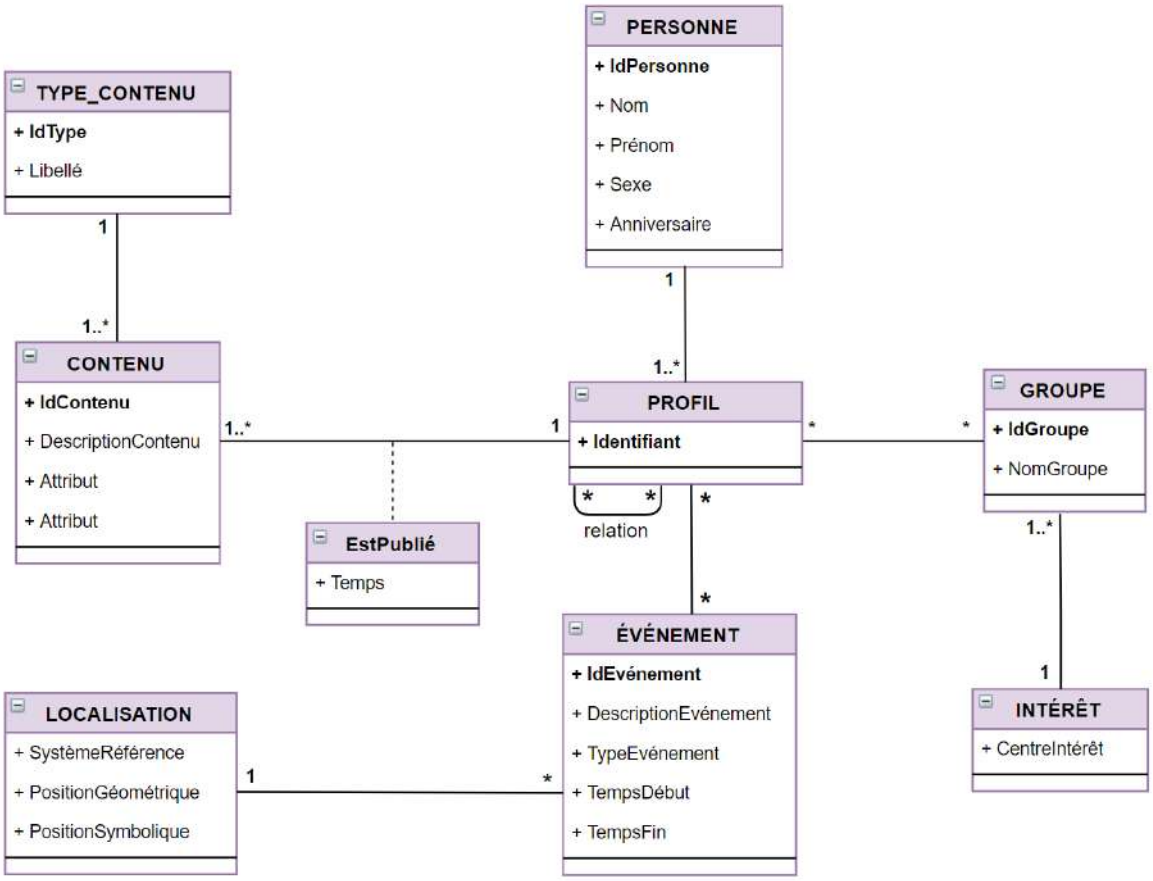


FIGURE 4.8 – Métadonnées issues des médias sociaux.

4.3.2.5 Modélisation des métadonnées issues de la mobilité et la géolocalisation

Les applications et services de mobilité génèrent une quantité importante d'informations parmi lesquelles les informations de localisation. Le modèle de données proposé à la Figure 4.9 présente un ensemble de métadonnées décrivant les informations de la mobilité telles que la localisation des équipements et des objets mobiles (véhicules, personnes), les horaires et trajets des systèmes de transport en commun.

4.3.2.6 Modèle générique de métadonnées

Les modèles de métadonnées proposés pour les différentes sources d'inormation contextuelle (métadonnées descriptives, métadonnées sémantiques, open data, média sociaux, mobilité et géolocalisation) sont intégrées dans le modèle générique de la Figure 4.10. Dans ce modèle générique, les relations spatio-temporelles sont généralement nécessaires pour le croisement des métadonnées issues des différentes sources. Par exemple, pour récupérer les segments vidéo dans lesquels la pollution empêche la visibilité, on doit croiser les localisations des caméras (métadonnées descriptives) et les métadonnées issues des open data (*NomPolluant, valeurEmise,* ***localisation***)

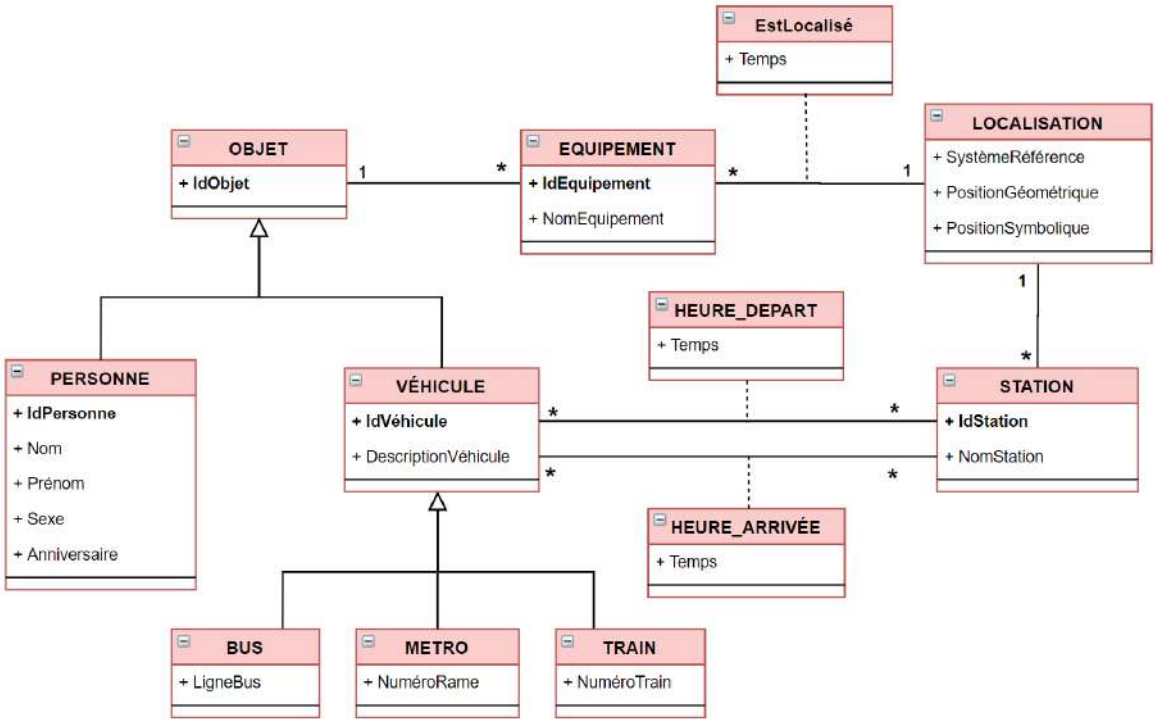


FIGURE 4.9 – Métadonnées issues de la mobilité et la géolocalisation.

pour sélectionner les vidéos de la zone d'intérêt (**relation spatiale** définie dans le modèle par les liens entre les classes "ENVIRONNEMENT", "LOCALISATION" et "CAMERA"), ensuite, pour les vidéos sélectionnées, on doit croiser les horaires (heure de début et heure de fin des vidéos) et le temps de survenue de la pollution (intervalle de temps défini dans le modèle par la classe "DURÉE" qui est une classe d'association résultant de la relation entre les classes "ENVIRONNEMENT" et "LOCALISATION") afin de récupérer les segments vidéos concernés (**relation temporelle**).

Le modèle générique rend donc possible l'intégration des informations contextuelles multi sources et offre la possibilité d'exploiter ces informations contextuelles à l'aide des raisonnements spatio-temporels. Ce modèle sera ensuite utilisé pour concevoir une base de données permettant de stocker les métadonnées afin de mettre en œuvre un mécanisme de filtrage et d'interrogation des masses de vidéos issues des systèmes de vidéosurveillance.

La section suivante décrit une représentation temporelle appropriée des métadonnées provenant des évènements dynamiques.

4.3.3 Représentation temporelle des évènements dynamiques

Les informations contextuelles prises en compte dans le modèle de données générique proposé proviennent parfois des phénomènes ou d'évènements dynamiques (ex : pluie, brouillard, pollution). L'étude ou l'analyse d'un évènement dynamique dépend d'un paramètre important qui est **l'échelle temps** selon laquelle cet évènement est

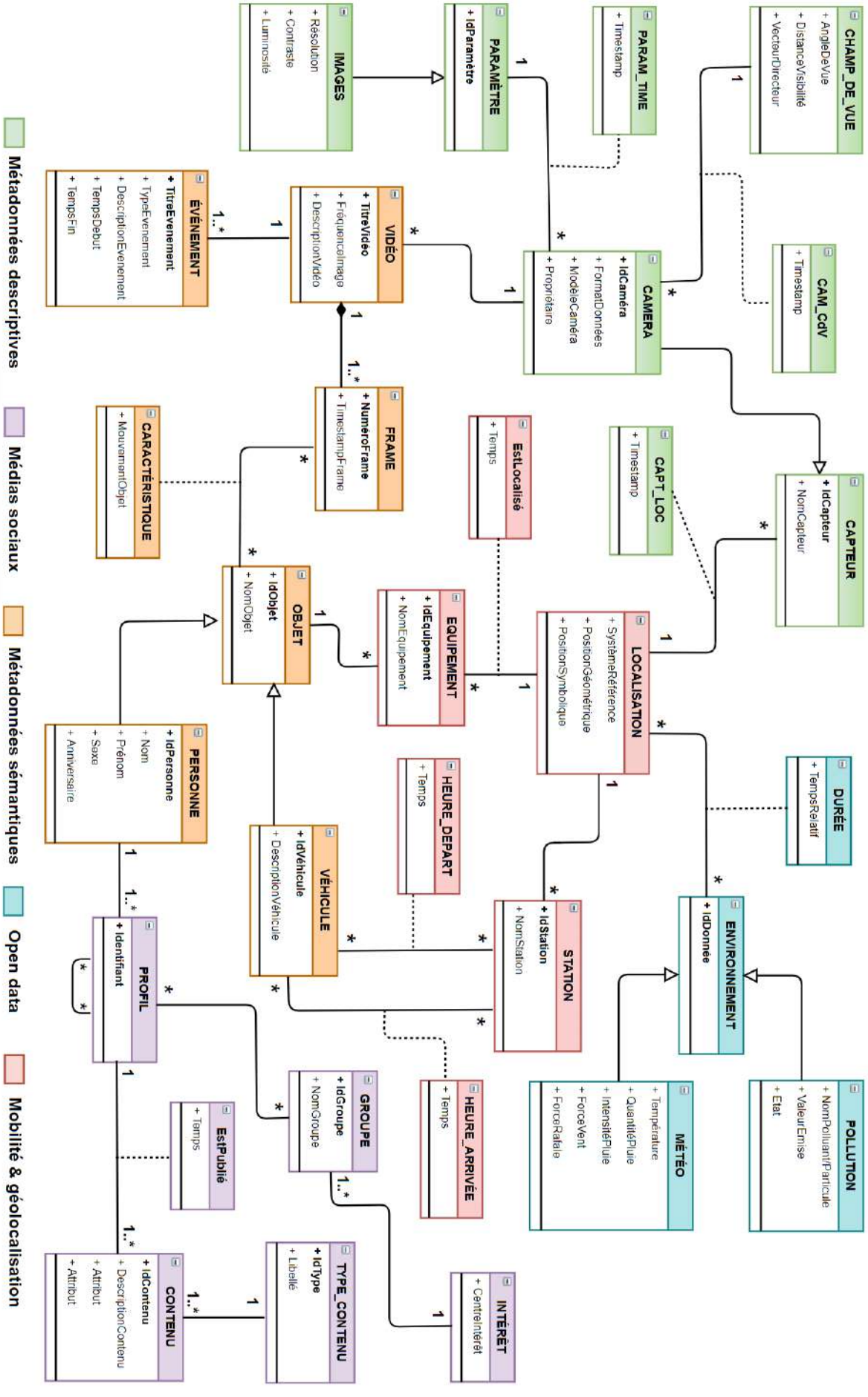


FIGURE 4.10 – Modèle générique des métadonnées de contexte de la vidéosurveillance.

observé. La dynamique de l'évènement peut varier de façon importante selon l'échelle de temps choisie. A titre d'exemple, la pluie est un "évènement" dynamique pouvant affecter la qualité des images générées par une caméra de vidéosurveillance. On souhaite utiliser les données liées à cet évènement telles que l'intervalle de temps de pluie et les intensités maximales de pluie (en millimètre) enregistrées dans une zone géographique, pour effectuer le filtrage des contenus vidéo générés par les caméras de surveillance installées dans cette zone : par exemple, s'il a plu dans le centre ville entre 10h12 et 10h36, les images générées par les caméras du centre ville pendant les instants où l'intensité maximale de pluie était supérieure à 1,75mm sont susceptibles d'être dégradées. Les informations telles que l'occurrence de la pluie et les intensités maximales de pluie enregistrées sont des données météorologiques disponibles en open data. Ces données sont enregistrées par minute pour chaque station météorologique et représentent une masse importante de données à modéliser et à stocker. Le stockage de ces données pour chaque minute peut s'avérer inutile pour cette étude, car on ne s'intéresse pas aux données générées pour chaque instant (chaque minute, chaque 10 minutes, chaque heure, chaque deux heures, chaque jour, etc.), mais on s'intéresse aux instants de variation des données. Par exemple, pour les données du tableau 4.1, la Figure 4.11 montre une représentation de l'intensité maximale de pluie en millimètre (mm) enregistrée par minute entre 10h12 et 10h36 en centre ville.

TABLE 4.1 – Intensités maximales de pluie exprimées en temps classique.

Id	**Temps classique**	**Intensité maximale de pluie (mm)**
1	10h12	0
2	10h13	0
3	10h14	0
4	10h15	0
5	10h16	0
6	10h17	0
7	10h18	2
8	10h19	2
...	...	...
...	...	...
24	10h36	0

Dans les intervalles [1 - 6], [9 - 13], [16 - 18] et [22 - 24] correspondants respectivement aux intervalles de temps [10h12 - 10h18], [10h21 - 10h25], [10h28 - 10h30], [10h34 - 10h36], l'intensité maximale de pluie est égale à 0 mm, ce qui s'interprète par une absence de pluie pendant ces intervalles de temps. Des variations de l'intensité maximale de pluie sont enregistrées dans les intervalles [7 - 8], [14 - 15] et [19 - 21] correspondants respectivement aux intervalles de temps [10h19 - 10h20], [10h26 - 10h27] et [10h31 - 10h33], et indiquent qu'il a plu. L'objectif n'étant pas d'analyser l'intensité

maximale de pluie par minute, le stockage dans la base de données des mesures pour chaque minute entraîne des coûts superflus en terme de temps de traitements. Il est donc nécessaire de choisir une échelle de temps appropriée pour la représentation des évènements dynamiques.

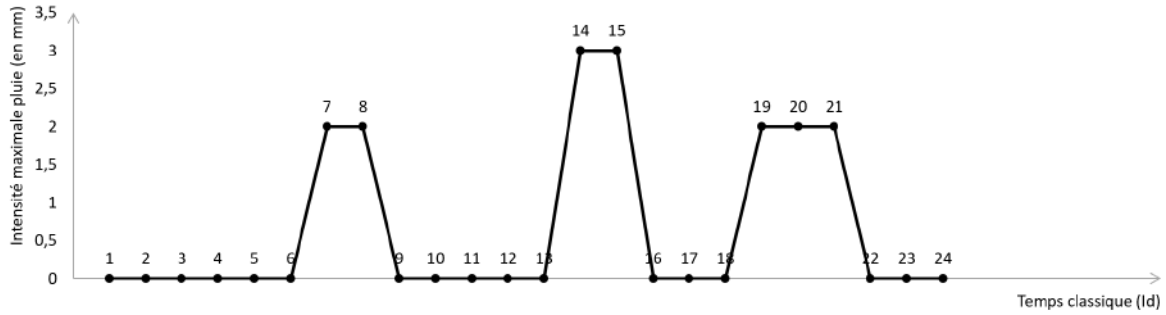


FIGURE 4.11 – Exemple de représentation classique d'un évènement dynamique.

Nous proposons une analyse des évènements dynamiques basée sur une échelle de temps adaptée à leurs variations. Une échelle de temps pertinente doit dépendre directement de l'évènement et non d'une unité absolue telle que l'heure, le jour ou le mois. Nous adoptons ici la notion de **temps relatif**, en contraste avec le "temps classique". Le temps relatif permet d'éviter une analyse "inutile" des évènements à toutes les heures et offre une échelle de temps continue et paramétrable selon les variations des évènements. Des travaux existants ont utilisé la notion de temps relatif dans différents domaines ou applications. [Eckmann et al., 2004] applique la notion de temps relatif pour analyser la diffusion dans un réseau d'e-mail. Les travaux de [Gauvin et al., 2013] utilisent le temps relatif pour analyser la dynamique de propagation des processus sur des réseaux complexes. Récemment, [Heymann and Le Grand, 2013] et [Albano et al., 2015] proposent des définitions du temps relatif pour l'analyse des graphes dynamiques.

Nous définissons comme une unité de temps relatif la variation d'un paramètre de l'évènement dynamique : par exemple *l'intensité maximale* pour la pluie, et *la valeur émise* pour la pollution. Le temps relatif évolue à chaque fois que le paramètre pris en compte pour l'évènement dynamique change. La Figure 4.12 montre un exemple de conversion en temps relatif de la représentation de la pluie en utilisant comme unité de temps relatif la variation de l'intensité maximale de pluie. Sur cette figure, on observe qu'aux instants représentés par Id = 1, Id = 2, Id = 3, Id = 4, Id = 5, Id = 6, l'intensité maximale de pluie enregistrée est égale à 0mm et ne varie pas. Par conséquent, nous avons un pas de temps relatif T1 qui couvre tous ces instants (Id = 1, Id = 2, Id = 3, Id = 4, Id = 5 et Id = 6). Aux instants représentés par Id = 7, et Id = 8, on observe une variation de l'intensité maximale de pluie enregistrée (de 0mm à 2mm). Cela entraîne la création d'un nouveau pas de temps relatif T2 qui couvre les instants (Id = 7, et Id = 8). Les mêmes observations permettent par la suite de créer des pas de temps relatif T3 couvrant les instants (Id = 9, Id = 10, Id = 11, Id = 12,

Id = 13), T4 couvrant les instants (Id = 14, Id = 15), T5 couvrant les instants (Id = 16, Id = 17, Id = 18), T6 couvrant les instants (Id = 19, Id = 20, Id = 21) et T7 couvrant les instants (Id = 22, Id = 23, Id = 24).

Les données à stocker et à analyser passent de 24 enregistrements (temps classique tableau 4.1) à 7 (temps relatif tableau 4.2). Le temps relatif permet donc une représentation adéquate des évènements dynamiques et réduit considérablement la quantité de données à stocker, ce qui facilite le processus d'interrogation des données (moins de données à interroger).

TABLE 4.2 – Intensités maximales de pluie exprimées en temps relatif.

Id	Temps relatif	Intensité maximale de pluie (mm)
T1	10h12 - 10h18	0
T2	10h19 - 10h20	2
T3	10h21 - 10h25	0
T4	10h26 - 10h27	3
T5	10h28 - 10h30	0
T6	10h31 - 10h33	2
T7	10h34 - 10h36	0

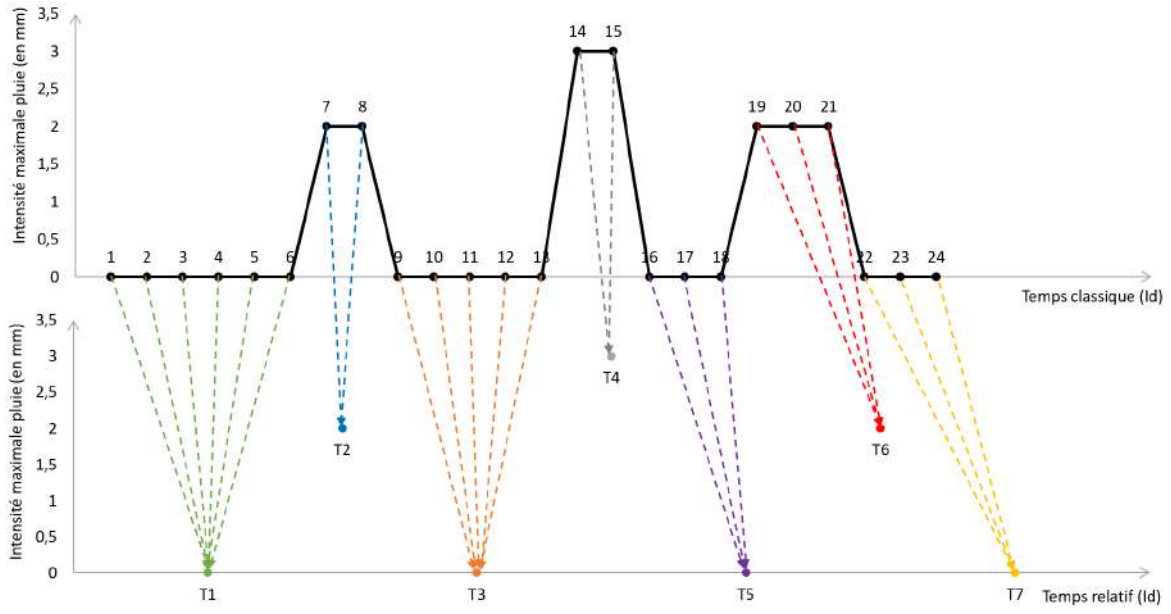


FIGURE 4.12 – Conversion du temps classique en temps relatif.

Fonction de conversion du temps classique en temps relatif

L'algorithme 3 représente une formulation de la conversion du temps classique en temps relatif. Pour un E ensemble de tuples, dans lequel chaque tuple est composé d'un temps "classique" (t_i) et d'une valeur (val_i), l'algorithme 3 retourne un ensemble R de tuples dans lequel chaque tuple est composé d'un intervalle de temps dit "rela-

tif" $[t_{start}, t_{tend}]$ et d'une valeur v. $[t_{start}, t_{tend}]$ délimite une séquence de temps (t_i) successifs dont les valeurs (val_i) sont égales et représentées dans R par v.

$$E : \{(t_i, val_i)\} \rightarrow R : \{([t^k_{start}, t^k_{tend}], v_k)\} \tag{4.1}$$

Algorithm 3: From classical to relative time

Input: a set of tuples composed of a time and a value $(t, val) : E$.
Output: a set of tuples composed of a time interval and a value $([t_{start}, t_{tend}], val)$.

```
1  i ← 0;
2  while i < size(E) do
3  |   v ← val_i;
4  |   t_start ← t_i;
5  |   while i < size(E) and v = val_i do
6  |   |   t_end ← t_i;
7  |   |   i ← i + 1;
8  |   end while
9  |   addResult([t_start, t_tend], v);
10 end while
```

La section suivante présente l'étape 3 de la mise en œuvre l'enrichissement contextuel.

4.3.4 Mécanisme de requêtage

Le mécanisme de requêtage proposé ici utilise la théorie des ensembles flous afin d'offrir une certaine robustesse aux informations **incertaines**, **incomplètes** ou **manquantes**. Les ensembles flous sont généralement efficaces pour la modélisation de l'imprécision et l'incertitude des données. Une donnée est dite **imprécise** si elle est considérée comme incomplète ou insuffisante pour fournir l'information demandée. Par exemple, savoir que *"la caméra C_1 est située entre 50 et 95 mètres de la station de métro Jean Jaurès"* est une information imprécise, car on ne peut donner la distance exacte. Cependant, on sait avec certitude que la distance est dans cet intervalle. Cette distance peut être décrite par les termes *"loin", "proche"* qui sont imprécis du fait qu'on ne peut les associer à une valeur référentielle du mètre (unité de mesure de distance). La notion d'imprécision peut être représentée grâce à la définition des valeurs plus ou moins possibles. Pour une distance *"proche"* par exemple, plus la valeur est faible, plus elle est une valeur possible. Quant à **l'incertitude**, elle représente l'incapacité de savoir si une information (ou affirmation) est vraie ou fausse. Elle se traduit par l'impossibilité de calculer avec précision le degré de vérité de l'information. L'imprécision des données peut être une source d'incertitude.

