



Université  
de Toulouse

# 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 *21/05/2015* par :

**Dana CODREANU**

**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**

---

---

### JURY

FRANÇOIS BREMOND

VINCENT ORIA

JEAN-FRANÇOIS SULZER

ANDRÉ PENINO

Président

Examineur

Invité

Invité

---

**École doctorale et spécialité :**

*MITT : Image, Information, Hypermedia*

**Unité de Recherche :**

*Institut de Recherche en Informatique de Toulouse UMR 5505*

**Directeur de Thèse :**

*Florence SEDES*

**Rapporteurs :**

*Thierry DELOT et