Le mécanisme de requêtage s'appuie sur le modèle générique de métadonnées décrit à la section 4.3.2.6. Ce modèle générique agrège des informations contextuelles pro-

venant de sources différentes. Chaque source d'information est considérée comme un critère de filtrage ou d'interrogation dans le mécanisme de requêtage. En fonction de ses besoins (amélioration des résultats), l'utilisateur peut donc effectuer des requêtes successives avec des critères d'interrogation portés sur les différentes sources d'informations contextuelles. L'idée est d'aboutir à un requêtage multi-niveaux comme illustré à la Figure 4.13. Sur cette figure, les données en entrée sont des vidéos filmées par des caméras installées dans une zone géographique (information spatiale) et pendant un intervalle de temps (information temporelle) donnés. Ces informations spatiales et temporelles sont prises en compte dans les critères à chaque niveau d'interrogation des métadonnées. L'ordre d'interrogation (choix de la source d'information contextuelle) dépend des besoins de l'opérateur humain. Le résultat à chaque niveau est soit un ensemble de données filtrées, soit des données classées selon leur degré d'importance, conférant ainsi à l'utilisateur des préférences sur l'ensemble des résultats.

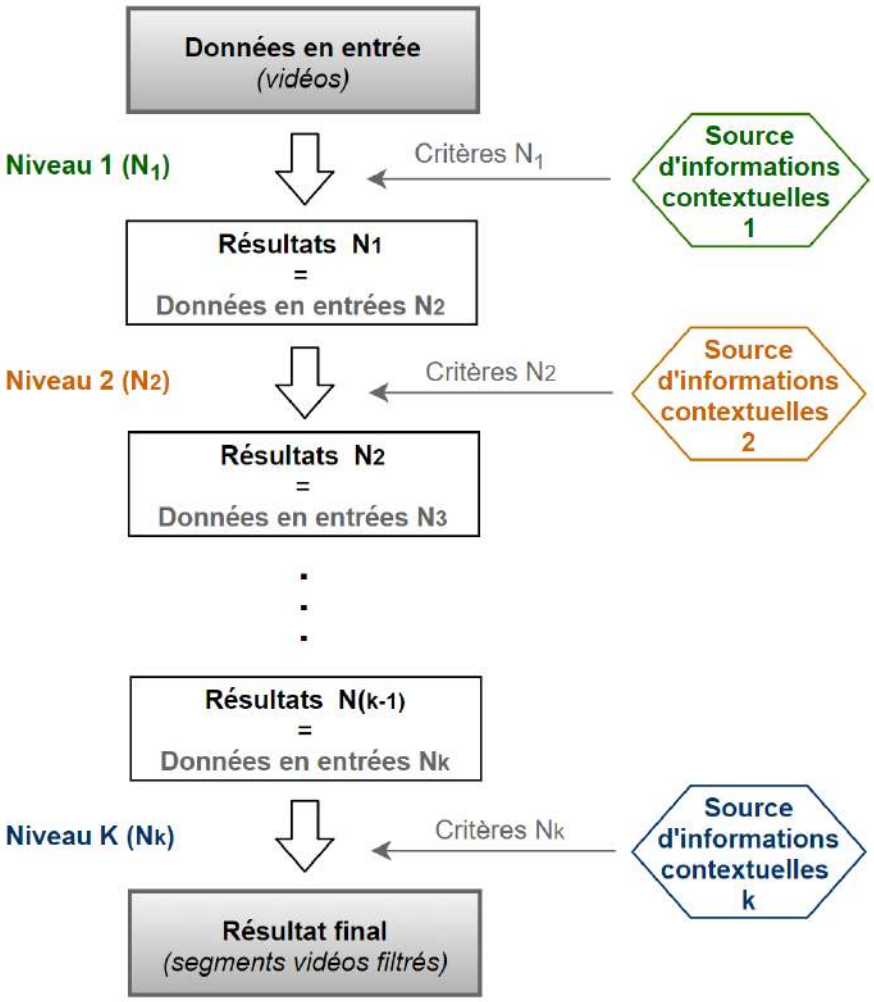


FIGURE 4.13 – Requêtage multi-niveaux.

Afin d'introduire la notion de préférence dans les requêtes, nous définissons des ensembles flous pour représenter les termes imprécis des requêtes. Cette définition est

décrite dans l'élaboration de la réponse à la requête présentée dans l'exemple suivant :

Exemple 1. On souhaite classer les segments d'une vidéo selon leur degré de visibilité qui peut être dégradé par la pluie lors de l'acquisition des images par les caméras. L'objectif est de faciliter la tâche à l'opérateur en lui permettant de visionner en priorité les segments vidéo avec les meilleurs degrés de visibilité et ensuite les segments dont les degrés de visibilité sont faibles (s'il le souhaite car probable qu'il n'y voit rien). Pour cela, on suppose une vidéo titrée *"enr_103.mp4"* et filmée le 05/12/2019 entre 08h25 et 11h00. Les informations de météo (intensités maximales de pluie par minute) concernant le lieu où la vidéo a été filmée pour cet intervalle de temps sont données dans le tableau 4.3.

TABLE 4.3 – Exemple de données météo (intensités maximales de pluie).

Id	**Temps classique (dddd/mm/jj hh :mm)**	**Intensité maximale de pluie (en mm)**
1	2019/12/05 08 :00	0.2
2	2019/12/05 08 :01	0.2
3	2019/12/05 08 :02	0.2
4	2019/12/05 08 :03	0.2
5	2019/12/05 08 :04	0.2
6	2019/12/05 08 :05	0.2
7	2019/12/05 08 :06	0.2
8	2019/12/05 08 :07	0
9	2019/12/05 08 :08	0
10	2019/12/05 08 :09	0
11	2019/12/05 08 :10	0
12	2019/12/05 08 :11	0
14	2019/12/05 08 :12	0
15	2019/12/05 08 :13	0
...	...	...
..	...	...
165	2019/12/05 11 :00	0.1

Construction de la réponse à la requête de l'exemple 1. Afin de faciliter l'interrogation des données météo du tableau 4.3, une représentation des intensités maximales de pluie en utilisant des temps relatifs est présentée au tableau 4.4. A partir de l'intensité maximale de pluie, il est possible d'estimer le niveau de visibilité des images filmées sous la pluie. L'intensité maximale de pluie est un scalaire exprimé en millimètre (mm). Plus sa valeur est grande, moins les images filmées sous la pluie sont visibles. Nous définissons ainsi *"la visibilité"* comme un ensemble flou représenté par la fonction d'appartenance illustrée à la Figure 4.14.

Selon cette représentation, la valeur de l'intensité maximale de pluie (notée Q) décrit la visibilité comme suit :

— Pour $Q \in [0, 0.3[$, la visibilité est *parfaite*.

TABLE 4.4 – Conversion des données météo en temps relatif.

Id	Temps relatif (dddd/mm/jj hh :mm - hh :mm)	Intensité maximale de pluie (en mm)
T1	2019/12/05 08 :00 - 08 :06	0.2
T2	2019/12/05 08 :07 - 08 :23	0
T3	2019/12/05 08 :24 - 08 :38	0.14
T4	2019/12/05 08 :37 - 08 :45	0.27
T5	2019/12/05 08 :46 - 08 :52	0.42
T6	2019/12/05 08 :53 - 09 :15	0.43
T7	2019/12/05 09 :16 - 09 :30	0.57
T8	2019/12/05 09 :31 - 09 :41	0.66
T9	2019/12/05 09 :42 - 10 :00	0.69
T10	2019/12/05 10 :01 - 10 :21	0.35
T11	2019/12/05 10 :22 - 10 :30	0.37
T12	2019/12/05 10 :31 - 10 :36	0.29
T14	2019/12/05 10 :37 - 10 :43	0.18
T15	2019/12/05 10 :44 - 11 :00	0.1

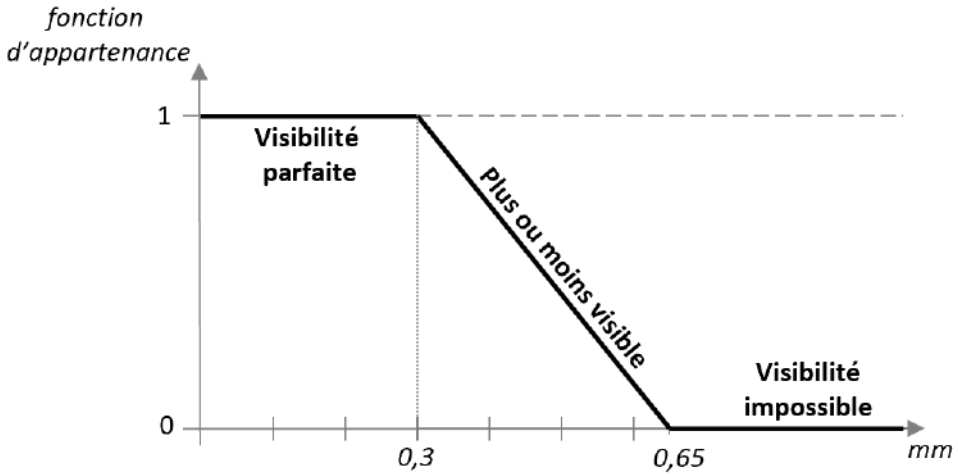


FIGURE 4.14 – Exemple de représentation floue de la "visibilité".

— Pour $Q \in [0.3, 0.65[$, des degrés de visibilité peuvent être calculés afin de fournir des résultats approximatifs selon leur degré de satisfaction.

— Pour $Q \geq 0.65$, la visibilité est *impossible*.

Nous calculons ensuite les degrés d'appartenance pour chaque enregistrement du tableau 4.4. Le processus de calcul du degré de *visibilité* (noté ρ) est présenté à la figure 4.15. Les résultats des calculs sont présentés dans le tableau 4.5.

La prochaine étape consiste à effectuer un croisement temporel entre la vidéo et l'évènement pluie, afin de retourner des segments vidéo selon leur degré de visibilité. Le croisement temporel est défini ici comme une opération qui consiste à faire un rapprochement entre les instants d'une vidéo et des intervalles de temps donnés afin de sélectionner ou annoter les instants de la vidéo. Un tel croisement temporel est illustré

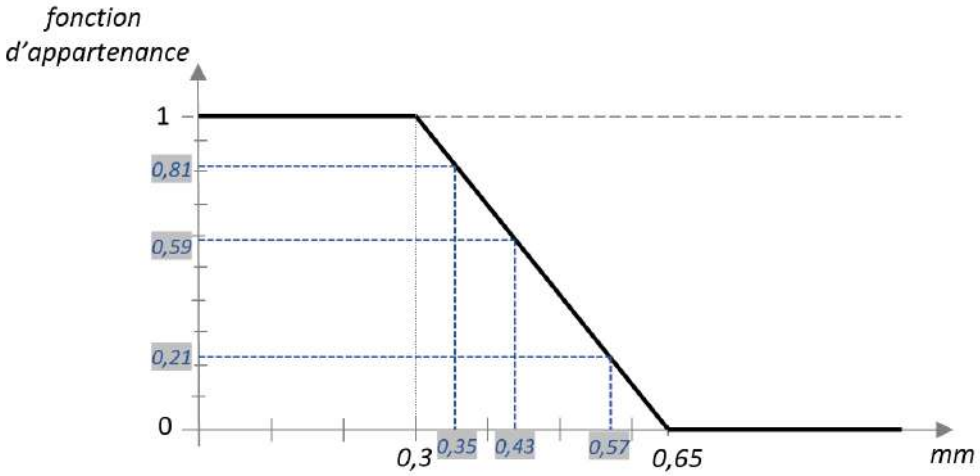


FIGURE 4.15 – Processus de calcul du degré de "visibilité".

TABLE 4.5 – Résultats du calcul des degrés de visibilité.

Id	Temps relatif	Intensité maximale de pluie (en mm)	Degré de visibilité (ρ)
T1	2019/12/05 08 :00 - 08 :06	0.2	1
T2	2019/12/05 08 :07 - 08 :23	0	1
T3	2019/12/05 08 :24 - 08 :38	0.14	1
T4	2019/12/05 08 :37 - 08 :45	0.27	1
T5	2019/12/05 08 :46 - 08 :52	0.42	0.60
T6	2019/12/05 08 :53 - 09 :15	0.43	0.59
T7	2019/12/05 09 :16 - 09 :30	0.57	0.21
T8	2019/12/05 09 :31 - 09 :41	0.66	0
T9	2019/12/05 09 :42 - 10 :00	0.69	0
T10	2019/12/05 10 :01 - 10 :21	0.35	0.81
T11	2019/12/05 10 :22 - 10 :30	0.37	0.82
T12	2019/12/05 10 :31 - 10 :36	0.29	1
T14	2019/12/05 10 :37 - 10 :43	0.18	1
T15	2019/12/05 10 :44 - 11 :00	0.1	1

à la Figure 4.16 et permet de segmenter la vidéo en fonction du degré de visibilité. Les différents segments obtenus et leur degré de visibilité sont listés et classés dans le tableau 4.6. Ce tableau représente le résultat de la requête décrite à l'exemple 1.

Les résultats du tableau 4.6 peuvent être affinés en formulant des requêtes dont les critères sont portés sur d'autres sources d'informations.

Exemple 2. On souhaite réduire le temps de visionnage en sélectionnant parmi les résultats obtenus à l'exemple 1, uniquement les segments vidéo dans lesquels il y a des objets (personnes et/ou véhicules). Les informations sur la présence des objets dans une vidéo sont des métadonnées sémantiques issues des algorithmes de traitement automatique de contenus. Ces métadonnées sont données pour chaque frame de la vidéo (*"enr_103.mp4"*) au tableau 4.7 ("O" si des objets sont présents dans la frame

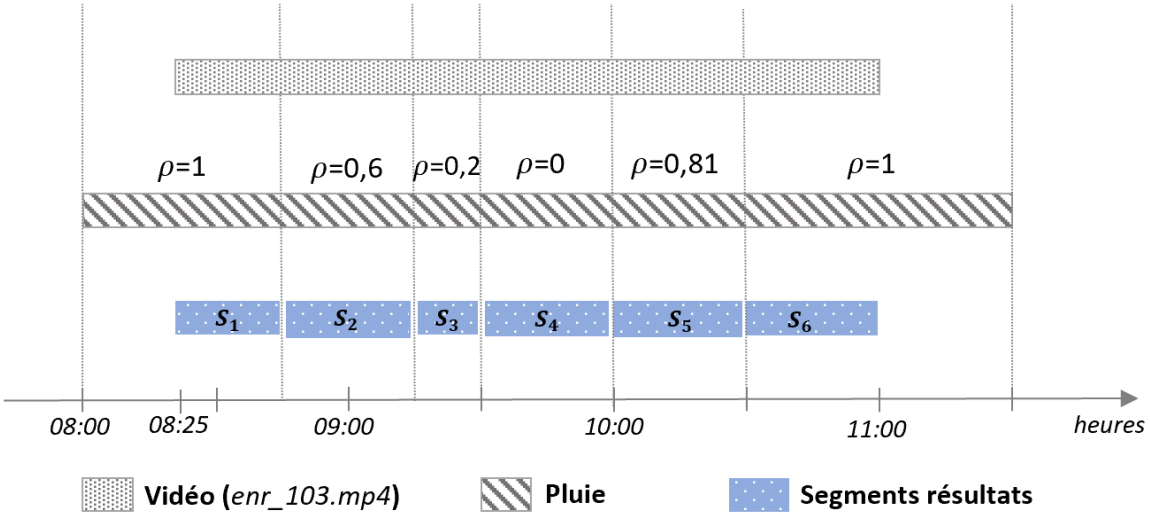


FIGURE 4.16 – Croisement temporel et segmentation de la vidéo.

TABLE 4.6 – Résultat de la requête décrite à l'exemple 1.

Rang	Libellé segment	Segment	Degré de visibilité (ρ)	Description
1	S_1	08h25 - 08h45	1	Visibilité parfaite
2	S_6	10h30 - 11h00	1	Visibilité parfaite
3	S_5	10h00 - 10h30	0.81	Visibilité bonne
4	S_2	08h45 - 09h15	0.6	Visibilité moyenne
5	S_3	09h15 - 09h30	0.2	Visibilité quasi-nulle
6	S_4	09h30 - 10h00	0	Visibilité nulle

et "N" sinon).

TABLE 4.7 – Exemple de métadonnées décrivant la présence des objets dans chaque frame.

Id	Frame	Présence d'objets
0	f0	O
1	f1	O
2	f2	O
...	f...	...
...	f...	...
69216	f69216	N
69217	f69217	N
...	f...	...
279000	f279000	O

Construction de la réponse à la requête de l'exemple 2. Dans cette exemple, la requête n'est pas floue (c.à.d. soit il y a des objets en mouvements dans la vidéo, soit il n'y en a pas). Donc il n'existe pas de fonction d'appartenance pour un quelconque calcul de degré d'imprécision. Grâce aux métadonnées disponibles, les séquences de

frames successives suivantes dans lesquelles il y a des objets sont sélectionnées : f_0 à f_{68400} et f_{171000} à f_{279000}. Ces séquences de frames successives correspondent respectivement aux intervalles de temps (instants de la vidéo) $[08h25, 09h03]$ et $[10h00, 11h00]$. La réponse à la requête s'obtient en faisant un croisement temporel entre ces intervalles de temps et les segments vidéo donnés en entrées (résultats obtenus à l'exemple 1) comme présenté à la Figure 4.17. Le tableau 4.8 décrit les résultats de la requête effectuée à l'exemple 2 (différents segments vidéo sélectionnés).

La Figure 4.18 illustre les résultats obtenus dans les deux exemples comme étant les résultats du mécanisme d'interrogation à 2 niveaux où le premier niveau d'interrogation est basé sur les métadonnées issues des open data, et le deuxième niveau est basé sur les métadonnées sémantiques.

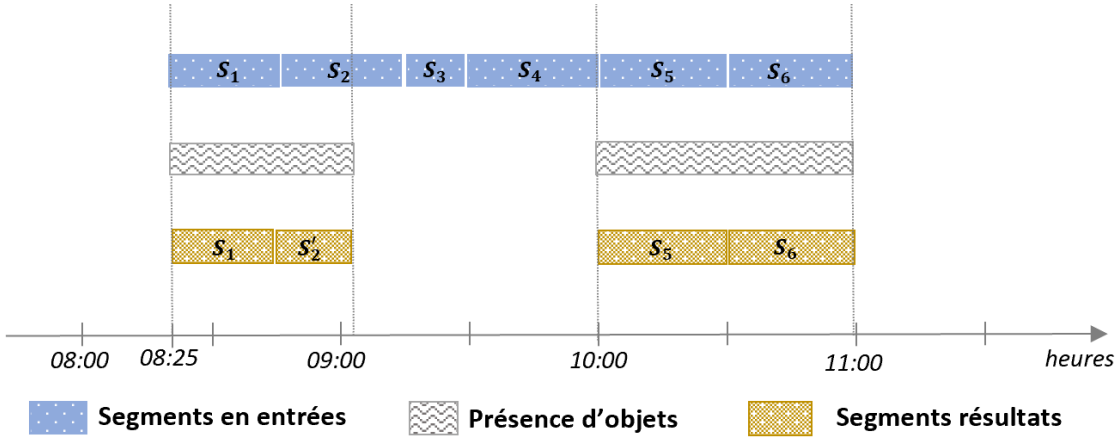


FIGURE 4.17 – Croisement temporel.

TABLE 4.8 – Résultat de la requête décrite à l'exemple 2.

Rang	Libellé segment	Segment	Description
1	S_1	08h25 - 08h45	Satisfaction parfaite
2	S_6	10h36 - 11h00	Satisfaction parfaite
3	S_5	10h00 - 10h35	Satisfaction moyenne
4	S'_2	08h46 - 09h03	Satisfaction moyenne

Formalisation du mécanisme de requêtage. Le mécanisme de requêtage multi-niveaux basé sur les informations contextuelles peut être généralisé par l'organigramme de la Figure 4.19. Ce mécanisme débute par le choix d'une source d'information contextuelle qui constituera le critère du filtrage/interrogation. Si la requête contient des prédicats flous ou implique des données imprécises, il est nécessaire de définir une représentation floue adaptée (exemple de représentation floue de la "visibilité" proposée à la Figure 4.14) afin d'évaluer les degrés de confiance liés à l'imprécision (exemple de la Figure 4.15). Sinon, passer directement au filtrage/interrogation proprement dit des

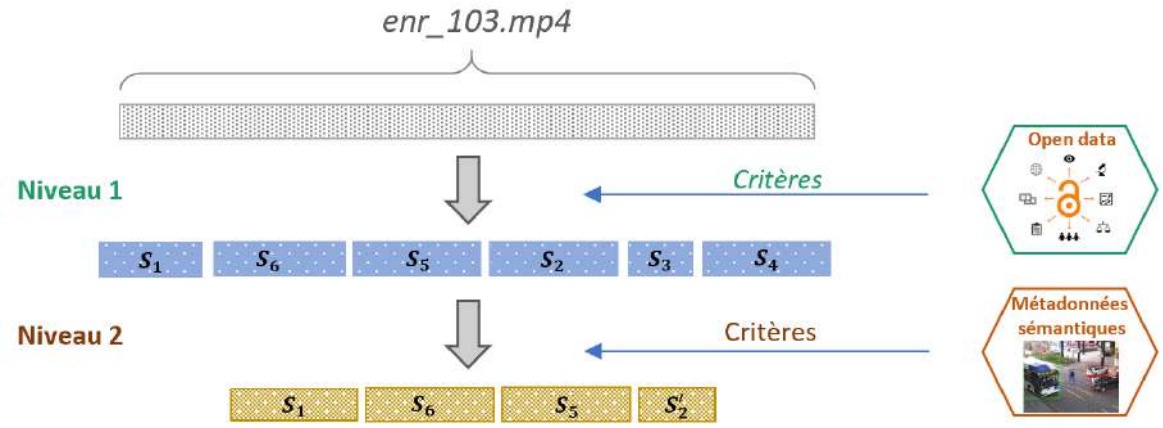


FIGURE 4.18 – Exemple d'interrogation multi-niveaux.

données (vidéos). Le filtrage ou l'interrogation consiste à évaluer les relations temporelles entre les intervalles de temps des vidéos et les informations temporelles liées aux informations contextuelles ou métadonnées (exemple de la figure 4.16). Les résultats du filtrage ou de l'interrogation peuvent être classés en fonction des degrés de confiance liés à l'imprécision. Il est possible de retourner au début du mécanisme et faire le choix d'une nouvelle source d'informations contextuelles afin d'améliorer ces résultats.

L'algorithme 4 implémente le mécanisme de requêtage exécuté à chaque niveau d'interrogation de la Figure 4.13.

Algorithm 4: Querying mechanism

Input: a set of videos : V, a contextual information source src and related membership function f, spatial information loc and temporal information t ;
Output: a set of video segments of interest with related degrees of interest ;
Data: tuple : set(value, degree) ;

```
1 metadata ← medataRetrieval(src, loc, t);
2 tuple ← fuzzyEval(metadata, f);
3 foreach v_i in V do
4 |   result_i ← temporalEval(v_i, tuple) ;
5 |   addResult(result_i);
6 end foreach
```

Cinq paramètres sont définis pour cet algorithme : un ensemble de vidéos à interroger, la source d'information contextuelle sélectionnée et la fonction d'appartenance permettant de calculer les degrés d'imprécision (si interrogation floue), les informations spatiales et temporelles à prendre en compte dans les critères d'interrogation. Les fonctions définies dans l'algorithme sont :

— La fonction $medataRetrieval(src, loc, t)$ retourne l'ensemble des métadonnées de la source src dont les informations spatiales et temporelles sont respectivement comprises dans loc et t. loc délimite la zone concernée par la requête et t

l'intervalle de temps à prendre en compte pour les informations contextuelles de la requête.

— La fonction $fuzzyEval(metadata, f)$ évalue les degrés de précision des métadonnées $metadata$ grâce à la fonction d'appartenance f. Cette fonction retourne pour chaque "$metadata$" un degré de précision (réel) dans l'intervalle $[0, 1]$ (exemple du tableau 4.5).

— La fonction $temporalEval(v_i, tuple)$ évalue les relations temporelles entre les intervalles de temps de la vidéo (v_i) et les informations temporelles liées aux métadonnées de $(tuple)$. Cette fonction retourne un ensemble de segments vidéo et le degré de satisfaction lié à la requête pour chaque vidéo (exemple du résultat du tableau 4.6).

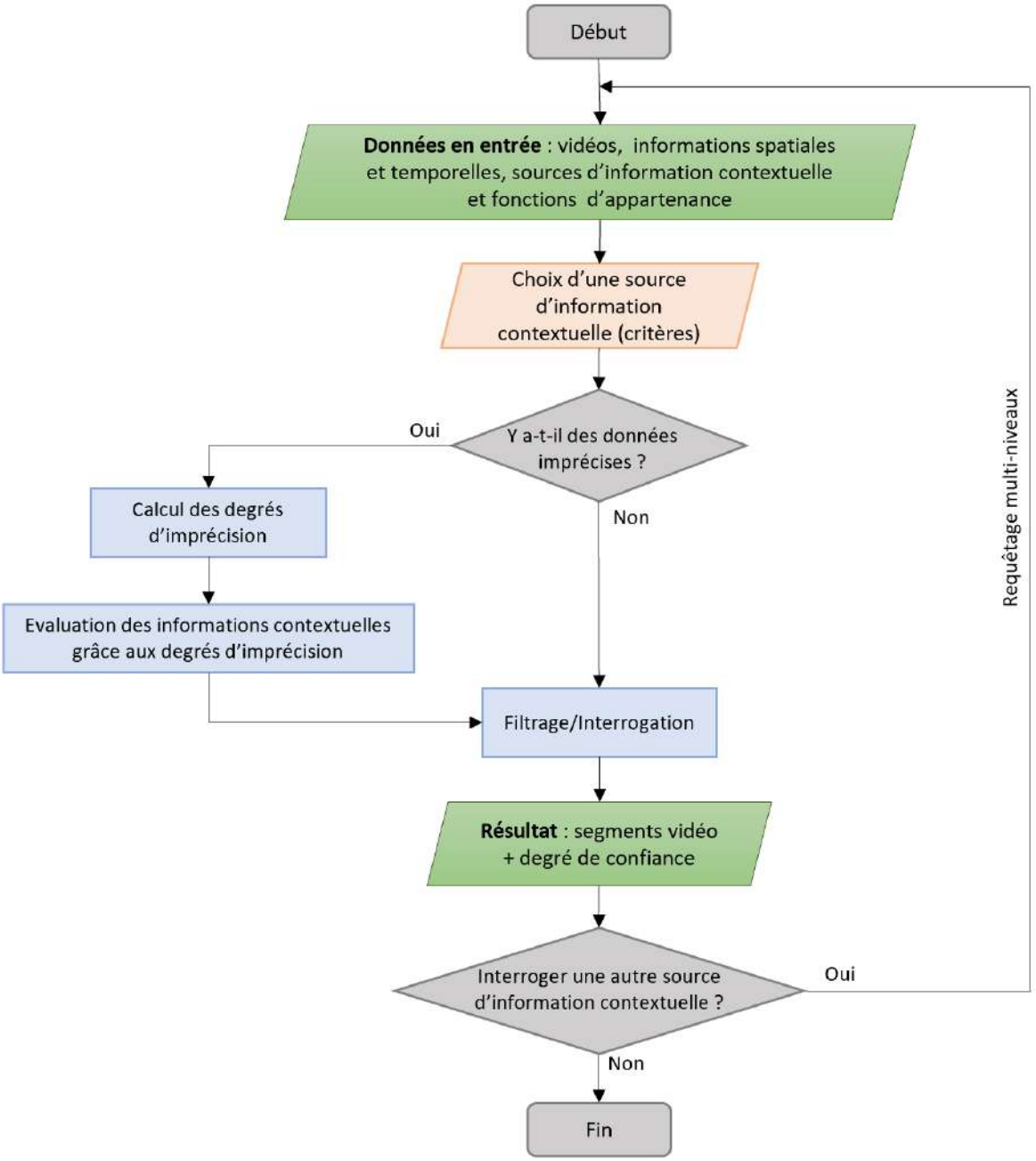


FIGURE 4.19 – Mécanisme de requêtage.

4.4 Conclusion

Il a été démontré par de nombreux chercheurs dans divers domaines que le contexte pourrait être utilisé pour améliorer la performance d'un système. La raison principale pour laquelle l'utilisation du contexte améliore la performance d'un système est l'existence de relations entre les informations contextuelles (données exogènes) et les informations traitées (données dépendantes) par le système. Le concept **d'enrichissement contextuel** développé dans ce chapitre s'appuie sur l'existence des relations entre les

informations contextuelles et les informations liées à un système afin d'améliorer les processus de filtrage et d'interrogation des données du dit système. L'enrichissement contextuel est alors défini comme un processus dont la mise en œuvre comporte un ensemble d'étapes basées sur les informations de contexte.

Les étapes génériques définies pour la mise en œuvre de l'enrichissement contextuel peuvent s'appliquer à divers systèmes. Pour le cas des systèmes de vidéosurveillance traité, différentes sources d'informations contextuelles utiles au filtrage et à l'interrogation de grands volumes de vidéos ont été énumérées et décrites grâce à l'étape d'analyse des informations contextuelles. L'étape de modélisation des informations contextuelles a permis de proposer un modèle générique de métadonnées qui facilite l'intégration de différentes sources d'informations contextuelles et donc l'interopérabilité d'informations contextuelles hétérogènes. L'interopérabilité est l'un des défis majeurs des systèmes qui prennent en compte le contexte. Un autre avantage de l'étape de modélisation est la proposition d'une représentation appropriée des informations contextuelles générées par les évènements dynamiques. Cette représentation permet un stockage "intelligent" des informations contextuelles. Le mécanisme de requêtage proposé à l'étape 3 est robuste aux informations imprécises et incertaines, et permet d'effectuer une interrogation "intelligente" basée sur le contexte. L'introduction de la notion de préférence floue offre la possibilité d'un requêtage flexible et permet donc de mieux répondre aux besoins de l'utilisateur en donnant des réponses approchées, alors qu'un requêtage "stricte" ou "booléen" donnerait un résultat vide, et en permettant de trier les résultats en fonction de leur degré de satisfaction au lieu de fournir des résultats indifférenciés.

Chapitre 5

Contribution à la norme ISO 22311/IEC 79

Différents types de métadonnées liées aux systèmes de vidéosurveillance ont été modélisées dans les deux chapitres précédents de cette thèse, conduisant à la proposition de deux modèles de métadonnées : un modèle générique des métadonnées de qualité et d'utilisabilité/utilisé des vidéos destiné au filtrage négatif, et un modèle générique des métadonnées de contexte de la vidéosurveillance exploitable pour l'interrogation intelligente des vidéos. Un des besoins des acteurs du domaine de la vidéosurveillance, concepteurs de la norme ISO 22311/IEC 79 destinée à l'interopérabilité des systèmes de vidéosurveillance, consiste en la généralisation des approches de modélisation de ces métadonnées. L'objectif d'un tel besoin est de tendre vers la fourniture des outils standards pour le filtrage en "amont" et l'interrogation intelligente des corpus de vidéosurveillance. A cet effet un modèle générique des métadonnées de vidéosurveillance conforme à la norme ISO 22311/IEC 79 est proposé dans ce chapitre. Ce modèle générique intègre les modèles de métadonnées proposés dans les deux chapitres précédents afin de tendre vers une généralisation de l'usage des métadonnées de vidéosurveillance et de fournir un outil d'assistance à la recherche dans les vidéos.

La section 5.1 présente les métadonnées de la norme. Une modélisation générique des métadonnées de vidéosurveillance utile pour le filtrage et l'interrogation des vidéos est proposée à la section 5.2.

5.1 Norme ISO 22311/IEC 79

La norme ISO 22311/IEC 79 vise à mettre en œuvre une interface d'échange des données entre différents systèmes de vidéosurveillance en proposant un format d'export (une structure et un dictionnaire) des données et des métadonnées issues de ces systèmes. Cette norme repose sur une combinaison de plusieurs normes techniques et propose un ensemble de pratiques et d'exigences minimales pour l'interopérabilité des systèmes de vidéosurveillance. Le format d'export de la norme couvre ces exigences d'interpérabilité pour les éléments suivants :

— Vidéo ;

— Audio ;

— Métadonnées : descriptive (identifiant de caméra, localisation, etc.), dynamique (date, temps, résultats d'identification, etc.) ;

— Encapsulation/packaging pour le fichier de sortie ;

— Accès/sécurité et intégrité des données ;

— Dispositions relatives à la vie privée ;

— Données informatives.

Dans cette étude, nous nous focalisons sur les exigences relatives aux métadonnées et un enrichissement potentiel du dictionnaire de métadonnées de la norme.

Dictionnaire des métadonnées de la norme

La norme propose une liste non exhaustive et extensible de métadonnées décrivant et permettant une définition non ambiguë de chaque source (capteur) ou événement audio-vidéo. Le contenu des métadonnées peut être divisé en deux parties : les informations sur les capteurs et les évènements. Ces informations peuvent être obligatoires (afin de garantir le minimum d'interopérabilité), recommandées ou facultatives.

La description des capteurs présentée à la Figure 5.1 est constituée de deux groupes de données : les données statiques comprenant les éléments de description générale (ID capteur, propriétaire, modèle, fabriquant, etc.), et les données dynamiques qui regroupent : les informations temporelles, la localisation (absolue et relative) du capteur, les informations liées au champs de vue du capteur (distance focale, taille du capteur, angle horizontal et vertical, etc.), les informations sur la scène observée (système de coordonnées de l'image, distance oblique, etc.) et les champs libres (Métadonnées expérimentales) prévus pour ajouter d'autres informations.

La description des métadonnées liées à l'évènement est présentée à la Figure 5.2. Les événements sont par nature des données dynamiques. La description proposée fournie des informations générales (ID évènement, Heure début, etc.) et des informations de localisation de l'évènement (absolue et relative). De nouvelles informations peuvent également être ajoutées grâce au champ libre "Métadonnées expérimentales".

Les métadonnées décrites dans le dictionnaire de la norme mettent en évidence le besoin d'interopérabilité des systèmes de vidéosurveillance, et peuvent aussi être utilisées pour des objectifs de recherche et de filtrage des contenus vidéo. Pour cela, le dictionnaire des métadonnées de la norme nécessite des enrichissements.

Dans le section suivante, nous proposons un enrichissement du dictionnaire des métadonnées de la norme afin de prendre en compte les besoins liées au filtrage et l'interrogation des contenus vidéo.

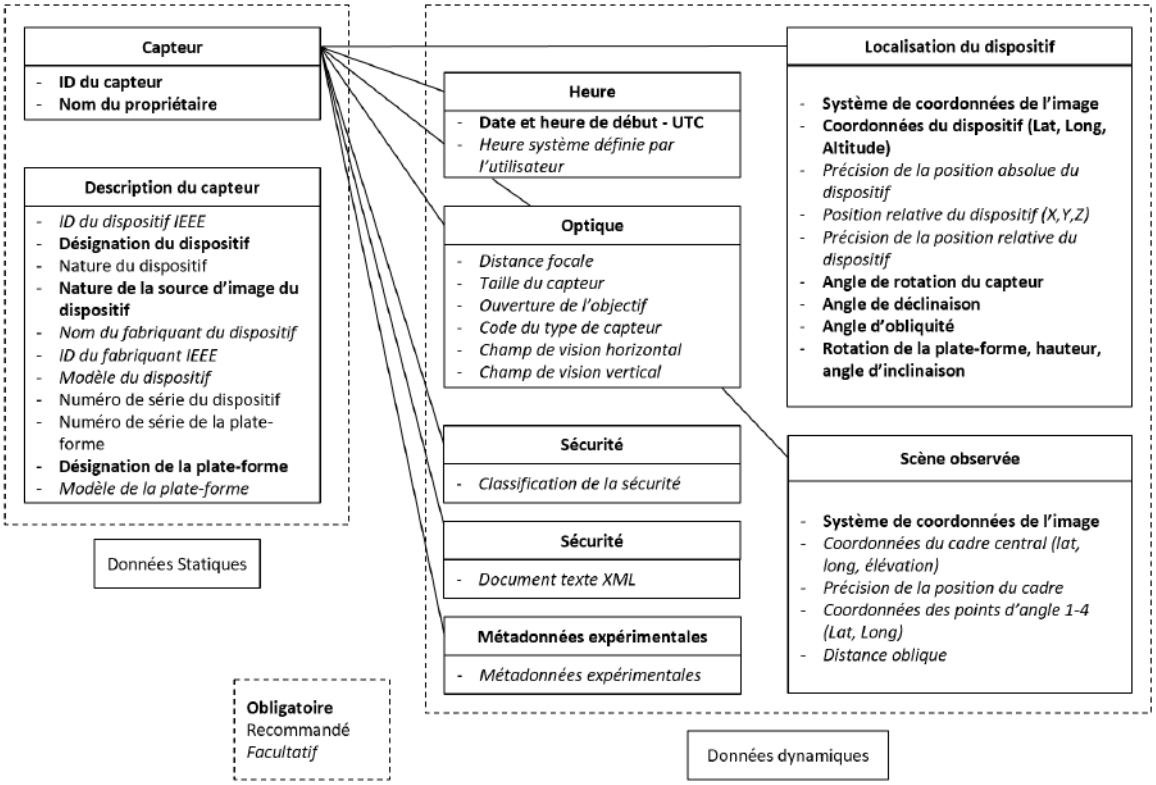


FIGURE 5.1 – Métadonnées liées au capteur.

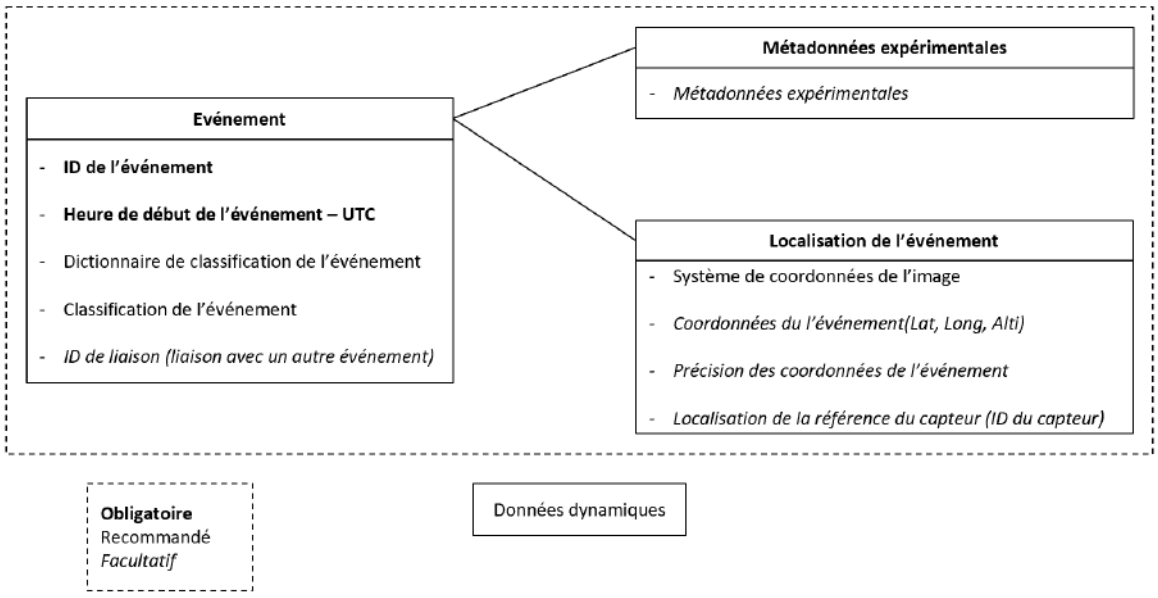


FIGURE 5.2 – Métadonnées liées à l'évènement.

5.2 Proposition d'un modèle générique de métadonnées de vidéosurveillance selon à la norme

Dans un contexte d'augmentation du nombre de systèmes de vidéosurveillance déployés, de leur évolutivité, et des exigences (ex : qualité, utilisabilité, contexte) liés

à leur utilisation, émane un besoin de révision de la norme ISO 22311/IEC 79. Un des aspects de cette révision concerne l'enrichissement du dictionnaire des métadonnées de la norme afin de le rendre le exhaustif possible.

Par conséquent, les métadonnées proposées dans cette thèse et exploitables dans le cadre du projet FILTER2 peuvent être considérées comme une brique de ce processus d'enrichissement des dictionnaires de la norme. Ces métadonnées dont les différentes sources sont présentées à la Figure 5.3, et qui décrivent les informations contextuelles (données exogènes) et les informations relatives à la qualité et à l'utilisabilité/utilité des vidéos, pourront compléter les dictionnaires de la norme afin qu'elle prenne en compte les besoins des applications de vidéosurveillance liés au filtrage "en amont" et à l'interrogation intelligente des contenus vidéo. Nous proposons donc un modèle générique (Figure 5.4) qui permet d'intégrer toutes ces métadonnées et qui supporte le mécanisme de requêtage multi-niveaux proposé au chapitre précédent, ce qui justifierait un mise en œuvre de la norme ISO 22311/IEC 79(via les expérimentations présentées au chapitre 6). Le modèle générique prend en compte les éléments de la norme telles que les métadonnées liées au capteur qui est la caméra (description, localisation, scène observée) et les métadonnées liées aux évènements (description et localisation).

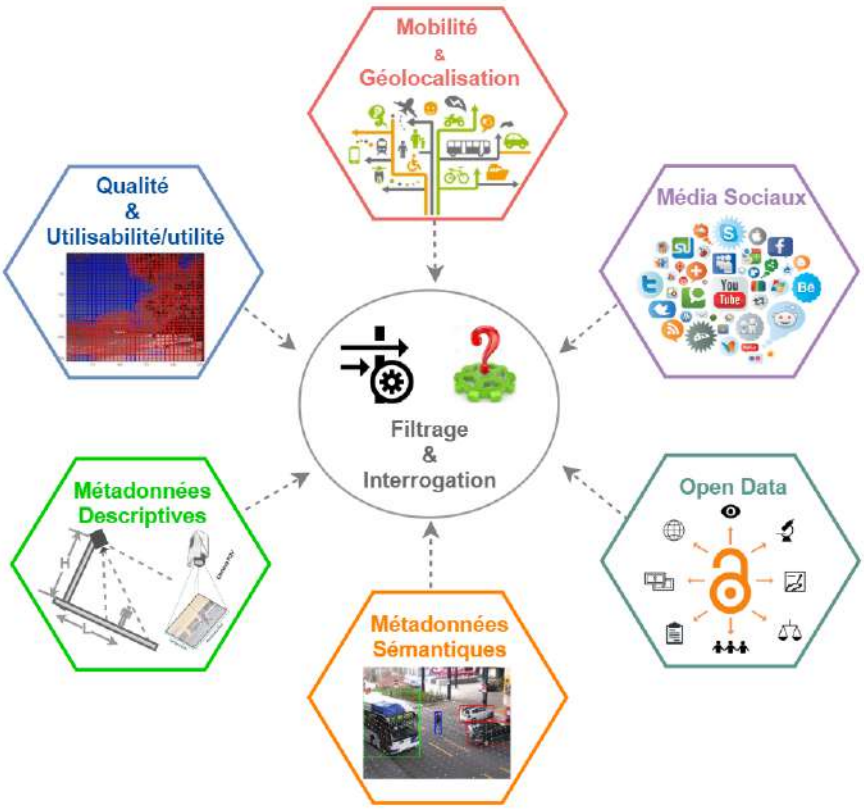


FIGURE 5.3 – Différentes sources de métadonnées.

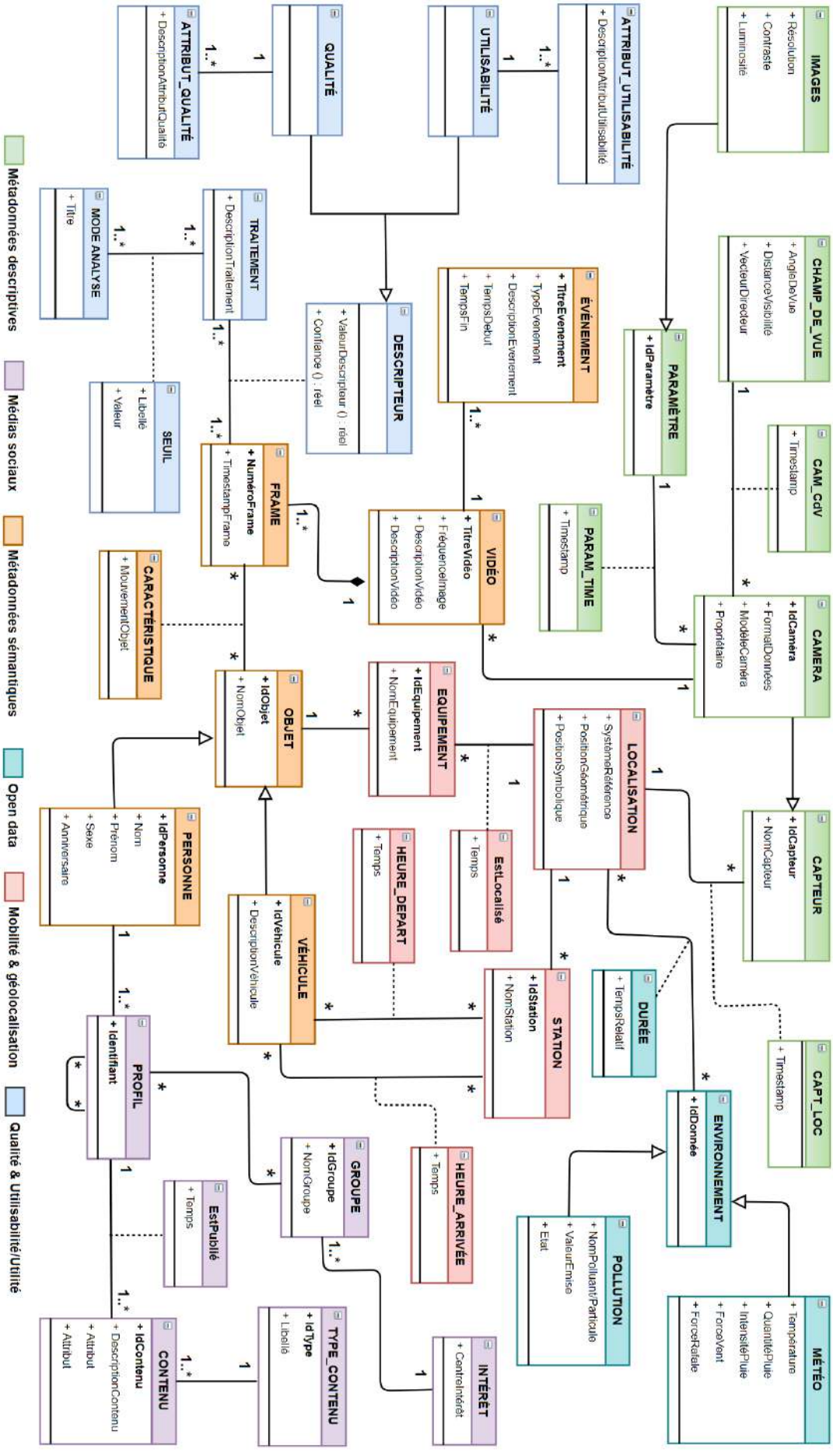


FIGURE 5.4 – Modèle générique des métadonnées.

5.3 Conclusion

Ce chapitre propose une généralisation de la modélisation des métadonnées utiles pour le filtrage et l'interrogation des corpus de vidéo issues des systèmes de vidéosurveillance. Le modèle proposé s'appuie sur les métadonnées de la norme, et propose d'autres types de métadonnées pouvant permettre d'enrichir les dictionnaires de la norme. Toutefois, ce modèle de métadonnées reste extensible, car elle offre la possibilité de définir de nouveaux éléments.

Chapitre 6

Application

Les modèles de données et les algorithmes proposés dans les deux chapitres précédents ont permis de développer un framework de filtrage et d'interrogation des masses de vidéos issues des systèmes de vidéosurveillance. Les approches de filtrage et d'interrogation proposées ont été illustrées par des exemples d'exécution des différents algorithmes supportant les modèles de données. L'objectif de ce chapitre est de présenter le framework développé, et de démontrer sa faisabilité et son utilité grâce à des expérimentations basées sur des données réelles. Les expérimentations réalisées portent sur le gain de temps de traitement et la réduction du temps de visionnage des vidéos.

La première section de ce chapitre présente l'architecture du framework proposé. Les expérimentations menées sont présentées dans la deuxièmes section.

6.1 Architecture du framework proposé

La Figure 6.1 représente l'architecture du framework proposé pour le filtrage et l'interrogation des grands volumes de vidéos. Ce framework est composé de trois grands modules : le module de collecte des métadonnées, le module interface utilisateur, et le module gestion et traitement des métadonnées.

Le prototype a été développé en Java et communique avec une base de données Oracle qui intègre l'extension oracle spatial permettant de stocker les données spatiales et d'effectuer des requêtes spatiales (voir annexe A).

Les sous-sections suivantes présentent en détail les différents modules du framework.

6.1.1 Module de collecte de métadonnées

Les sources de métadonnées prises en compte dans ce travail sont multiples et peuvent être classées en quatre grands groupes : les informations contextuelles, les capteurs, les vidéos et les outils d'analyse vidéo.

Informations contextuelles. La collecte des métadonnées issues des source d'informations contextuelles (open data, médias sociaux, mobilité et géolocalisation) peut se faire grâce à des API et des formats d'échange ou de partage de données (ex : CVS,

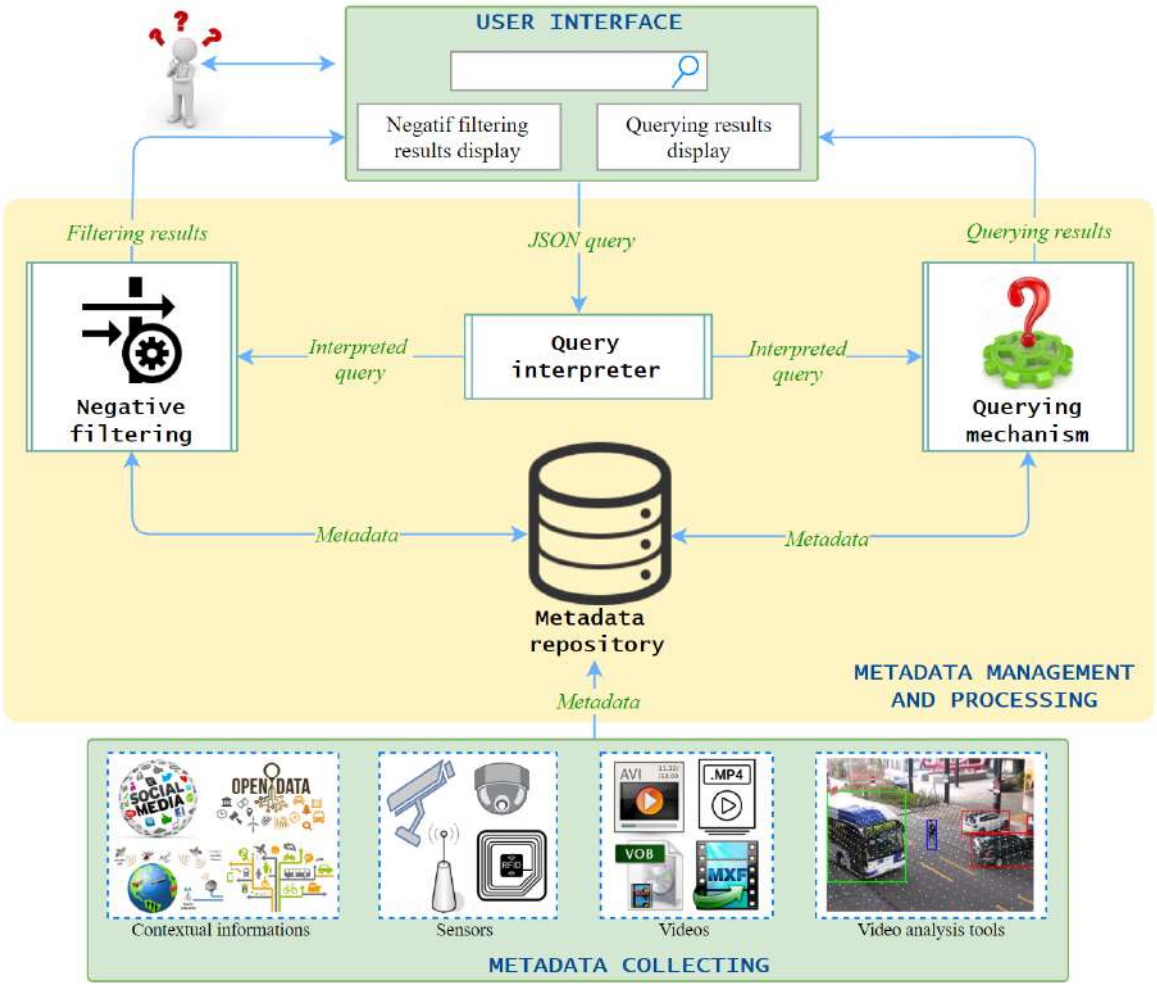


FIGURE 6.1 – Architecture du framework proposé.

JSON, Excel). Dans le cadre de nos expérimentations, les métadonnées collectées sont issues des données ouvertes de météo disponibles sur le portail Data Toulouse Metropole aux formats de fichier JSON, CSV et Excel. Ces métadonnées sont ensuite représentées selon une échelle temps relative (comme illustré à la section 4.3.3) afin de faciliter leur stockage et leur traitement.

Capteurs. Les métadonnées liées aux caméras de vidéosurveillance et aux capteurs de géolocalisation (ex :bornes wifi et lecteurs RFID) ou tout dispositif pouvant être localisé (ex : smartphone) sont d'une grande importance pour un tel framework. Un des problèmes majeurs qui se pose concernant la collecte des métadonnées liées aux capteurs est le format de stockage. Une solution optimale consisterait à modéliser ces métadonnées conformément à la norme ISO 22311/IEC 79 qui offre une structure normalisée des métadonnées liées aux capteurs (voir section 5.1). Le modèle de données permettant d'intégrer les métadonnées liées aux caméras de vidéosurveillance prises en compte dans les expérimentations menées est conforme à la norme 22311/IEC 79.

Vidéos et ***outils d'analyse vidéo***. Dans ce framework, les métadonnées de qualité et d'utilisabilité/utilité de vidéo sont fournies par les partenaires du projet FILTER

2 sur lequel nous avons travaillé. Nous avons extrait les métadonnées sémantiques telles que la présence et le mouvement des objets dans la vidéo grâce aux algorithmes de deep learning tels que YOLO [1].

L'ensemble des métadonnées collectées sont organisées et stockées sous la base des modèles de données proposés.

6.1.2 Module interface utilisateur

L'interface utilisateur permet de définir les requêtes de l'utilisateur ou d'utiliser les requêtes déjà définies dans des fichiers au format JSON, de visualiser les données enregistrées dans la base de données et de visualiser les résultats. La visualisation des données spatiales sur une carte a été faite grâce à l'intégration de l'API Google Maps. Elle peut être utilisée pour visualiser les positions des caméras et leurs champs de vision. Il est possible de définir, modifier et visualiser des trajectoires d'objets mobiles (personnes, véhicules) sur la carte de navigation (voir Figure 6.2). Un autre avantage de l'interface graphique est qu'elle offre la possibilité de visionner les résultats qui sont des extraits vidéo et les annotations (voir Figure 6.3).

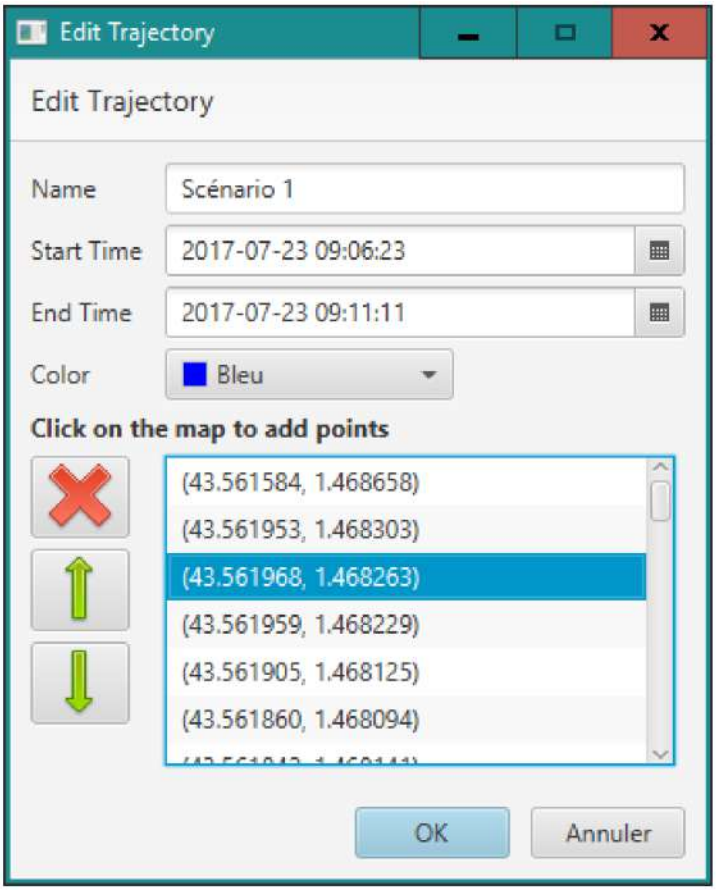


FIGURE 6.2 – Exemple d'une interface de construction de requête.

1. https ://pjreddie.com/darknet/yolo/

FIGURE 6.3 – Visualisation des extraits vidéo.

6.1.3 Module gestion et traitement des métadonnées

Le module de gestion et traitement des métadonnées est la partie la plus importante du framework. Il comporte quatre composants : le stockage des métadonnées, l'interpréteur des requêtes, le filtrage négatif et le mécanisme de requêtage.

Stockage des métadonnées (Metadata repository). Un composant éssentiel du framework est la base de données spatiales. C'est une base de données améliorée et optimisée pour stocker, manipuler et interroger des données reliées à des objets référencés géographiquement, y compris des géométries (points, lignes et polygones). Les améliorations comprennent les types de données spatiales, les fonctions spatiales et les index spatiaux. De plus, la base de données spatiales permet de traiter et d'interroger des données spatiales comme par exemple le point le plus proche, l'affichage des objets dans une zone et le calcul de la distance entre deux objets. Toutes ces opérations peuvent être calculées de manière optimisée à l'aide des index spatiaux. Les métadonnées collectées dans le cadre de nos expérimentations ont été stockées dans une base de données *Oracle Spatial* dont le schéma a été construit grâce aux modèles de métadonnées proposés aux chapitres 3 et 4.

Interpréteur de requête (Query interpreter). Ce composant interprète la requête de l'utilisateur pour la rendre utilisable par le framework. Il prend en entrée une requête JSON. Le template de la requête pour le filtrage négatif est illustré à la Figure 6.4 et est décrit comme suit :

— *"query_location"* indique la zone géographique pour laquelle la requête sera ef-

fectuée. Formellement, elle restreint la requête aux vidéos issues des caméras situées dans la zone donnée. La zone d'intérêt peut être délimitée grâce à une position symbolique ou par une succession de positions géométriques qui forment une géométrie.

— *"query_period"* limite la plage de temps (heure de début et heure de fin) pour la requête.

— *"filtering_mode"* spécifie le mode d'enquête considéré dans la requête et les seuils de compatibilité aux différents traitements dans ce mode d'enquête.

Un exemple de requête JSON formulé grâce à ce template est présenté à la figure 6.5. Dans cette requête, il s'agit d'effectuer le filtrage négatif en mode urgent pour le traitement "détection de véhicule". Les seuils de compatibilité sont présentés dans la requête.

Le template de la requête pour l'enrichissement contextuel est illustré à la Figure 6.6 et est décrit comme suit :

— *"query_location"* représente l'information spatiale, c'est à dire la zone d'intérêt concernée par le requête (ex : une surface).

— *"query_period"* représente l'information temporelle prise en compte dans la requête (heure de début et heure de fin de la requête).

— *"contextual_information_source"* spécifie la source d'informations contextuelles, le paramètre à prendre en compte et la fonction d'appartenance permettant de calculer les degrés d'imprécision dans la requête (s'il y a lieu).

Un exemple de requête JSON formulé grâce à ce template est présenté à la figure 6.7. Pour cette requête, deux sources d'informations contextuelles sont utilisées : les métadonnées sémantiques et les données ouvertes (open data). Une fonction d'appartenance est définie pour le paramètre intensité maximale de pluie et permettra de calculer les degrés d'imprécision dans le mécanisme de requêtage.

Filtrage négatif (Negative filtering). Ce composant implémente les algorithmes de filtrage négatif proposés au chapitre 3 et retourne les résultats à l'utilisateur via le module interface graphique.

Mécanisme de requêtage (Querying mechanism). Le système de requêtage multi-niveaux présenté dans le chapitre 4 est implémenté par ce composant. Les différents niveaux de d'interrogation sont définis par la requête JSON de l'utilisateur.

```
{
"query":
    {
        "Spatio-temporal": [
            {
                "query_location": [
                    {
                    "zone_of_interest":""
                    }
                ],
                "query_period": [
                    {
                    "start":"",
                    "end":""
                    }
                ]
            }
        ],
        "filtering_mode":
        {
            "mode": [
                {
                "title":"",
                "processing": [
                        {
                        "name":"",
                        "Threshold": [
                                {
                                "label":"",
                                "ValMin": ,
                                "ValMax":
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }
}
```

FIGURE 6.4 – Template de requête JSON pour le filtrage négatif.

```
{
"query":{
        "Spatio-temporal":[
            {
                "query_location":[
                    {
                    "zone_of_interest":"Place du Capitole"
                    }
                ],
                "query_period":[
                    {
                    "start":"2017-07-23 09:00:00",
                    "end":"2017-07-23 10:00:00"
                    }
                ]
            }
        ],
        "filtering_mode":
        {
            "mode":
            [
                {
                "title":"urgent",
                "processing":
                    [
                        {
                        "name":"détection de véhicule",
                        "Threshold":[
                                {
                                "label":"seuil compatibilité",
                                "ValMin":0.85,
                                "ValMax":1
                                },
                                {
                                "label":"seuil incompatibilité",
                                "ValMin":0.0,
                                "ValMax":0.84
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }
}
```


FIGURE 6.5 – Exemple de requête JSON pour le filtrage négatif.

```
{
"query": {
        "Spatio-temporal": [
            {
                "query_location": [
                    {
                    "zone_of_interest": ""
                    }
                ],
                "query_period": [
                    {
                    "start": "",
                    "end": ""
                    }
                ]
            }
        ],
        "contextual_information_source":
        [
            {
            "source": "",
            "parameter": "",
            "membership function": [
                    {
                     "title": "",
                     "range": ""
                    }
                ]
            }
        ]
    }
}
```

FIGURE 6.6 – Template de requête JSON pour l'enrichissement contextuel.

```
{
"query":{
    "Spatio-temporal":[
            {
                "query_location":[
                    {
                    "zone_of_interest":"Université Paul Sabatier"
                    }
                ],
                "query_period":[
                    {
                    "start":"2017-07-23 09:00:00",
                    "end":"2017-07-23 10:00:00"
                    }
                ]
            }
        ],
        "contextual_information_source":
        [
            {
            "source":"métadonnées sémantiques",
            "parameter":"mouvement des objets",
            "membership function":[
                    {
                     "title":"",
                     "range":""
                    }
                ]
            },
            {
            "source":"open data",
            "parameter":"intensité maximale de pluie",
            "membership function":[
                    {
                     "title":"visibilité -> parfaite",
                     "range":"[0, 0.3]"
                    },
                    {
                     "title":"visibilité -> imprésice (plus ou moins visible)",
                     "range":"]0.3, 0.65["
                    },
                    {
                     "title":"visibilité -> nulle (impossible de voir)",
                     "range":"[0.65, 1]"
                    }
                ]
            }
        ]
    }
}
```


FIGURE 6.7 – Exemple de requête JSON pour l'enrichissement contextuel.

6.2 Expérimentations et résultats

Dans cette section, nous présentons les résultats des expérimentations menées afin de valider le framework proposé. Ces expérimentations ont été effectuées sur le dataset ToCaDa [Malon et al., 2018], qui contient une collection de vidéos filmées sur le campus de l'Université de Toulouse III - Paul Sabatier. Les tests effectués ont pour but de démontrer la faisabilité de nos propositions.

6.2.1 Présentation du dataset

Une description détaillée du dataset ToCaDa (Toulouse Campus surveillance Dataset) est présentée dans les travaux de l'équipe et du laboratoire [Malon et al., 2018]. Cette sous-section donne un résumé utile à la compréhension de nos expérimentations. Le dataset a été réalisé pour servir aux approches et applications multidisciplinaires telles que la reconstruction de scènes 4D, l'identification et le suivi des objets, la détection d'évènements audio, et la modélisation et l'interrogation de métadonnées multisources. Ce dataset contient deux ensembles de 25 vidéos temporellement synchronisées et correspondant à deux scénarios prédéfinis. Les vidéos ont été filmées le 17 juillet 2017 à 9h50 pour le prémier scénario et à 11h04 pour le deuxième scénario. Les caméras étaient disposées comme suit :

— 9 caméras se trouvaient à l'intérieur du bâtiment principal et filmaient l'extérieur à partir des fenêtres des différents étages. Toutes ces caméras étaient focalisées sur le parking et le chemin menant à l'entrée principale du bâtiment avec de grands champs de vision qui se chevauchent.

— 8 caméras se trouvaient devant le bâtiment, avec de grands champs de vision se chevauchant également (ces 9+8=17 caméras avec des champs de vision qui se chevauchent sont visibles sur la Figure 6.8).

— 8 caméras ont été disposées plus loin, dispersées sur le campus de l'université (voir Figure 6.9). Les champs de vision de ces caméras sont disjoints.

Une vingtaine d'acteurs ont été invités à suivre deux scénarios réalistes en exécutant des actions prédéfinies, comme conduire une voiture, marcher, entrer ou sortir d'un bâtiment, ou tenir un objet à la main pendant le tournage. En plus des actions ordinaires, certains comportements suspects sont présents. Plus précisément :

— Dans le premier scénario, une voiture suspecte (C) avec deux hommes à l'intérieur (D le conducteur et P le passager) arrive et se gare devant le bâtiment (à la vue des caméras dont les champs de vision se chevauchent). P descend de la voiture C et entre dans le bâtiment. Deux minutes plus tard, P quitte le bâtiment en tenant un paquet et monte dans C. C quitte le parking et s'éloigne

FIGURE 6.8 – Bâtiment principal qui regroupe 17 caméras dont les champs de vision se chevauchent [Malon et al., 2018].

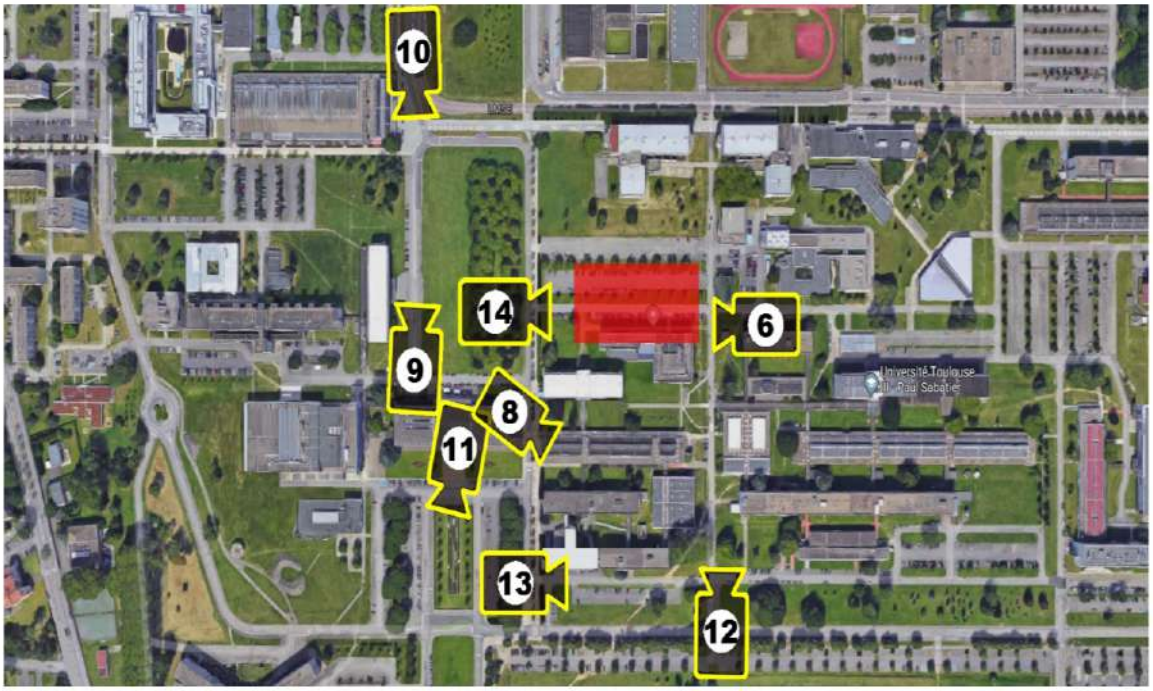


FIGURE 6.9 – Positions des caméras disposées sur le campus de l'Université Paul Sabatier. Ces 8 caméras ont des vues disjointes [Malon et al., 2018]. La zone rouge correspond à la Figure 6.8

du campus universitaire (en passant devant certaines des caméras à champs de vision disjoints).

— Dans le second scénario, la situation est similaire avec une voiture suspecte (C) et deux hommes à l'intérieur (D le conducteur et P le passager) qui arrive et se gare devant le bâtiment (à nouveau à la vue des caméras dont les champs de vision se chevauchent). P descend de C et entre dans le bâtiment. Une minute plus tard, une femme se plaint à D de son mauvais stationnement. C s'en va rapidement et s'arrête dans le champ de vision de la caméra 8. Environ une minute plus tard, P quitte le bâtiment principal en tenant un paquet, et s'enfuit. P rencontre C un peu plus loin (dans le champ de vision de la caméra 8), entre dans C, et C quitte rapidement le campus universitaire (en passant dans les champs de vision de la plupart des caméras).

Dans nos expérimentations, nous ne prenons pas en compte toutes les caméras situées devant et à l'intérieur du bâtiment principal, car elles ont été déployées pour des besoins de reconstruction des scènes 4D, ce qui explique le chevauchement de leurs champs de vision. Parmi ces caméras, nous avons sélectionné trois (caméras 2, 5 et 25 de la Figure 6.8) dont les champs de vision se recoupent et permettent une couverture maximale de la zone souhaitée. Toutes les caméras de la Figure 6.9 sont prises en compte dans les expérimentations. Les caméras 2, 5, 25 de la Figure 6.8 ont respectivement pour identifiant $camA$, $camB$, $camC$, et les caméras 6, 8, 9, 10, 11, 12, 13, 14 de la Figure 6.9 ont respectivement pour identifiant $camD$, $camE$, $camF$, $camG$, $camH$, $camI$, $camJ$, $camK$. Chaque caméra a filmé au total pendant 10 minutes 28 secondes (4 minutes 48 secondes pour le scénario 1 et 5 minutes 40 secondes pour le scénario 2). Les vidéos issues des 11 caméras ($camA$ à $camK$) ont respectivement pour identifiant V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}. Donc dans les expériences, chaque vidéo V_i est constituée des 4 minutes 48 secondes du premier scénario, et des 5 minutes 40 secondes du deuxième scénario, c'est à dire une vidéo couvre les intervalles de temps [09 :50 :00 - 09 :54 :48] et [11 :04 :00 - 11 :09 :40].

6.2.2 Expérience 1 - Filtrage négatif

Les tests effectués ont pour but de mesurer l'impact du filtrage négatif proposé sur le temps de traitement des vidéos par les algorithmes de traitement automatique. Dans cette expérience nous exécutons nos algorithmes de filtrage négatif sur les vidéos issues des onze caméras retenues pour les expérimentations.

6.2.2.1 Mise en place de l'expérience

Afin de tester les algorithmes proposés pour le filtrage négatif, nous considérons les spécifications du projet FILTER 2, c'est à dire trois traitements (détection des véhicules, détection de visage, détection et lecture des plaques d'immatriculation), deux modes d'analyse (urgent et approfondi), et un filtrage selon les critères de qualité.

Les métadonnées d'utilisabilité/utilité n'étant pas disponibles, les critères d'utilisabilité/utilité ne sont pas pris en compte. Pour cette expérience, les vidéos issues des caméras sélectionnées étant de bonne qualité, nous avons dégradé de manière aléatoire certains segments vidéo afin de filtrer sur des critères de qualité. Nous avons effectué trois types de dégradations : la *dégradation A* (Figure 6.10 (a)) consistait à ajouter des pixels, la *dégradation B* (Figure 6.10 (b)) consistait à assombrir, et la *dégradation C* (Figure 6.10 (c)) consistait à ajouter du flou. Les trois dégradations ont été faites dans le but d'évaluer la qualité d'image pour trois types de traitements retenus : *dégradation A* pour le traitement détection et lecture automatique des plaques (DLAP), *dégradation B* pour le traitement détection de véhicule, et *dégradation C* pour le traitement détection de visage.

La qualité de chaque frame a été évaluée en utilisant la métrique BRISQUE (Blind Referenceless Image Spatial Quality Evaluator) [Mittal et al., 2011]. La métrique BRISQUE représente la métrique la plus régulièrement utilisée dans l'analyse de qualité d'image (Image Quality Assessment) sans référence [Charrier et al., 2015], c'est-à-dire ne nécessitant pas d'image de référence possédant une bonne qualité. L'extraction de cette métrique pour chaque frame de la vidéo dégradée a été implémentée en Python à l'aide du module "pybrisque" proposé par Kushashwa Ravi Shrimali (sous licence MIT) [2]. Les métadonnées de qualité extraites ont été sauvegardées dans la base de données. Un extrait des métadonnées de la base est présenté à la Figure 6.11.

6.2.2.2 Paramètres de l'expérience

La localisation, l'information temporelle, et les seuils de compatibilité des vidéos avec les traitements pour différents modes d'analyse sont des paramètres importants pour le filtrage négatif.

Localisation - Elle délimite une zone d'intérêt pour la recherche. On s'intéresse aux caméras de surveillance installées dans cette zone. Il s'agit dans cette expérience de "l'Université Paul Sabatier" (représentée par une surface).

Information temporelle - Elle représente les intervalles de temps à prendre en compte dans les vidéos générées par les caméras situées dans la zone d'intérêt définie par la localisation. Dans cette expérience, il s'agit des intervalles de temps suivants : le "17 juillet 2017 de 9h50 à 9h55 (scénario 1) et de 11h04 à 11h10 (scénario 2)".

Seuils de compatibilité - Comme défini dans le chapitre 3, le seuil de compatibilité est un intervalle dans lequel un traitement est applicable à une frame ou ensemble de frames pour un mode d'analyse donné. Dans le processus du filtrage négatif, des seuils de compatibilité sont définis pour chaque traitement en fonction du mode d'analyse. Dans cette expérience les seuils définis pour les métriques de qualité

2. https ://github.com/krshrimali/No-Reference-Image-Quality-Assessment-using-BRISQUE-Model

(a) Image originale

(b) Dégradation A

(c) Dégradation B

(d) Dégradation C

FIGURE 6.10 – Dégradation des images

d'image utilisées sont présentés à la Figure 6.12. La qualité d'image (Q) pour une frame est une valeur appartenant à l'intervalle $[0, 100]$. Plus cette valeur est grande, plus l'image est de bonne qualité.

Les opérateurs de vidéosurveillance peuvent par la suite paramétrer les seuils selon

VIDEOID	FRAMEID	FRAMENUM	TIMEUTC	DETECTION_VEHICULE	DETECTION_VISAGE	DLAP
V1	V1F4095	4095	23/07/17 07:08:39,500000000	59,6219105278966	85,4408931412862	57,6679826884258
V1	V1F4096	4096	23/07/17 07:08:39,533333000	58,504716191589	85,115146196822	58,7888829614312
V1	V1F4097	4097	23/07/17 07:08:39,566667000	57,4669176574981	84,6482308401267	59,9498595768566
V1	V1F4098	4098	23/07/17 07:08:39,600000000	56,0527790844348	85,0288223665784	59,883947692636
V1	V1F4099	4099	23/07/17 07:08:39,633333000	56,2497356965584	84,9850771970943	59,9543414719667
V1	V1F4100	4100	23/07/17 07:08:39,666667000	58,3374726145051	85,4191208250749	60,1993336348336
V1	V1F4101	4101	23/07/17 07:08:39,700000000	59,3546508548222	85,2637439723441	61,6516605741563
V1	V1F4102	4102	23/07/17 07:08:39,733333000	59,9272701152066	85,4324580412071	61,3192339195775
V1	V1F4103	4103	23/07/17 07:08:39,766667000	59,7056207262156	85,2456878694835	61,6787469268718
V1	V1F4104	4104	23/07/17 07:08:39,800000000	58,775555899935	85,2746694129655	61,8391136438235
V1	V1F4105	4105	23/07/17 07:08:39,833333000	59,0715896349354	85,7363089776969	61,6460731833235
V1	V1F4106	4106	23/07/17 07:08:39,866667000	58,6468062151391	85,8228501444102	61,3704735227811
V1	V1F4107	4107	23/07/17 07:08:39,900000000	57,3217471635015	85,3212681839721	60,7187890811646
V1	V1F4108	4108	23/07/17 07:08:39,933333000	56,7095889112779	85,4752245968936	59,5646601977443
V1	V1F4109	4109	23/07/17 07:08:39,966667000	57,1306273497405	85,8345307695921	58,3154680634796
V1	V1F4110	4110	23/07/17 07:08:40,000000000	57,1765276072144	85,9913307411558	57,8541177876096
V1	V1F4111	4111	23/07/17 07:08:40,033333000	56,5393437254449	85,2190483046719	59,1072191362284
V1	V1F4112	4112	23/07/17 07:08:40,066667000	56,693312931765	85,2547321050339	60,0784315219526

FIGURE 6.11 – Extrait de la base de données des métadonnées de qualité.

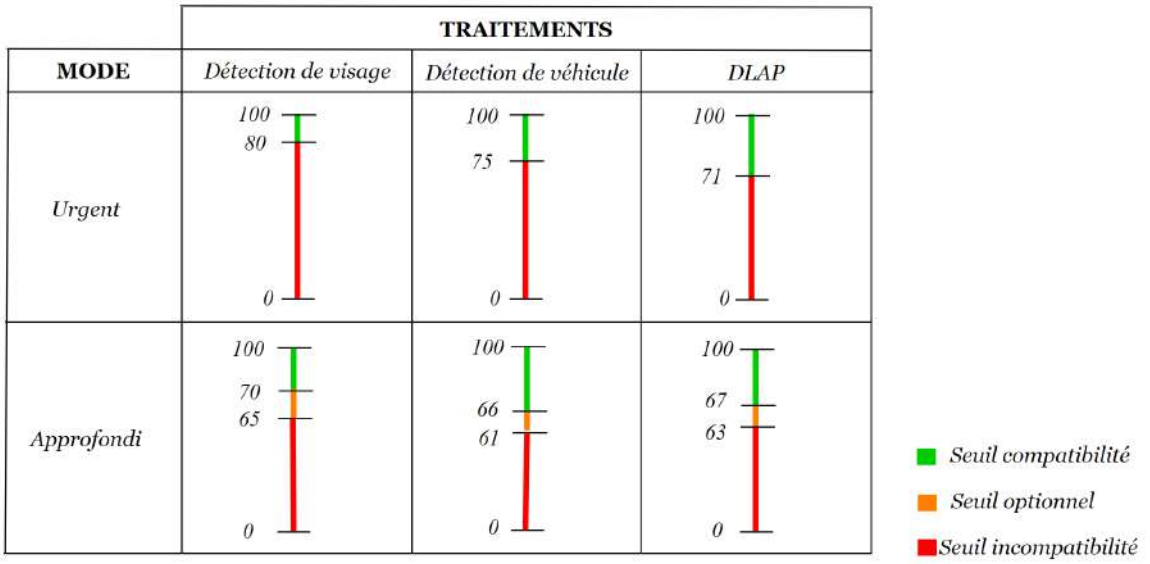


FIGURE 6.12 – Seuils de compatibilité définis pour l'expérience.

leurs besoins en termes de rapidité de traitement. Par conséquent, définir un seuil "générique" de compatibilité d'un traitement avec un mode d'analyse dévient complexe, car il est difficile de prédire la modification du seuil initiale par un opérateur. Comme perspective, nous envisageons d'utiliser des méthodes de machine learning se basant sur les interactions des utilisateurs pour apprendre et prédire, de manière générique ou personnalisée, les comportements les plus adaptés aux différentes situations (pour chaque traitement dans différents modes d'analyse).

6.2.2.3 Résultats et interprétations

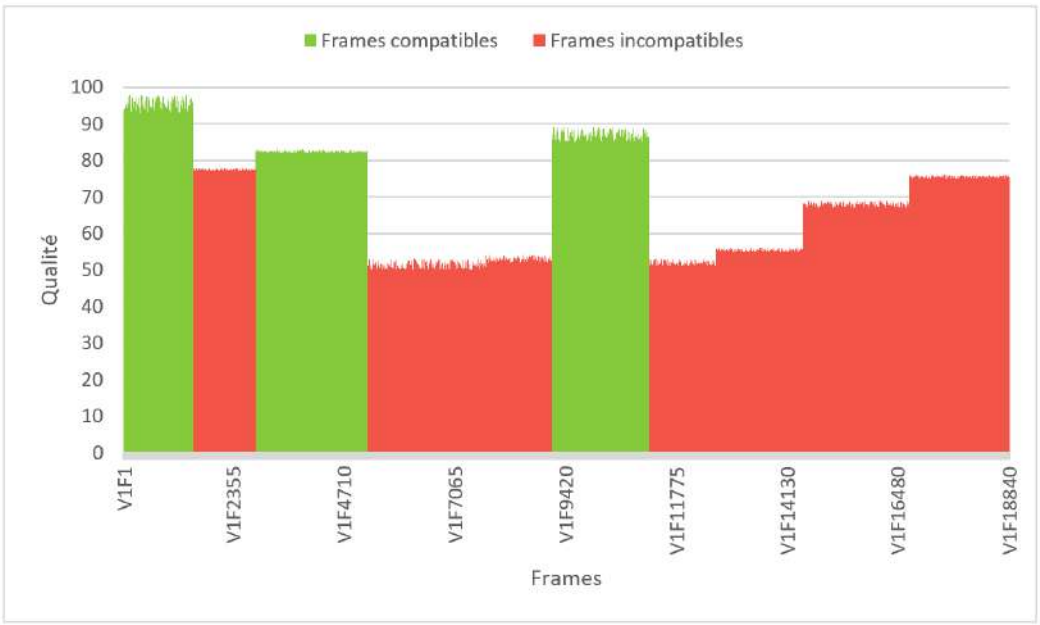
La requête JSON exécutée dans cette expérience est présentée à la Figure 6.13.

Le filtrage négatif pour une vidéo donnée, renvoie pour chaque frame de la vidéo sa compatibilité (*compatible, optionnel, incompatible*) avec chaque traitement (*détec-*

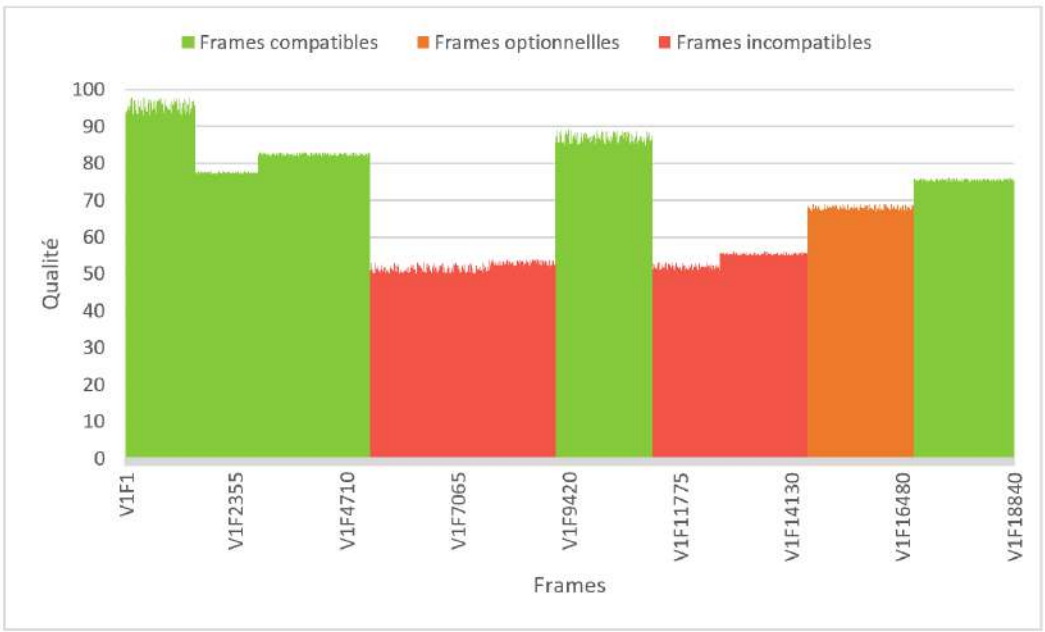
```
{
"query":
    {
        "Spatio-temporal":[
            {
            "query_location":[{"zone_of_interest":"Université Paul Sabatier"}],
            "query_period":[
                    {"start":"2017-07-17 09:50:00", "end":"2017-07-17 09:55:00"},
                    {"start":"2017-07-17 11:04:00", "end":"2017-07-17 11:10:00"}]
            }
        ],
        "filtering_mode":
        {
            "mode":[
                {
                "title":"Urgent",
                "processing":[
                        {
                        "name":"Détection de visage",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":80, "ValMax":100},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":79}]
                        },
                        {
                        "name":"Détection de véhicule",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":75, "ValMax":100},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":74}]
                        },
                        {
                        "name":"Détection et lecture automatique de plaques",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":71, "ValMax":100},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":70}]
                        }
                    ]
                },
                {
                "title":"Approfondi",
                "processing":[
                        {
                        "name":"Détection de visage",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":70, "ValMax":100},
                                {"label":"Seuil optionnel", "ValMin":65, "ValMax":69},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":64}]
                        },
                        {
                        "name":"Détection de véhicule",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":66, "ValMax":100},
                                {"label":"Seuil optionnel", "ValMin":61, "ValMax":65},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":60}]
                        },
                        {
                        "name":"Détection et lecture automatique de plaques",
                        "Threshold":[
                                {"label":"Seuil compatibilité", "ValMin":67, "ValMax":100},
                                {"label":"Seuil optionnel", "ValMin":63, "ValMax":66},
                                {"label":"Seuil incompatibilité", "ValMin":0, "ValMax":62}]
                        }
                    ]
                }
            ]
        }
    }
}
```


FIGURE 6.13 – Requête JSON pour le filtrage négatif.

tion de visage, détection de véhicule, détection et lecture automatique des plaques) en fonction des différents modes d'analyse (*urgent, approfondi*). Les Figures 6.14, 6.15, et 6.16 représentent respectivement le filtrage négatif de la vidéo V_1 pour les trois types de traitement en fonction de chaque mode d'analyse. Sur ces figures, les frames de couleur verte sont *compatibles*, les frames de couleur orange sont *optionnel*, et les frames de couleur rouge sont *incompatibles* avec le traitement choisi dans le mode d'analyse donné.



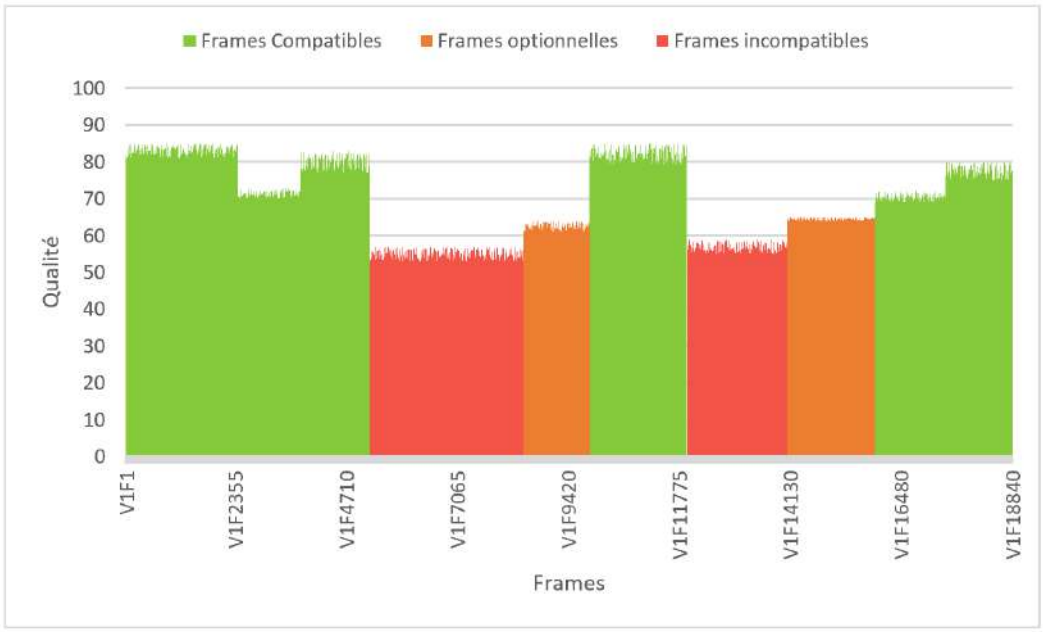
(a) Mode urgent.



(b) Mode approfondi.

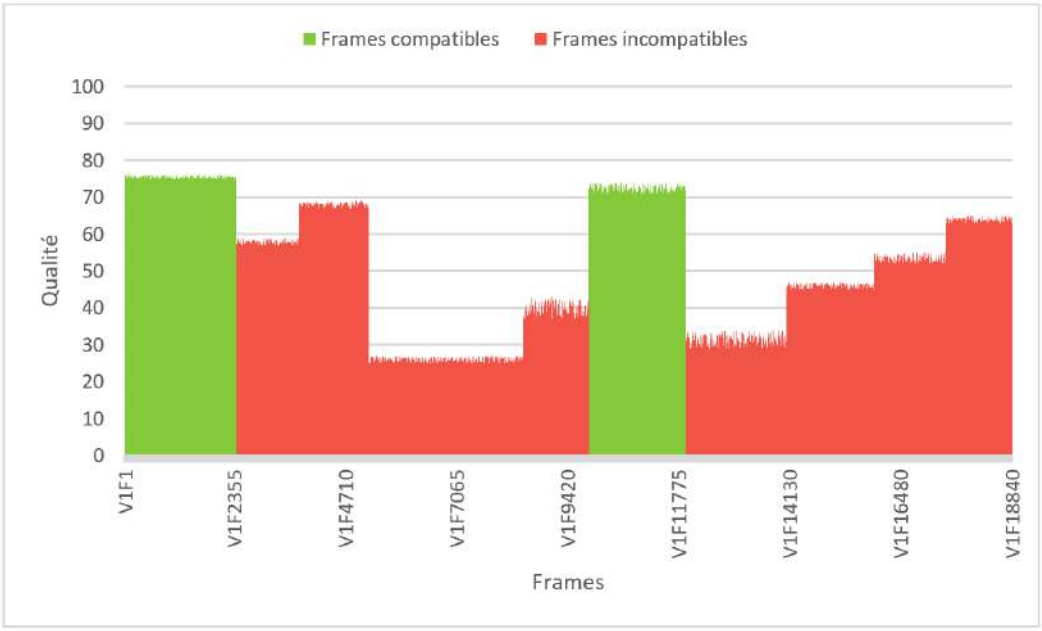
FIGURE 6.14 – Détection de visage pour la vidéo V_1.

(a) Mode urgent.

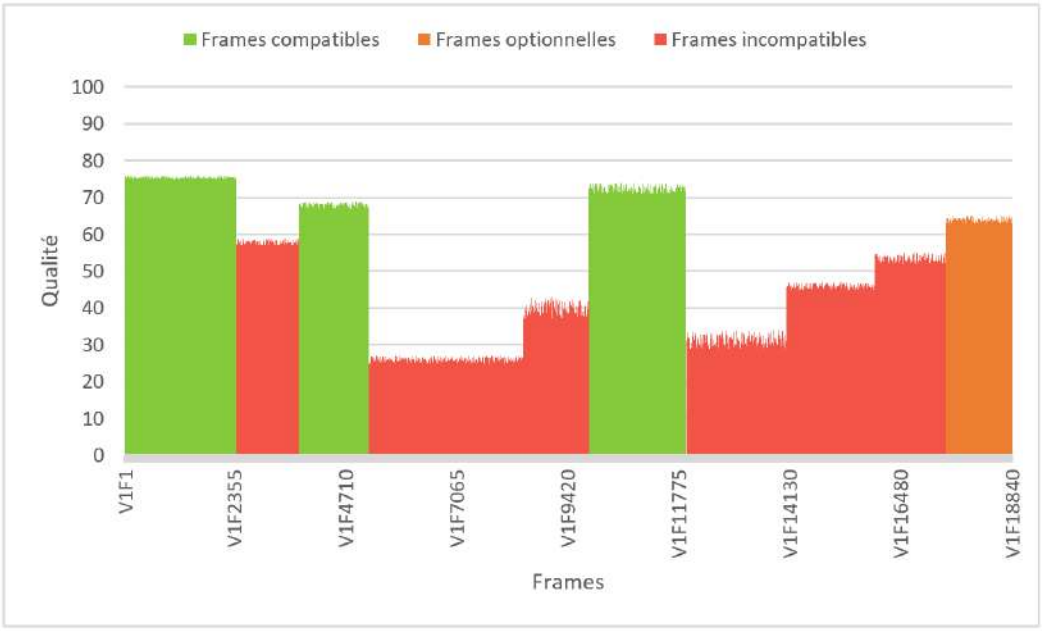


(b) Mode approfondi.

FIGURE 6.15 – Détection de véhicule pour la vidéo V_1.



(a) Mode urgent.



(b) Mode approfondi.

FIGURE 6.16 – Détection et lecture automatique des plaques pour la vidéo V_1.

Le résultat du filtrage négatif est retourné sous forme de segments vidéo constitués de frames successives de même couleur. Par exemple, pour le traitement *"Détection de visage"*, le résultat du filtrage négatif de la vidéo V_1 est :

— **Mode urgent** :
  — Segments compatibles : $[V_1F_1,\ V_1F_{1963}]$, $[V_1F_{2747},\ V_1F_{5031}]$, et $[V_1F_{9274},\ V_1F_{11383}]$.
  — Segments incompatibles : $[V_1F_{1964}, V_1F_{2746}]$, $[V_1F_{5032}, V_1F_{9274}]$, et $[V_1F_{11384},\ V_1F_{18840}]$.

— **Mode approfondi** :

— Segments compatibles : $[V_1F_1, V_1F_{4927}]$, $[V_1F_{9275}, V_1F_{11561}]$, et $[V_1F_{16532}, V_1F_{18840}]$.

— Segments optionnels : $[V_1F_{14242}, V_1F_{16531}]$.

— Segments incompatibles : $[V_1F_{4928}, V_1F_{9274}]$, et $[V_1F_{11562}, V_1F_{14241}]$.

Les résultats du filtrage négatif pour les autres vidéos sont représentés de façon similaire.

6.2.2.4 Évaluation

Dans cette section, nous évaluons le **gain de temps de traitement** obtenu en utilisant l'approche proposée. Ce gain peut être calculé pour chaque traitement dans un mode d'analyse donné.

Les graphes des Figures 6.17, 6.18, et 6.19 montrent les temps de vidéo à exploiter après le filtrage pour chaque traitement dans les différents modes d'analyse. La durée totale de la vidéo initiale ($10min28sec = 628sec$) est affichée pour chaque graphe. La Figure 6.17 montre le résultat pour le mode urgent, la Figure 6.18 montre le résultat dans le meilleur des cas (c'est à dire sans prendre en compte les segments vidéo optionnels) pour le mode approfondi, et la Figure 6.19 montre le résultat dans le pire des cas (c'est à dire en tenant compte des segments vidéo optionnels) pour le mode approfondi.

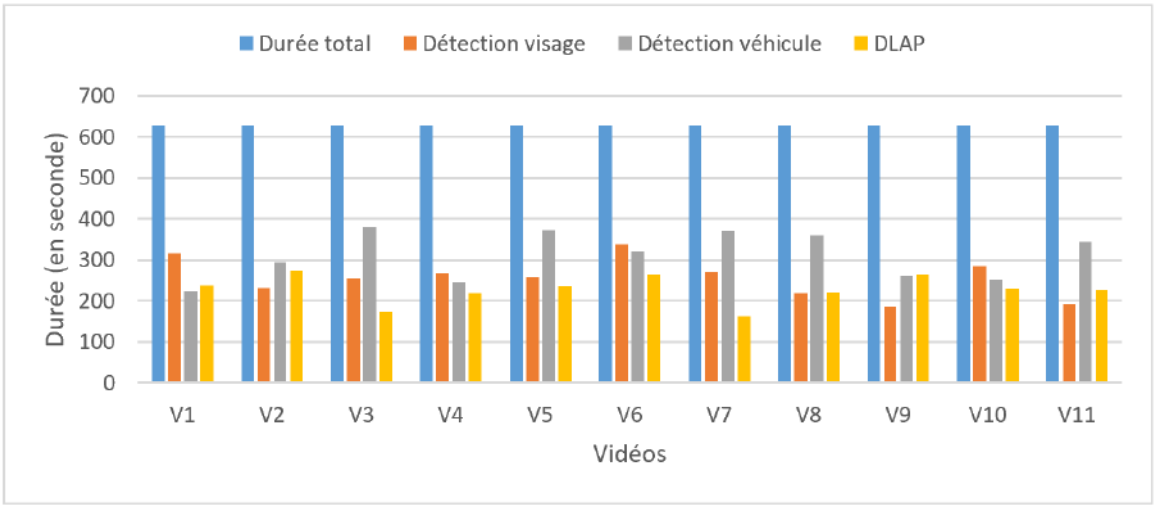


FIGURE 6.17 – Durée de vidéo à exploiter après filtrage pour chaque traitement dans le mode urgent.

Le gain de temps G_{temps} pour une vidéo donnée représente le rapport du temps total de traitement de la vidéo sans filtrage T_{total} sur le temps total de traitement de la vidéo avec filtrage $T_{approche}$.

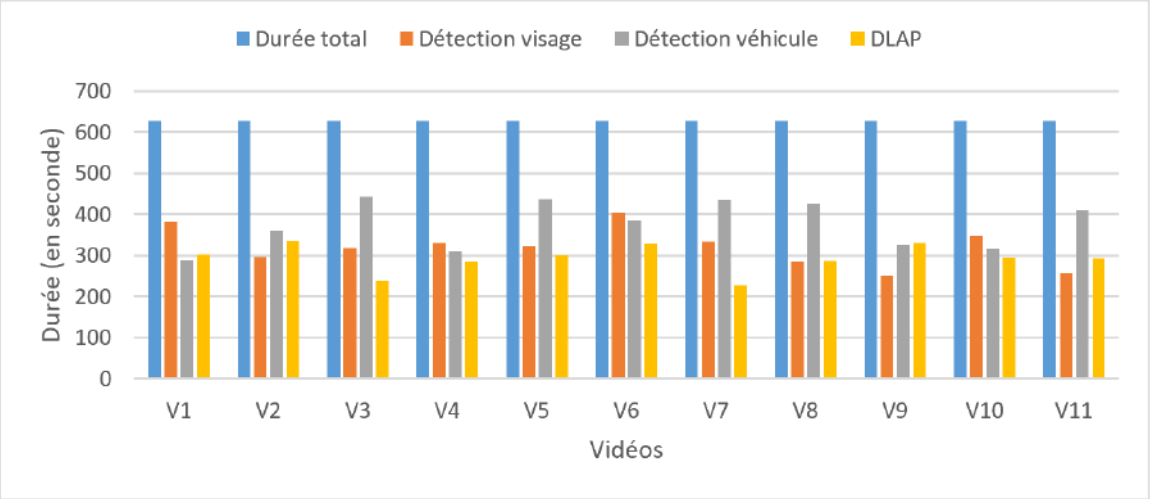


FIGURE 6.18 – Durée de vidéo à exploiter (meilleur des cas) après filtrage pour chaque traitement dans le mode approfondi.

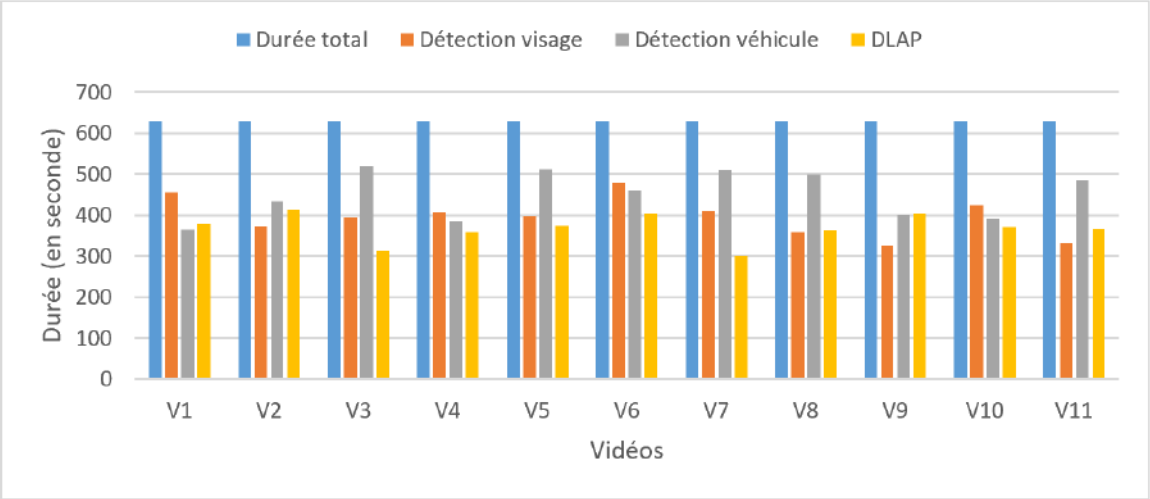


FIGURE 6.19 – Durée de vidéo à exploiter (pire des cas) après filtrage pour chaque traitement dans le mode approfondi.

$$G_{temps} = \frac{T_{total}}{T_{approche}} \tag{6.1}$$

où $T_{total} = n_{total} * t_{traitement}$ et $T_{approche} = (n_{total} * t_{filtrage}) + (n_{reste} * t_{traitement})$, avec :

— n_{total} est le nombre de frame à traiter avant filtrage,

— n_{reste} est le nombre de frame à traiter après filtrage,

— $t_{traitement}$ est le temps de traitement d'une frame par un algorithme de traitement automatique (ex : détection de véhicule),

— $t_{filtrage}$ est le temps de filtrage d'une frame par notre algorithme de filtrage négatif.

L'équation 6.1 devient :

$$G_{temps} = \frac{n_{total} * t_{traitement}}{(n_{total} * t_{filtrage}) + (n_{reste} * t_{traitement})} \tag{6.2}$$

Nous avons évalué le gain de temps de calcul pour les différents traitements en utilisant un ordinateur possédant les caractéristiques logicielles suivantes :

— système d'exploitation : Windows 10 Professionnel 64 bits ;

— processeur : CPU Core i7, 2.50GHz 2.50GHz ;

— RAM : 16Go.

Les algorithmes de détection utilisés sont ceux de YOLO [3], dont la vitesse de traitement est en moyenne 0.025 $frame/s$ (avec notre configuration logicielle).

Les gains de temps pour les traitements "détection de visage", "détection de véhicules", et "détection et lecture automatique des plaques" correspondant à l'ensemble des 11 vidéos prises en compte dans cette expérience sont respectivement présentées aux figures 6.20, 6.21, et 6.22.

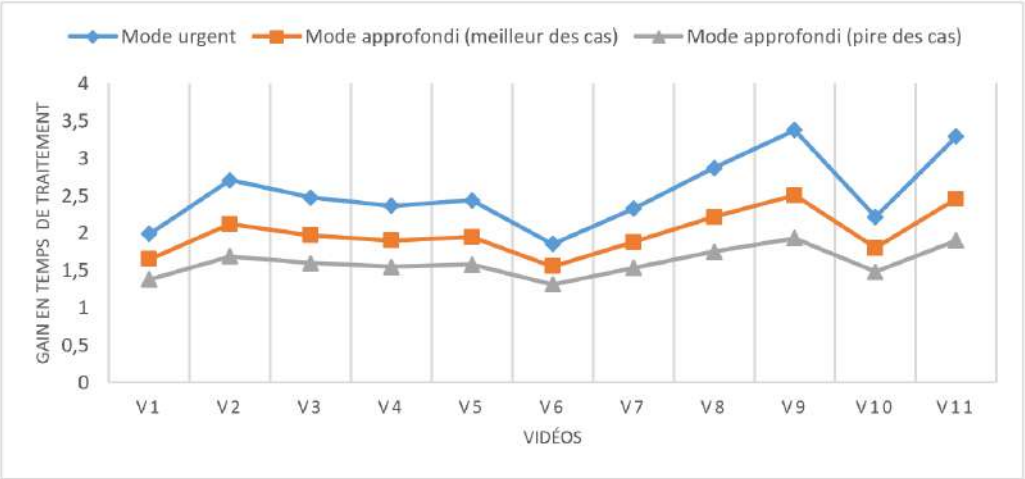


FIGURE 6.20 – Gain de temps pour le traitement détection de visage.

Ces gains de temps sont calculés pour chaque mode d'analyse : urgent et approfondi("meilleur des cas" signifie que les segments vidéo optionnels ne sont pas pris en compte, et "pire des cas" signifie que ces segments sont pris en compte).

Prenons l'exemple de la vidéo *"V4"* à la Figure 6.20 : les gains de temps pour le traitement "détection de visage" en mode urgent, approfondi au meilleur des cas et approfondi au pire des cas sont respectivement *2.5*, *2*, et *1.5*. Cela signifie que le filtrage négatif dans ces modes respectifs a permis de diviser le temps de traitement par *2.5*, *2*, et *1.5*.

3. https ://pjreddie.com/darknet/yolo/

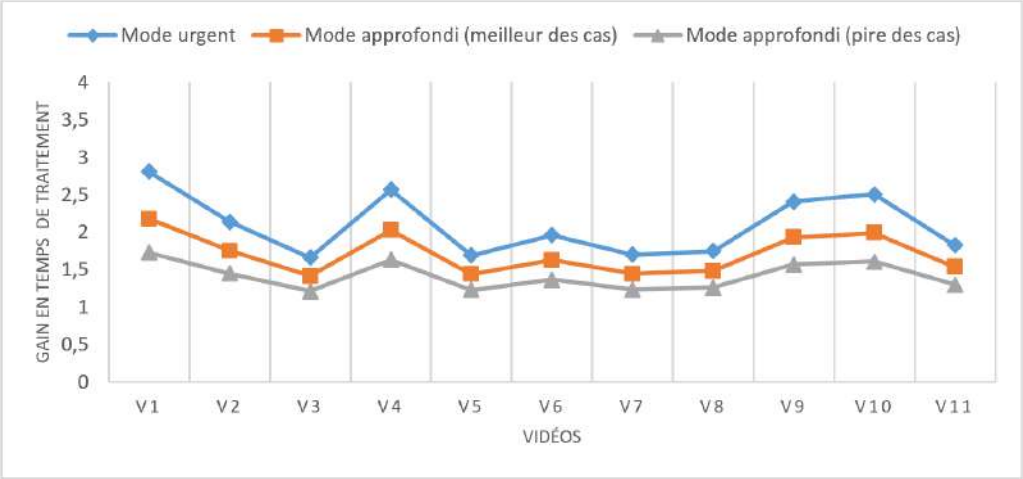


FIGURE 6.21 – Gain de temps pour le traitement détection de véhicule.

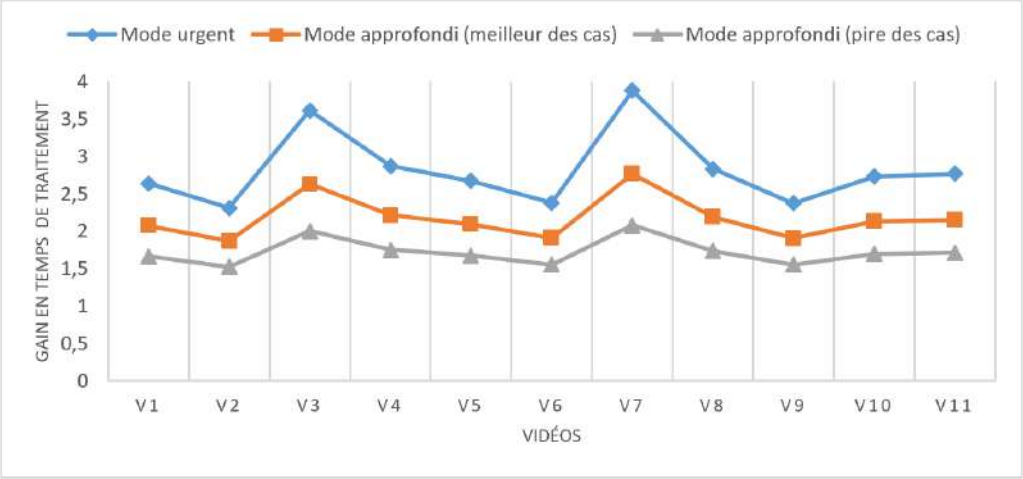


FIGURE 6.22 – Gain de temps pour le traitement détection détection et lecture automatique des plaques.

6.2.3 Expérience 2 - Enrichissement contextuel

Les tests effectués dans cette expérience visent à démontrer la faisabilité et l'apport de notre approche d'enrichissement contextuel dans le cadre de l'interrogation intelligente des vidéos. Les informations contextuelles disponibles et utilisées dans le processus d'enrichissement contextuel sont : les métadoonées sémantiques (présence et mouvement des objets dans les vidéos) et les métadonnées issues des open data (vitesse maximale de pluie). L'objectif est d'aider les opérateurs de vidéosurveillance en leur proposant les segments vidéos susceptibles de contenir les objets cibles (recherchés). Dans les scénarios définis, les objets cibles sont la voiture suspecte C, son conducteur D et le passager P.

6.2.3.1 Mise en place de l'expérience

Les vidéos utilisées dans cette expérience sont celles d'origine (sans aucune dégradation ajoutée). Dans le processus d'enrichissement contextuel, trois niveaux de requêtage sont implémentés (Figure 6.23) :

— Le premier niveau est basé sur les métadonnées décrivant la présence des objets (personnes et/ou véhicules) dans les vidéos. Il permet de filtrer les vidéo afin d'éviter à l'opérateur de visionner les séquences dans lesquels il n'y a pas d'objets.

— Le deuxième niveau est basé sur les métadonnées décrivant le mouvement des objets (personnes et/ou véhicules) dans les vidéos et permet d'éliminer les séquences vidéo dans lesquelles il n'y a pas de mouvement. Cela évite par exemple aux opérateur de visionner des images de parking dans lesquelles il n'y a pas de mouvement (car il y a presque toujours des véhicules dans les parkings, mais pas assez de mouvement et ces instants sans mouvements sont inutiles).

— Le troisième et dernier niveau de requêtage est basé sur les métadonnées liées à l'évènement pluie. Elle permet de classer les vidéos selon leur degré de visibilité lorsqu'elles ont été filmées dans des conditions météorologiques empêchant la visibilité.

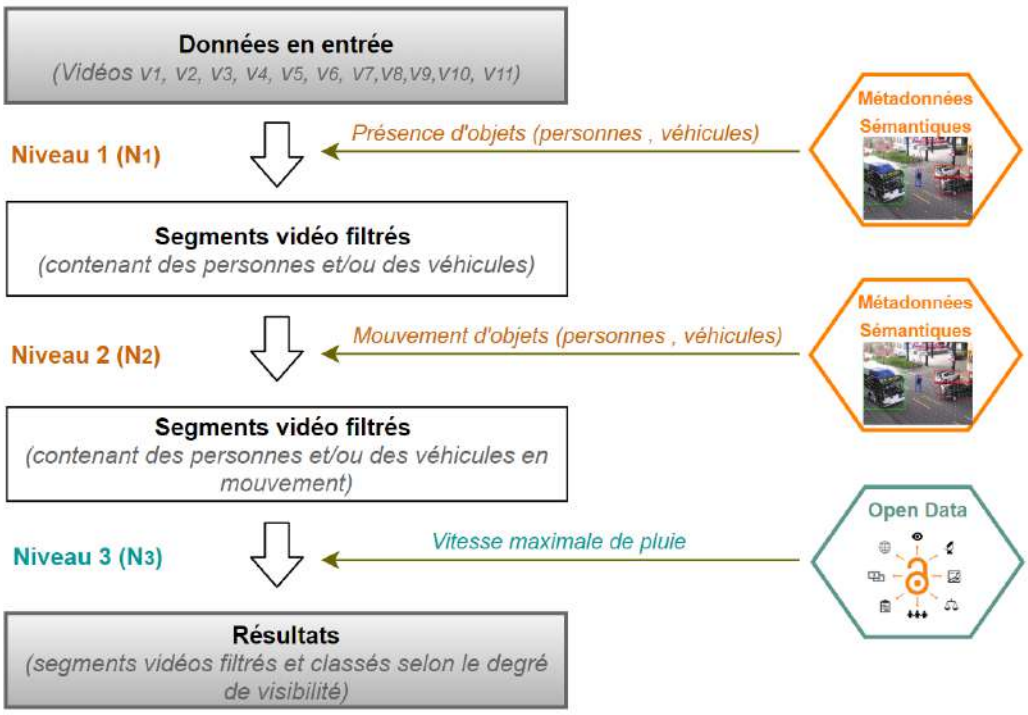


FIGURE 6.23 – Requêtage à 3 niveaux.

Les métadonnées décrivant la présence des objets (véhicules et personnes) dans les vidéos sont extraites des vidéos grâce aux algorithmes de détection d'objets YOLO.

Ensuite, les métadonnées décrivant le mouvement des objets sont obtenues en comparant les cordonnées des rectangles englobants (bounding box) des objets détectés dans les frames successives.

Les vidéos du dataset ayant été filmées dans de bonnes conditions météorologiques (pas de brouillard, pas de pollution, très faible pluie qui a durée moins d'une minute, etc.), aucun problème lié à la visibilité lors de l'acquisition ne se pose dans ce cas. par conséquent, le troisième niveau de requêtage prenant en compte les conditions météorologiques n'aura aucun impact sur l'enrichissement contextuel. Dans cette expérience, nous avons généré des données de météo synthétiques afin de tester notre approche dans de mauvaises conditions météorologiques. Les données synthétiques sont présentées à la Figure 6.24 et décrivent les informations suivantes : l'heure, l'intensité maximale de pluie en millimètre, la température en degré et la valeur en pourcentage de l'humidité.

ID	HEURE	INTENSITE_MAX_PLUIE	TEMPERATURE	HUMIDITE
1	17/07/2017 09:50:00	0	10,7	87
2	17/07/2017 09:51:00	0	11	89
3	17/07/2017 09:52:00	0,2	10,5	87
4	17/07/2017 09:53:00	0,2	10,3	90
5	17/07/2017 09:54:00	0,3	10,4	91
6	17/07/2017 09:55:00	0,4	10,6	90
7	17/07/2017 11:04:00	0,6	11,1	91
8	17/07/2017 11:05:00	0,6	11,2	91
9	17/07/2017 11:06:00	0,4	10,9	92
10	17/07/2017 11:07:00	0,2	10,7	91
11	17/07/2017 11:08:00	0,2	10,3	90
12	17/07/2017 11:09:00	0,2	10,1	88
13	17/07/2017 11:10:00	0	10,2	89

FIGURE 6.24 – Données synthétiques de météo.

6.2.3.2 Paramètres de l'expérience

Les paramètres indispensables dans le processus d'enrichissement contextuel sont : l'information spatiale, l'information temporelle, les fonction d'appartenance permettant de calculer les degrés d'imprécision dans la requête (s'il y a lieu).

Information spatiale - Elle détermine la zone concernée par la requête. Les informations contextuelles collectées et les caméras de surveillance prises en compte dans la requête sont celles appartenant à cette zone. Il s'agit dans cette expérience de "l'Université Paul Sabatier" (représentée par une surface).

Information temporelle - Elle représente les intervalles de temps à prendre en compte dans les vidéos générées par les caméras et pour les informations contextuelles de la requête. Dans cette expérience, il s'agit des intervalles de temps suivants : le "17 juillet 2017 de 9h50 à 9h55 (scénario 1) et de 11h04 à 11h10 (scénario 2)".

Fonction d'appartenance - Elle permet de calculer le degré d'imprécision dans la requête. Dans cette expérience, elle intervient au niveau 3 du requêtage qui consiste à évaluer la visibilité des vidéos en fonction des conditions météorologiques (cas de la pluie) observées lors de l'enregistrement des vidéos. Une mesure pouvant permettre de quantifier l'impact de la pluie sur la visibilité des vidéos est "la vitesse maximale de pluie". Après un croisement de notre observation (humaine) du phénomène pluie et des enregistrements générés par les capteurs de météo, nous déduisons une représentation de l'impact de la vitesse maximale de pluie sur la visibilité des images lors de leur acquisition par la fonction d'appartenance $f(x)$ de la Figure 6.25, qui calcul le degré d'imprécision pour une vitesse maximale de pluie x

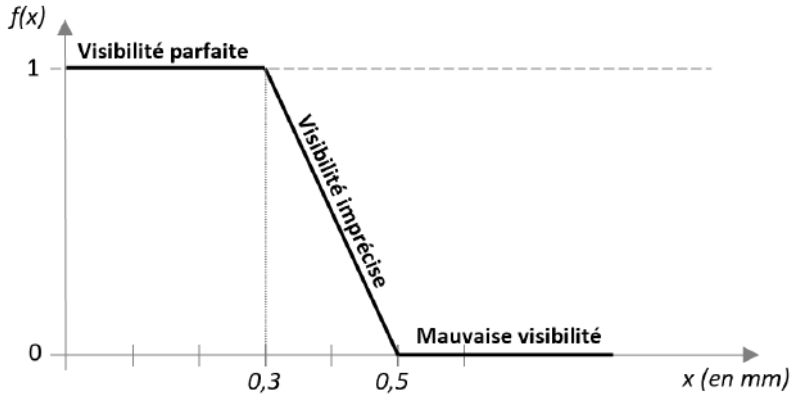


FIGURE 6.25 – Fonction d'appartenance pour la visibilité des vidéos.

Néanmoins, l'observation humaine du phénomène pluie n'est pas suffisante pour généraliser la représentation de l'imprécision de la visibilité des images dégradées par la pluie. Nous envisageons comme perspective de déterminer le degré d'imprécision de la visibilité des images grâce à un apprentissage basé sur le croisement des vidéos et des données générées par les capteurs de météo.

6.2.3.3 Résultats et interprétations

La requête JSON exécutée dans cette expérience est présentée à la Figure 6.26.
Premier niveau de requêtage - Il vise à sélectionner pour chaque vidéo (V1, V2, V3, V4, V5, V6, V7, V8, V9, V10 et V11) de l'expérience, les segments vidéo dans lesquels il y a des personnes et des véhicules. Ces segments vidéos sont les résultats du requêtage. Comme exemple, les résultats du premier niveau de requêtage pour la vidéo *V1 et V5* sont présentés au tableau 6.1.

Des exemples de frames sélectionnées et de frames éliminées à ce niveau de requêtage sont respectivement présentés aux Figure 6.27 et 6.28.

La Figure 6.29 présente le nombre frame à visionner dans les deux cas suivant :

```
{
"query":{
        "Spatio-temporal":[
            {
                "query_location":[
                    {
                    "zone_of_interest":"Université Paul Sabatier"
                    }
                ],
                "query_period":[
                    {"start":"2017-07-17 09:50:00", "end":"2017-07-17 09:55:00"},
                    {"start":"2017-07-17 11:04:00", "end":"2017-07-17 11:10:00"}
                ]
            }
        ],
        "contextual_information_source":
        [
            {
            "source":"métadonnées sémantiques",
            "parameter":"Présence des personnes et/ou véhicules",
            "membership function":[
                    {"title":"","range":""}
                ]
            },
            {
            "source":"métadonnées sémantiques",
            "parameter":"Mouvement des personnes et/ou véhicules",
            "membership function":[
                    { "title":"","range":""}
                ]
            },
            {
            "source":"Open data",
            "parameter":"Vitesse maximale de pluie",
            "membership function":[
                    {"title":"Visibilité parfaite","range":"[0, 0.3]"},
                    {"title":"Visibilité imprécise","range":"]0.3, 0.5["},
                    {"title":"Mauvaise visibilité","range":"[0.5, 1]"}
                ]
            }
        ]
    }
}
```


FIGURE 6.26 – Requête JSON pour l'enrichissement contextuel.

TABLE 6.1 – Exemple de résultats au premier niveau de requêtage.

Vidéos	**Libellés segments**	**Segments (frame_début -frame_fin)**	**Intervalles de temps (format hh :mm :ss)**
V1	S1	[f0 - f8640]	[09 :50 :00 - 09 :54 :48]
	S2	[f8641 - f18840]	[11 :04 :00 - 11 :09 :40]
V5	S1	[f1050 - f4980]	[09 :50 :35 - 09 :52 :46]
	S2	[f5580 - f5820]	[09 :53 :06 - 09 :53 :14]
	S3	[f 8040 - f8640]	[09 :54 :28 - 09 :54 :48]
	S4	[f 9150- f10380]	[11 :04 :17 - 11 :04 :58]
	S5	[f11190 - f12270]	[11 :05 :25 - 11 :06 :01]
	S6	[f12600 - f14550]	[11 :06 :12 - 11 :07 :17]
	S7	[f14640 - f16980]	[11 :07 :20 - 11 :08 :38]

— *Manuel* : correspond au nombre de frame à visionner sans requêtage (vidéo entière).

FIGURE 6.27 – Exemples de frames sélectionnées au premier niveau de requêtage.

FIGURE 6.28 – Exemples de frames éliminées au premier niveau de requêtage.

— *Requêtage basé sur la présence des objets* : correspond au nombre de frame à visionner après le premier niveau de requêtage.

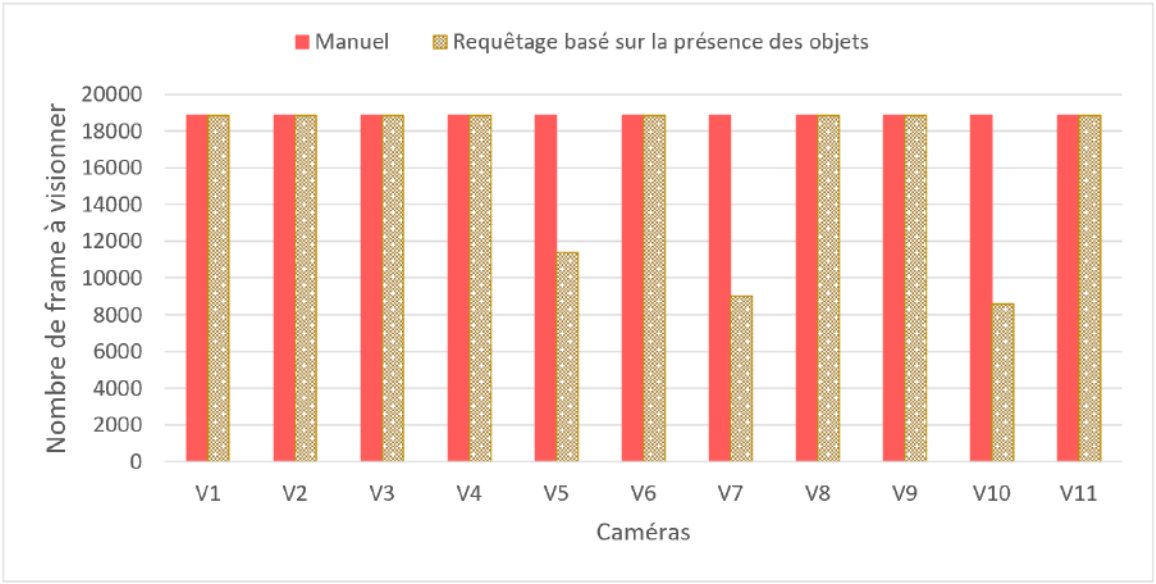


FIGURE 6.29 – Nombre de frame à visionner après le premier niveau de requâtage.

Sur la Figure 6.29, le nombre de frame à visionner pour chacune des vidéos *V5, V7 et V10* après le premier niveau de requêtage a été "considérablement" réduit par rapport au nombre de frame à visionner pour la vidéo entière. Par contre, pour chacune des vidéos *V1, V2, V3, V4, V6, V8, V9, et V11*, le nombre de frame à visionner après

le premier niveau de requêtage est égale au nombre de frame à visionner pour la vidéo entière. Cela est dû au fait que les champs de vue de ces caméras croisent des parkings dans lesquels il y a des véhicules stationnés (exemples de la Figure 6.30. Un tel cas de figure nécessite un requêtage basé sur d'autres métadonnées telles que celles décrivant le mouvement des objets dans la vidéo.

FIGURE 6.30 – Exemples de frames contenant de véhicules presque en permanence mais pas toujours de mouvement.

Deuxième niveau de requêtage - A ce niveau, les données en entrée sont les résultats du premier niveau de requêtage, c'est à dire, pour chaque vidéo, les segments dans lesquels il y a des personnes et/ou des véhicules. L'objectif ici est de sélectionner parmi ces segments vidéo ceux dans lesquels les objets sont en mouvement. Comme exemple, le résultat du deuxième niveau de requêtage pour la vidéo *V1* est présenté au tableau 6.2.

TABLE 6.2 – Exemple de résultats au deuxième niveau de requêtage.

Vidéos	**Libellés segments**	**Segments (frame_début - frame_fin)**	**Intervalles de temps (format hh :mm :ss)**
V1	S1	[f0 - f540]	[09 :50 :00 - 09 :50 :18]
	S2	[f600 - f1020]	[09 :50 :20 - 09 :50 :34]
	S3	[f1440 - f2550]	[09 :50 :48 - 09 :51 :25]
	S4	[f2790 - f2940]	[09 :51 :33 - 09 :51 :38]
	S5	[f3870 - f5430]	[09 :52 :09 - 09 :53 :01]
	S6	[f5850 - f6450]	[09 :53 :15 - 09 :53 :35]
	S7	[f7410 - f8520]	[09 :54 :07 - 09 :54 :44]
	S8	[f8820 - f9090]	[11 :04 :06 - 11 :04 :15]
	S9	[f 9240 - f9420]	[11 :04 :20 - 11 :04 :26]
	S10	[f9990 - f11040]	[11 :04 :45 - 11 :05 :20]
	S11	[f11730 - f13500]	[11 :05 :43 - 11 :06 :42]
	S12	[f13860 - f14550]	[11 :06 :54 - 11 :07 :17]
	S13	[f14850 - f15510]	[11 :07 :27 - 11 :07 :49]
	S14	[f16260 - f16470]	[11 :08 :14 - 11 :08 :21]
	S15	[f16590 - f16740]	[11 :08 :25 - 11 :08 :30]
	S16	[f16920 - f17040]	[11 :08 :36 - 11 :08 :40]
	S17	[f17550 - f18780]	[11 :08 :57 - 11 :09 :38]

Des exemples de frames sélectionnées sont présentés à la Figure 6.31. Le nombre de frame à visionner pour chaque vidéo après le deuxième niveau de filtrage est présenté à la Figure 6.34. Sur cette figure, on peut voir la différence du nombre de frame à visionner entre les deux niveaux de requêtage.

FIGURE 6.31 – Exemples de frames sélectionnées au deuxième niveau de requêtage.

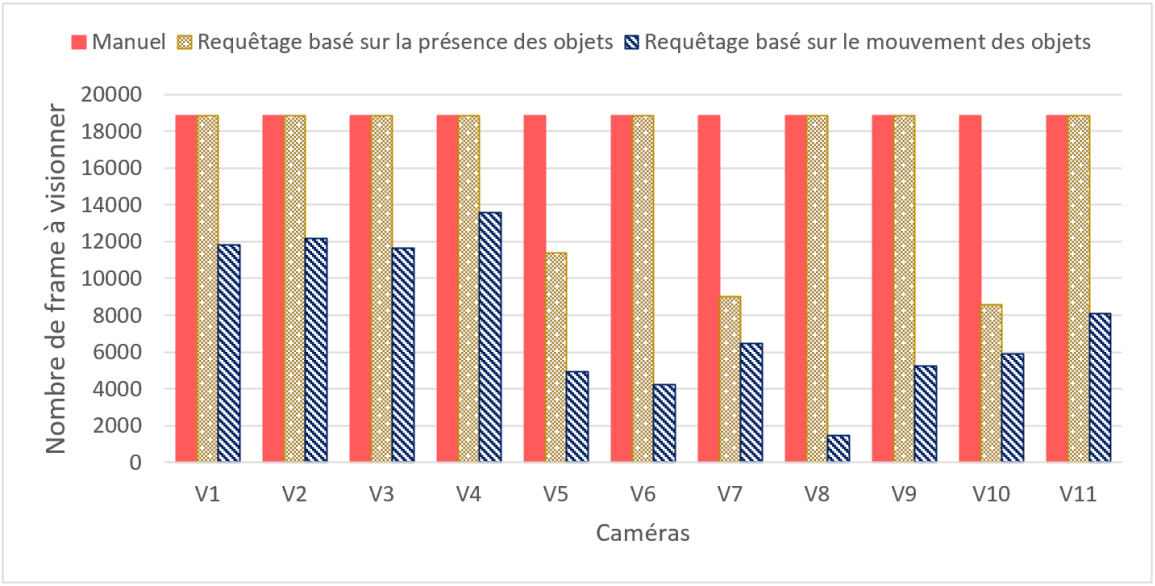


FIGURE 6.32 – Nombre de frame à visionner après le deuxième niveau de requâtage.

Troisième niveau de requêtage - L'objectif consiste à faire un "ranking" des segments vidéo (résultats du deuxième niveau de requêtage) en fonction de leur degré de visibilité défini grâce à l'évaluation des dégradations dues à la pluie lors de l'acquisition des vidéos.

La donnée considérée pour l'évaluation de l'influence de la pluie sur la visibilité des images étant l'intensité maximale de pluie, la première étape consiste à calculer le degré de visibilité associé à chaque intensité maximale de pluie enregistrée pendant l'acquisition des vidéos (données synthétiques de météo, Figure 6.24), en utilisant la fonction d'appartenance présentée dans les paramètres de l'expérience (Figure 6.25). Nous avons effectué une conversion temps classique - temps relatif des intensités maximales de pluie grâce à l'algorithme de conversion proposé dans notre approche (algo-

rithme 3). Le résultat de cette conversion est présenté à la Figure 6.33. Les degrés de visibilité associés à chaque intensité maximale de pluie de ce résultat sont présentés au tableau 6.3.

Id	Temps relatif	Intensité_max_pluie
T1	17/07/2017 [09:50:00 - 09:51:59]	0
T2	17/07/2017 [09:52:00 - 09:53:59]	0,2
T3	17/07/2017 [09:54:00 - 09:54:59]	0,3
T4	17/07/2017 [09:55:00 - 09:55:59]	0,4
T5	17/07/2017 [11:04:00 - 11:05:59]	0,6
T6	17/07/2017 [11:06:00 - 11:06:59]	0,4
T7	17/07/2017 [11:07:00 - 11:09:59]	0,2
T8	17/07/2017 [11:10:00 - 11:10:59]	0

FIGURE 6.33 – Conversion temps classique - temps relatif des intensités maximales de pluie.

TABLE 6.3 – Degrés de visibilité associés aux intensités maximales de pluie.

Id	**Temps relatif**	**Intensité maximale de pluie**	**Degré de visibilité(ρ)**
T1	17/07/2017 [09 :50 :00 - 09 :51 :59]	0	1
T2	17/07/2017 [09 :52 :00 - 09 :53 :59]	0,2	1
T3	17/07/2017 [09 :54 :00 - 09 :54 :59]	0,3	1
T4	17/07/2017 [09 :55 :00 - 09 :55 :59]	0,4	0,65
T5	17/07/2017 [11 :04 :00 - 11 :05 :59]	0,6	0
T6	17/07/2017 [11 :06 :00 - 11 :06 :59]	0,4	0,65
T7	17/07/2017 [11 :07 :00 - 11 :09 :59]	0,2	1
T8	17/07/2017 [11 :10 :00 - 11 :10 :59]	0	1

La deuxième étape consiste à évaluer le degré de visibilité des différents segments vidéo (résultats deuxième niveau de requêtage) en faisant un croisement temporel entre les intervalles de temps de chacun de ces segments vidéo et les intervalles de temps liés aux intensités maximales de pluie. Ce croisement temporel pour la vidéo *V1* permet d'avoir les degrés de visibilité pour chaque segment vidéo comme présentés dans le tableau 6.4.

La troisième étape consiste à faire un classement en fonction du degré de visibilité associé à chaque segment vidéo. Le tableau 6.5 présente le classement des segments de la vidéo *V1* en fonction de leur degré de visibilité. Ce tableau représente le résultat du troisième niveau de requêtage pour la vidéo *V1*. Les segments vidéo dont les degrés de visibilité sont égaux à 0 sont proposés à l'opérateur de vidéosurveillance en dernière position, il peut décider de les visionner ou pas.

TABLE 6.4 – Degré de visibilité de chaque segment de la vidéo *V1*.

Vidéos	Libellés segments	Intervalles de temps (format hh :mm :ss)	Degré de Visibilité(ρ)
V1	S1	[09 :50 :00 - 09 :50 :18]	1
	S2	[09 :50 :20 - 09 :50 :34]	1
	S3	[09 :50 :48 - 09 :51 :25]	1
	S4	[09 :51 :33 - 09 :51 :38]	1
	S5	[09 :52 :09 - 09 :53 :01]	1
	S6	[09 :53 :15 - 09 :53 :35]	1
	S7	[09 :54 :07 - 09 :54 :44]	1
	S8	[11 :04 :06 - 11 :04 :15]	0
	S9	[11 :04 :20 - 11 :04 :26]	0
	S10	[11 :04 :45 - 11 :05 :20]	0
	S11	[11 :05 :43 - 11 :06 :42]	0,65
	S12’	[11 :06 :43 - 11 :06 :59]	0,65
	S12”	[11 :07 :00 - 11 :07 :17]	1
	S13	[11 :07 :27 - 11 :07 :49]	1
	S14	[11 :08 :14 - 11 :08 :21]	1
	S15	[11 :08 :25 - 11 :08 :30]	1
	S16	[11 :08 :36 - 11 :08 :40]	1
	S17	[11 :08 :57 - 11 :09 :38]	1

TABLE 6.5 – Exemple de résultats au troisième niveau de requêtage.

Vidéos	Rang	Libellés segments	Intervalles de temps (format hh :mm :ss)	Degré de Visibilité(ρ)	Description
V1	1	S1	[09 :50 :00 - 09 :50 :18]	1	Visibilité parfaite
	2	S2	[09 :50 :20 - 09 :50 :34]	1	Visibilité parfaite
	3	S3	[09 :50 :48 - 09 :51 :25]	1	Visibilité parfaite
	4	S4	[09 :51 :33 - 09 :51 :38]	1	Visibilité parfaite
	5	S5	[09 :52 :09 - 09 :53 :01]	1	Visibilité parfaite
	6	S6	[09 :53 :15 - 09 :53 :35]	1	Visibilité parfaite
	7	S7	[09 :54 :07 - 09 :54 :44]	1	Visibilité parfaite
	8	S12”	[11 :07 :00 - 11 :07 :17]	1	Visibilité parfaite
	9	S13	[11 :07 :27 - 11 :07 :49]	1	Visibilité parfaite
	10	S14	[11 :08 :14 - 11 :08 :21]	1	Visibilité parfaite
	11	S15	[11 :08 :25 - 11 :08 :30]	1	Visibilité parfaite
	12	S16	[11 :08 :36 - 11 :08 :40]	1	Visibilité parfaite
	13	S17	[11 :08 :57 - 11 :09 :38]	1	Visibilité parfaite
	14	S11	[11 :05 :43 - 11 :06 :42]	0,65	Visibilité moyenne
	15	S12’	[11 :06 :43 - 11 :06 :59]	0,65	Visibilité moyenne
	16	S8	[11 :04 :06 - 11 :04 :15]	0	Visibilité nulle
	17	S9	[11 :04 :20 - 11 :04 :26]	0	Visibilité nulle
	18	S10	[11 :04 :45 - 11 :05 :20]	0	Visibilité nulle

6.2.3.4 Évaluation

Dans cette section, nous présentons une évaluation de l’approche d’enrichissement contextuel proposée. Les résultats obtenus grâce à cette approche sont évalués par rapport à la vérité terrain (*"Ground Truth"*). Le data set TOCADA offre une annotation

complète des vidéos, permettant ainsi d'élaborer la vérité terrain qui représente ici le nombre "effectif" de frames à visionner pour chaque vidéo, c'est à dire l'ensemble des frames dans lesquelles les suspects (cibles) apparaissent pour chaque vidéo.

La Figure 6.34 présente nombre de frames à visionner pour chaque vidéo dans les trois cas suivant :

— **Manuel** : nombre de frames à visionner pour la vidéo entière.

— **Enrichissement contextuel** : nombre de frame à visionner pour chaque vidéo après avoir appliqué le processus d'enrichissement contextuel. Il s'agit ici du nombre obtenu au deuxième niveau de requêtage, car au troisième niveau de requêtage, les segments vidéo sont classés en fonction de leur degré de visibilité et le nombre de frames à visionner ne varie pas par rapport au deuxième niveau de requêtage. Les segments vidéo dont les degrés de visibilité sont égaux à 0 peuvent être ignorés par l'opérateur de vidéosurveillance, mais ils sont proposés comme dans les résultats et classés en dernière position.

— **Vérité terrain** : nombre réel de frames à visionner dans lesquelles apparaissent les cibles (véhicule, conducteur, passager).

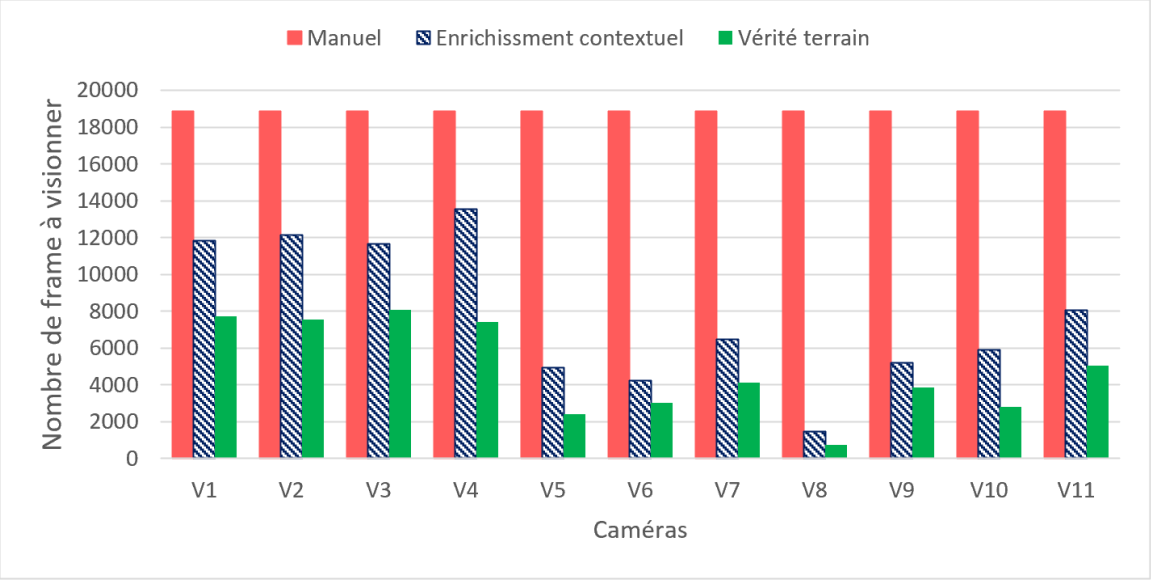


FIGURE 6.34 – Comparaison des résultats de l'enrichissement contextuel à la vérité terrain.

Sur cette figure (6.34), la comparaison montre d'une part que les nombres de frames à visionner pour toutes les vidéos après l'enrichissement contextuel sont très proches des nombres réels de frames à visionner (vérité terrain), sauf pour la vidéo *V4*. D'autre part, le nombre total de frame à visionner dans le cas "Manuel" est $NbF_M = 18840 * 11 = 207240$ $frames$ équivalent à **1 heure 55 minutes 08 secondes** de vidéo à visionner, et le nombre total de frame à visionner après l'enrichissement contextuel

est $NbF_{EC} = 85470\ frames$ équivalent à **47 minutes 29 secondes** de vidéo à visionner. Le rapport ($\delta = \frac{NbF_{EC}}{NbF_M} = 0.41$) montre que l'enrichissement contextuel a permis une réduction d'environ 60% du temps total de visionnage des vidéos entières.

Afin d'évaluer l'exactitude et l'exhaustivité des résultats de l'enrichissement contextuel, nous avons calculé la précision et le rappel. Nous notons P et R les mesures de la précision et du rappel liées à l'ensemble des frames récupérées par notre méthode. Dans notre contexte, $P < 1$ signifie que l'ensemble des résultats contient des frames non pertinentes, et $R < 1$ signifie que certaines frames pertinentes ont été ignorées. Nous calculons P_i et R_i pour la vidéo i comme suit :

$$P(i) = \frac{|f_A(i) \bigcap f_P(i)|}{|f_A(i)|}$$

$$R(i) = \frac{|f_A(i) \bigcap f_P(i)|}{|f_V(i)|}$$

où $f_A(i)$, $f_P(i)$ et $f_V(i)$ représentent respectivement l'ensemble de frames récupérées par notre approche, l'ensemble de frames pertinentes récupérées par notre approche, et l'ensemble de frames de la vérité terrain pour la vidéo i.

La précision globale P et le rappel global R sont calculés comme suit :

$$P = \frac{\sum_{i=0}^{n} P(i)}{n} = 0.60$$

$$R = \frac{\sum_{i=0}^{n} R(i)}{n} = 0.99$$

Les valeurs de P et R démontrent respectivement que 60% des frames récupérées par notre approche d'enrichissement contextuel sont pertinentes, et la quasi-totalité des frames pertinentes ont été récupérées.

6.3 Conclusion

Dans ce chapitre, nous avons présenté une description détaillée de l'architecture du framework développé pour le filtrage négatif et l'interrogation "intelligente" des vidéos issues des systèmes de vidéosurveillance. Ce framework s'appuie sur le modèle générique de métadonnées proposé et implémente les algorithmes et le mécanisme de requêtage proposés dans les chapitres précédents. Les expérimentations menées ont permis de démontrer la faisabilité et l'utilité de l'approche proposée dans notre contexte d'application. Les résultats des expériences pour l'ensemble des datasets utilisés sont prometteurs et peuvent améliorer grâce aux techniques d'apprentissage automatique et profond.

Chapitre 7

Conclusion générale et perspectives

Les travaux présentés dans ce manuscrit s'inscrivent dans le cadre général du filtrage négatif et l'interrogation "intelligente" des données grâce à la modélisation des métadonnées multi sources et hétérogènes. Ces travaux s'appliquent aux systèmes qui traitent de grands volumes de données. L'exemple étudié dans cette thèse est celui des systèmes de vidéosurveillance, et l'objectif consiste à assister les opérateurs de vidéosurveillance dans leur tâche en leur fournissant des outils permettant de réduire le grand volume de vidéo à visionner et implicitement le temps de recherche, tout en résolvant les problèmes d'interopérabilité liés aux systèmes de vidéosurveillance. Pour cela, les axes de recherche tels que l'intégration, la gestion et l'interrogation des données et métadonnées hétérogènes ont été étudiés. Leur implémentation dans le cadre des systèmes de vidéosurveillance a été présentée dans cette thèse. Ce chapitre présente le bilan de l'approche proposée et conclut sur les perspectives.

7.1 Bilan

Tout au long de ce document, nous avons développé une approche envisageable pour la résolution de la problématique énoncée à l'introduction de cette thèse. Cette approche a été présentée à la fois en termes de modélisation, de technique et d'implémentation. Afin d'atteindre les objectifs fixés, nous avons exploité des travaux basés sur différents domaines de recherche :

— **Gestion et intégration des données**. L'hétérogénéité des systèmes de vidéosurveillance due aux spécifications des capteurs/caméras (fabricants, formats de données et métadonnées, etc.) et aux différents contextes d'acquisition des contenus a été abordée sous un angle de modélisation des (méta)données. La modélisation des métadonnées spatiales et temporelles a été choisie pour l'intégration des différentes sources d'informations contextuelles.

— **Interrogation des données**. Dans les systèmes traitant de grands volumes de données, l'interrogation efficace et/ou intelligente des données est un atout majeur pour des utilisateurs qui souhaitent trouver des informations désirées parmi celles disponibles. L'application de la logique floue aux systèmes d'interrogation des données semble prometteuse. Dans cette thèse, le mécanisme de requêtage

ou d'interrogation développé est basé sur des notions de préférences floues visant à proposer aux utilisateurs des éléments de réponse en fonction de leur degré de satisfaction des conditions des requêtes.

— **Intelligence artificielle**. L'intelligence artificielle englobe un ensemble de technologies et concepts tels que l'apprentissage automatique (*machine learning*), les réseaux de neurones, et l'apprentissage profond (*deep learning*). Les technologies de deep learning telles que les algorithmes de détection YOLO, ont permis l'extraction des caractéristiques et des métadonnées vidéo utilisées dans l'approche proposée. Ces algorithmes ont aussi servi dans le cadre des expérimentations.

En s'appuyant sur ces différents domaines de recherche, nos contributions portent sur les trois points suivants : (i) la mise en œuvre du filtrage négatif des contenus de vidéosurveillance, (ii) le développement d'un processus d'enrichissement contextuel basé sur un raisonnement spatio-temporel, et (iii) la proposition d'une modélisation générique des métadonnées de vidéosurveillance dans le but d'enrichir le dictionnaire des métadonnées de la norme ISO 22311/ IEC 77.

Mise en œuvre du filtrage négatif. Le filtrage négatif proposé a permis dans le cadre de l'analyse a posteriori des vidéos, de gagner du temps de traitement en éliminant parmi les vidéos à analyser, les séquences inexploitables sur la base de la qualité et de l'utilisabilité/utilité des vidéos. Les algorithmes de filtrages définis s'appuient sur une modélisation générique des métadonnées de qualité et d'utilisabilité/utilité des vidéos.

Enrichissement contextuel via les métadonnées spatiales et temporelles. L'enrichissement contextuel proposé est un processus comportant trois étapes itératives (analyse des informations contextuelles, modélisation des informations contextuelles et développement du mécanisme de requêtage) et permettant d'intégrer diverses sources d'informations contextuelles. Les notions de préférences floues sont incluses dans le mécanisme de requêtage afin de prendre en compte les préférences de l'utilisateur.

Généralisation de la modélisation des métadonnées de vidéosurveillance. Un modèle générique de métadonnées de vidéosurveillance intégrant les métadonnées décrivant le mouvement et le champ des caméras, les métadonnées issues des algorithmes d'analyse du contenu, et les métadonnées issues des informations contextuelles, a été proposé pour compléter le dictionnaire des métadonnées de la norme ISO 22311/ IEC 79 afin de mettre en évidence en plus des besoins d'interopérabilité des systèmes de vidéosurveillance, les besoins de recherche et de filtrage des contenus vidéo.

Les expérimentations menées à partir du framework développé ont permis de démontrer la faisabilité de notre approche dans un cas réel et de valider nos propositions. Les métadonnées utilisées dans ces expérimentations sont extraites des vidéos pour les unes (métadonnées de qualité vidéo et métadonnées sémantiques) et sont des données

synthétiques pour les autres (informations sur la météo).

7.2 Perspectives

Nos travaux ouvrent des voies de recherche pour l'amélioration de l'approche proposée. A court terme, nous envisageons dans le cadre du projet FILTER2 d'intégrer les métadonnées de qualité et d'utilisabilité/utilité provenant des autres sous-projets. Le modèle de donnée proposé à cet effet dans ce travail est générique et extensible. Des tests dans un cadre opérationnel sont prévus avec la Police Technique et Scientifique afin d'évaluer le l'approche et le framework proposés dans des cas réels d'enquêtes policières. Dans le mécanisme de filtrage négatif proposé, la définition des seuils de compatibilité des traitements aux différents modes d'analyse n'est pas "définitive". Nous envisageons d'utiliser des méthodes d'apprentissage automatique sur un ensemble considérable de vidéos afin de définir des seuils de compatibilité qui soient représentatifs.

Dans les expériences sur le processus d'enrichissement contextuel, l'évaluation des dégradations des vidéos liées aux phénomènes environnementaux (pluie, brouillard, etc.) est basée sur des observations humaines, ce qui peut varier d'un humain à un autre. En effet, si une "forte" pluie ou un "fort" brouillard était présent à un instant d'intérêt lors de l'acquisition des images par une caméra, il est alors plus probable que la qualité du segment vidéo associé en soit dégradé. Pour exploiter ce type d'information contextuelle dans l'évaluation de la dégradation de segment vidéo, nous envisageons de proposer un modèle d'apprentissage profond permettant d'estimer la dégradation a priori d'un segment de vidéo à partir des informations météorologiques à l'instant de capture. À titre prospectif, la mise en place d'un tel modèle pourrait être conduit comme suit :

1. Accumulation de mesures météorologiques et des segments vidéos correspondant ; à titre d'exemple, nous possédons sur le site de l'UPS une station météorologique fournissant des mesures précises (https ://www.pente-ups.fr/meteo_data/about/), et la possibilité d'acquérir des segments vidéo dans d'un système de video-surveillance contrôlé (ex : corpus TOCADA https ://zenodo.org/record/1219421).
2. Évaluation de la qualité des segments vidéo en fonction d'une tâche données (ex : reconnaissance de plaques d'immatriculation, détection de véhicule, etc.).
3. Conception d'un modèle profond de type DNN (Deep Nerual Network) et ou RNN (Récurrent Neural network) permettant de prendre en entrée les données météorologiques et produire en sortie la qualité du ou des segments vidéo correspondants.
4. Apprentissage du modèle et évaluation de sa capacité prédictive sur un dataset

de test (non-utilisé durant l'apprentissage).

A moyen ou long terme, nous envisageons de mettre en œuvre les opérations d'agrégation et d'inférence contextuelle définies dans cette thèse et qui s'appliqueraient très bien à une source d'informations contextuelles (très prometteuse dans le contexte de la vidéosurveillance) qu'est les médias sociaux et dont une modélisation a été proposée dans cette thèse.

Les travaux présentés ouvrent les portes à d'autres applications, notamment dans le cadre de la réduction du volume de données et des applications sensibles aux contexte.

Annexe A

Bases de données spatiales

Le termes information spatiale, information géographique, données géo-spatiales ou géo-information, regroupent des données sur la localisation, la forme et les relations entre espaces géographiques. Une base de données spatiales est une base de données optimisée pour stocker et requêter des données reliées à des objets référencés géographiquement, y compris des points, les lignes et des polygones. Alors que les bases de données classiques peuvent comprendre différents types de données numériques et caractères, des fonctions additionnelles ont besoin d'être ajoutées pour traiter les types de données spatiales. Celles-ci sont typiquement appelées géométrie ou caractère. Les bases de données spatiales fournissent des opérateurs, des fonctions spéciales et des index pour interroger et manipuler les données en utilisant le langage SQL par exemple. Les bases de données spatiales ont plusieurs rôles : stockage de données spatiales, indexation spatiale, requêtes spatiales.

Il existe plusieurs Systèmes de Gestion de Base de Données (SGBD) spatiales lesquels les plus utilisés sont :

— DB2 Spatial extender

— Microsoft SQL Server (depuis la version 2008)

— MyGIS (extension intégrée de MySQL depuis la v.4.1)

— Oracle spatial

— PostGIS (extension intégrée de PostgreSQL)

Une comparaison de ces SGBD Spatiales est proposée dans le tableau A.1 a permis de retenir **Oracle Spatial** dans cette thèse.

Table A.1 – Comparaison des SGBD spatiales.

SGBD spatial	Index spatial	Support Raster	Référencement linéaire	Modèle topologique	Support 3D
DB2-Spatial extender	R-Tree	Non	Oui	Non	Peu
SQL Server-Spatial	Grid	Non	Non	Non	Peu
MySQL-MyGIS	R-Tree	Non	Non	Non	Non
Oracle-Spatial	**R-Tree**	**Oui**	**Oui**	**Oui**	**Oui**
PostgreSQL-PostGIS	R-Tree	Non	Non	Oui	Non

Bibliographie

[Abowd and Mynatt, 2000] Abowd, G. D. and Mynatt, E. D. (2000). Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1) :29–58.

[Adomavicius and Ricci, 2009] Adomavicius, G. and Ricci, F. (2009). Recsys' 09 workshop 3 : workshop on context-aware recommender systems (cars-2009). In *Proceedings of the third ACM conference on Recommender systems*, pages 423–424. ACM.

[Adomavicius et al., 2005] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, 23(1) :103–145.

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, 17(6) :734–749.

[Agrawal et al., 2006] Agrawal, R., Rantzau, R., and Terzi, E. (2006). Context-sensitive ranking. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 383–394. ACM.

[Ahmed et al., 2010] Ahmed, T., Ahmed, S., Ahmed, S., and Motiwala, M. (2010). Real-time intruder detection in surveillance networks using adaptive kernel methods. In *2010 IEEE International Conference on Communications*, pages 1–5. IEEE.

[Akrivas et al., 2002] Akrivas, G., Wallace, M., Andreou, G., Stamou, G., and Kollias, S. (2002). Context-sensitive semantic query expansion. In *Proceedings 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS 2002)*, pages 109–114. IEEE.

[Albano et al., 2015] Albano, A., Guillaume, J.-L., Heymann, S., and Le Grand, B. (2015). Studying graph dynamics through intrinsic time based diffusion analysis. In *Applications of Social Media and Social Network Analysis*, pages 103–124. Springer.

[Amer and Regazzoni, 2005] Amer, A. and Regazzoni, C. (2005). Introduction to the special issue on video object processing for surveillance applications. *Real-Time Imaging*, 11(3) :167–171.

[An and Kim, 2012] An, T.-K. and Kim, M.-H. (2012). Context-aware video surveillance system. *Journal of Electrical Engineering and Technology*, 7(1) :115–123.

[Au and Chan, 2005] Au, W.-H. and Chan, K. C. (2005). Mining changes in association rules : a fuzzy approach. *Fuzzy sets and systems*, 149(1) :87–104.

[Bai et al., 2013] Bai, L., Yan, L., and Ma, Z. (2013). Determining topological relationship of fuzzy spatiotemporal data integrated with xml twig pattern. *Applied intelligence*, 39(1) :75–100.

[Bauer et al., 2003] Bauer, J., Kutsche, R.-D., and Ehrmanntraut, R. (2003). Identification and modeling of contexts for different information scenarios in air traffic. *Technische Universität Berlin, Diplomarbeit*, pages 1–114.

[Beniwal and Arora, 2012] Beniwal, S. and Arora, J. (2012). Classification and feature selection techniques in data mining. *International Journal of Engineering Research & Technology*, 1(6) :1–6.

[Bezdek et al., 2012] Bezdek, J. C., Dubois, D., and Prade, H. (2012). *Fuzzy sets in approximate reasoning and information systems*, volume 5. Springer Science & Business Media.

[Carincotte et al., 2006] Carincotte, C., Desurmont, X., Ravera, B., Brémond, F., Orwell, J., Velastin, S., Odobez, J.-M., Corbucci, B., Palo, J., and Cernocky, J. (2006). Toward generic intelligent knowledge extraction from video and audio : the eu-funded caretaker project. *IET Conference on Crime and Security*, pages 470–475.

[Chang et al., 2007] Chang, Y., Lee, D.-J., Hong, Y., and Archibald, J. (2007). Unsupervised video shot detection using clustering ensemble with a color global scale-invariant feature transform descriptor. volume 2008, pages 1–10. Springer.

[Charrier, 2011] Charrier, C. (2011). *Modélisation statistique et classification par apprentissage pour la qualité des images*. PhD thesis.

[Charrier et al., 2015] Charrier, C., Saadane, A., and Fernandez-Maloigne, C. (2015). Comparison of no-reference image quality assessment machine learning-based algorithms on compressed images. In *Image Quality and System Performance XII*, volume 9396, pages 1–9. International Society for Optics and Photonics.

[Chen and Kotz, 2000] Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. *Dartmouth Computer Science Technical Report TR2000-381*, pages 1–16.

[Chen et al., 2013] Chen, Y.-L., Chen, T.-S., Huang, T.-W., Yin, L.-C., Wang, S.-Y., and Chiueh, T.-c. (2013). Intelligent urban video surveillance system for automatic vehicle detection and tracking in clouds. In *2013 IEEE 27th international conference on advanced information networking and applications (AINA)*, pages 814–821. IEEE.

[Codreanu, 2015] Codreanu, D. (2015). *Modélisation des métadonnées spatio-temporelles associées aux contenus vidéos et interrogation de ces métadonnées à*

partir des trajectoires hybrides : Application dans le contexte de la vidéosurveillance. PhD thesis, Université Paul Sabatier.

[Dadgostar et al., 2011] Dadgostar, F., Bigdeli, A., and Smith, T. (2011). An automated face enrolment and recognition system across multiple cameras on cctv networks. In *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–2. IEEE.

[de Caluwe et al., 2013] de Caluwe, R., De Tré, G., and Bordogna, G. (2013). *Spatio-temporal databases : flexible querying and reasoning*. Springer Science & Business Media.

[Deng et al., 2010] Deng, H., Lee, M. W., Hakeem, A., Javed, O., Yin, W., Yu, L., Scanlon, A., Rasheed, Z., and Haering, N. (2010). Fast forensic video event retrieval using geospatial computing. In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*, pages 1–8. ACM.

[Dey et al., 2001] Dey, A. K., Abowd, G. D., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction*, 16(2-4) :97–166.

[Dubois and Prade, 1997] Dubois, D. and Prade, H. (1997). Using fuzzy sets in flexible querying : Why and how ? In *Flexible query answering systems*, pages 45–60. Springer.

[Eckmann et al., 2004] Eckmann, J.-P., Moses, E., and Sergi, D. (2004). Entropy of dialogues creates coherent structures in e-mail traffic. *Proceedings of the National Academy of Sciences*, 101(40) :14333–14337.

[Edelman and Bijhold, 2010] Edelman, G. and Bijhold, J. (2010). Tracking people and cars using 3d modeling and cctv. *Forensic science international*, 202(1-3) :26–35.

[Fletcher, 2011] Fletcher, P. (2011). Is cctv effective in reducing anti-social behaviour. *Internet Journal of Criminology. UK : Lancaster University, Unpublished dissertation*.

[Fragkiadaki et al., 2012] Fragkiadaki, K., Zhang, G., and Shi, J. (2012). Video segmentation by tracing discontinuities in a trajectory embedding. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1853. IEEE.

[Franklin and Flaschbart, 1998] Franklin, D. and Flaschbart, J. (1998). All gadget and no representation makes jack a dull environment. In *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 155–160.

[Gauvin et al., 2013] Gauvin, L., Panisson, A., Cattuto, C., and Barrat, A. (2013). Activity clocks : spreading dynamics on temporal networks of human contact. volume 3, pages 1–6. Nature Publishing Group.

[Gerónimo and Kjellström, 2014] Gerónimo, D. and Kjellström, H. (2014). Unsupervised surveillance video retrieval based on human action and appearance. In *2014 22nd International Conference on Pattern Recognition*, pages 4630–4635. IEEE.

[Gill and Spriggs, 2005] Gill, M. and Spriggs, A. (2005). *Assessing the impact of CCTV*, volume 292. Home Office Research, Development and Statistics Directorate London.

[Gupta and Jain, 1997] Gupta, A. and Jain, R. (1997). Visual information retrieval. *Communications of the ACM*, 40(5) :70–79.

[Han et al., 2011a] Han, J., Haihong, E., Le, G., and Du, J. (2011a). Survey on nosql database. In *2011 6th international conference on pervasive computing and applications*, pages 363–366. IEEE.

[Han et al., 2011b] Han, J., Pei, J., and Kamber, M. (2011b). *Data mining : concepts and techniques*. Elsevier.

[Henricksen et al., 2003] Henricksen, K., Indulska, J., and Rakotonirainy, A. (2003). Generating context management infrastructure from high-level context models. In *In 4th International Conference on Mobile Data Management (MDM)-Industrial Track*, pages 1–6. Citeseer.

[Heymann and Le Grand, 2013] Heymann, S. and Le Grand, B. (2013). Monitoring user-system interactions through graph-based intrinsic dynamics analysis. In *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–10. IEEE.

[Hu et al., 2008] Hu, P., Indulska, J., and Robinson, R. (2008). An autonomic context management system for pervasive computing. In *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 213–223. IEEE.

[Hu et al., 2011] Hu, W., Xie, N., Li, L., Zeng, X., and Maybank, S. (2011). A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6) :797–819.

[Huang et al., 2012] Huang, D.-A., Kang, L.-W., Yang, M.-C., Lin, C.-W., and Wang, Y.-C. F. (2012). Context-aware single image rain removal. In *2012 IEEE International Conference on Multimedia and Expo*, pages 164–169. IEEE.

[Ichikawa and Hirakawa, 1986] Ichikawa, T. and Hirakawa, M. (1986). Ares : a relational database with the capability of performing flexible interpretation of queries. *IEEE Transactions on Software Engineering*, 12(5) :624–634.

[Jones, 2005] Jones, G. J. (2005). Challenges and opportunities of context-aware information access. In *International Workshop on Ubiquitous Data Management*, pages 53–60. IEEE.

[Kitchin, 2014] Kitchin, R. (2014). The real-time city ? big data and smart urbanism. *GeoJournal*, 79(1) :1–14.

[Klontz and Jain, 2013] Klontz, J. C. and Jain, A. K. (2013). A case study of automated face recognition : The boston marathon bombings suspects. *Computer*, 46(11) :91–94.

[Klyne et al., 2004] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M. H., and Tran, L. (2004). Composite capability/preference profiles (cc/pp) : Structure and vocabularies 1.0. Available at https://www.w3.org/TR/CCPP-struct-vocab/ (15 January 2004).

[Knappmeyer et al., 2010] Knappmeyer, M., Kiani, S. L., Frà, C., Moltchanov, B., and Baker, N. (2010). Contextml : A light-weight context representation and context management schema. In *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, pages 367–372. IEEE.

[Lawrence, 2000] Lawrence, S. (2000). Context in web search. *IEEE Data Eng. Bull.*, 23(3) :25–32.

[Lee and Pagliaro, 2013] Lee, H. and Pagliaro, E. (2013). Forensic evidence and crime scene investigation. *Journal of Forensic Investigation*, 1(1) :1–5.

[Lee et al., 2013] Lee, J.-V., Chuah, Y.-D., and Chai, C.-T. (2013). A multilevel home security system (mhss). *International Journal of Smart Home*, 7(2) :49–60.

[Lim et al., 2014] Lim, M. K., Tang, S., and Chan, C. S. (2014). isurveillance : Intelligent framework for multiple events detection in surveillance videos. *Expert Systems with Applications*, 41(10) :4704–4715.

[Ling et al., 2008] Ling, X., Chao, L., Huan, L., and Zhang, X. (2008). A general method for shot boundary detection. In *2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, pages 394–397. IEEE.

[Linoff and Berry, 2011] Linoff, G. S. and Berry, M. J. (2011). *Data mining techniques : for marketing, sales, and customer relationship management*. John Wiley & Sons.

[Ma, 2005] Ma, Z. (2005). Fuzzy information modeling with the uml. In *Advances in fuzzy object-oriented databases : Modeling and applications*, pages 153–176. IGI Global.

[Maamar et al., 2006] Maamar, Z., Benslimane, D., and Narendra, N. C. (2006). What can context do for web services ? *Communications of the ACM*, 49(12) :98–103.

[Malon et al., 2018] Malon, T., Roman-Jimenez, G., Guyot, P., Chambon, S., Charvillat, V., Crouzil, A., Péninou, A., Pinquier, J., Sèdes, F., and Sénac, C. (2018). Toulouse campus surveillance dataset : scenarios, soundtracks, synchronized videos

with overlapping and disjoint views. In *Proceedings of the 9th ACM Multimedia Systems Conference*, pages 393–398.

[Mangalampalli and Pudi, 2009] Mangalampalli, A. and Pudi, V. (2009). Fuzzy association rule mining algorithm for fast and efficient performance on very large datasets. In *2009 IEEE International Conference on Fuzzy Systems*, pages 1163–1168. IEEE.

[Mansoor et al., 2007] Mansoor, W., Khedr, M., Benslimane, D., Maamar, Z., Hauswirth, M., Aberer, K., Stefanidis, K., Pitoura, E., and Vassiliadis, P. (2007). A context-aware preference database system. page 439–600. Emerald Group Publishing Limited.

[Mathur and Bundele, 2016] Mathur, G. and Bundele, M. (2016). Research on intelligent video surveillance techniques for suspicious activity detection critical review. In *2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pages 1–8. IEEE.

[Mittal et al., 2011] Mittal, A., Moorthy, A. K., and Bovik, A. C. (2011). Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 723–727. IEEE.

[Mokbel and Levandoski, 2009] Mokbel, M. F. and Levandoski, J. J. (2009). Toward context and preference-aware location-based services. In *Proceedings of the eighth ACM international workshop on data engineering for wireless and mobile access*, pages 25–32. ACM.

[Muyeba et al., 2008] Muyeba, M., Khan, M. S., and Coenen, F. (2008). Fuzzy weighted association rule mining with weighted support and confidence framework. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 49–61. Springer.

[Nam et al., 2012] Nam, Y., Rho, S., and Park, J. H. (2012). Intelligent video surveillance system : 3-tier context-aware surveillance system with metadata. *Multimedia Tools and Applications*, 57(2) :315–334.

[Nasreen and Shobha, 2013] Nasreen, A. and Shobha, G. (2013). Key frame extraction from videos-a survey. volume 3, pages 194–198. Technopark Publications.

[Niebles et al., 2008] Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3) :299–318.

[Palmisano et al., 2008] Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE transactions on knowledge and data engineering*, 20(11) :1535–1549.

[Panniello and Gorgoglione, 2012] Panniello, U. and Gorgoglione, M. (2012). Incorporating context into recommender systems : an empirical comparison of context-based approaches. *Electronic Commerce Research*, 12(1) :1–30.

[Panta et al., 2018] Panta, F. J., Roman-Jimenez, G., and Sèdes, F. (2018). Modeling metadata of cctv systems and indoor location sensors for automatic filtering of relevant video content. In *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–9. IEEE.

[Panta and Sèdes, 2016a] Panta, F. J. and Sèdes, F. (2016a). Mobile objects in indoor environment : Trajectories reconstruction. In *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media*, pages 332–336. ACM.

[Panta and Sèdes, 2016b] Panta, F. J. and Sèdes, F. (2016b). Querying indoor spatio-temporal data by hybrid trajectories. In *Proceedings of the Eighth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 11–18. ACM.

[Pascoe, 1998] Pascoe, J. (1998). Adding generic contextual capabilities to wearable computers. In *2nd international symposium on wearable computers*, pages 92–99. Ieee Computer Soc.

[Perera et al., 2013] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2013). Context aware computing for the internet of things : A survey. *IEEE communications surveys & tutorials*, 16(1) :414–454.

[Ponomarenko et al., 2008] Ponomarenko, N., Lukin, V., Egiazarian, K., Astola, J., Carli, M., and Battisti, F. (2008). Color image database for evaluation of image quality metrics. In *2008 IEEE 10th workshop on multimedia signal processing*, pages 403–408. IEEE.

[Rodden et al., 1998] Rodden, T., Cheverst, K., Davies, K., and Dix, A. (1998). Exploiting context in hci design for mobile systems. In *Workshop on human computer interaction with mobile devices*, pages 21–22. Citeseer.

[Saad et al., 2010] Saad, M. A., Bovik, A. C., and Charrier, C. (2010). A dct statistics-based blind image quality index. *IEEE Signal Processing Letters*, 17(6) :583–586.

[Sachs, 1976] Sachs, W. M. (1976). An approach to associative retrieval through the theory of fuzzy sets. *journal of the American Society for Information Science*, 27(2) :85–87.

[Schilit et al., 1994] Schilit, B., Adams, N., and Want, R. (1994). Context-aware computing applications. In *1994 First Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE.

[Schiller and Voisard, 2004] Schiller, J. and Voisard, A. (2004). *Location-based services*. Elsevier.

[Sèdes and Panta, 2017] Sèdes, F. and Panta, F. J. (2017). (meta-) data modelling : gathering spatio-temporal data for indoor-outdoor queries. *SIGSPATIAL Special*, 9(1) :35–42.

[Semertzidis et al., 2010] Semertzidis, T., Dimitropoulos, K., Koutsia, A., and Grammalidis, N. (2010). Video sensor network for real-time traffic monitoring and surveillance. *IET intelligent transport systems*, 4(2) :103–112.

[Strang and Linnhoff-Popien, 2004] Strang, T. and Linnhoff-Popien, C. (2004). A context modeling survey. In *Workshop Proceedings*, pages 1–8.

[Sèdes et al., 2012] Sèdes, F., Marraud, D., Jean-François, S., Mulat, C., and Cépas, B. (2012). A posteriori analysis for investigative purposes. *Intelligent Video Surveillance Systems*, pages 33–46.

[Tahani, 1976] Tahani, V. (1976). A fuzzy model of document retrieval systems. *Information Processing & Management*, 12(3) :177–187.

[van den Eeden et al., 2016] van den Eeden, C. A., de Poot, C. J., and Van Koppen, P. J. (2016). Forensic expectations : Investigating a crime scene with prior information. *Science & justice*, 56(6) :475–481.

[Vert et al., 2002] Vert, G., Stock, M., and Morris, A. (2002). Extending erd modeling notation to fuzzy management of gis data files. *Data & Knowledge Engineering*, 40(2) :163–179.

[Wang et al., 2011] Wang, Y.-K., Fan, C.-T., Cheng, K.-Y., and Deng, P. S. (2011). Real-time camera anomaly detection for real-world video surveillance. In *2011 International Conference on Machine Learning and Cybernetics*, volume 4, pages 1520–1525. IEEE.

[Ward et al., 1997] Ward, A., Jones, A., and Hopper, A. (1997). A new location technique for the active office. *IEEE Personal communications*, 4(5) :42–47.

[Welsh and Farrington, 2008] Welsh, B. C. and Farrington, D. P. (2008). Effects of closed circuit television surveillance on crime. *Campbell systematic reviews*, 4(1) :1–73.

[Wu et al., 2008] Wu, X., Yuen, P. C., Liu, C., and Huang, J. (2008). Shot boundary detection : an information saliency approach. In *2008 Congress on Image and Signal Processing*, volume 2, pages 808–812. IEEE.

[Wygralak, 2013] Wygralak, M. (2013). Intelligent counting under information imprecision. volume 292. Springer.

[Xia et al., 2007] Xia, D., Deng, X., and Zeng, Q. (2007). Shot boundary detection based on difference sequences of mutual information. In *Fourth International Conference on Image and Graphics (ICIG 2007)*, pages 389–394. IEEE.

[Yanwei et al., 2011] Yanwei, S., Guangzhou, Z., and Haitao, P. (2011). Research on the context model of intelligent interaction system in the internet of things. In *2011 IEEE International Symposium on IT in Medicine and Education*, volume 2, pages 379–382. IEEE.

[Yogameena and Priya, 2015] Yogameena, B. and Priya, K. S. (2015). Synoptic video based human crowd behavior analysis for forensic video surveillance. In *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE.

[Yujie and Licai, 2010] Yujie, Z. and Licai, W. (2010). Some challenges for context-aware recommender systems. In *2010 5th International Conference on Computer Science & Education*, pages 362–365. IEEE.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3) :338–353.

[Zadrozny et al., 2009] Zadrozny, S., De Tré, G., De Caluwe, R., and Kacprzyk, J. (2009). An overview of fuzzy approaches to flexible database querying. In *Database technologies : concepts, methodologies, tools, and applications*, pages 135–156. IGI Global.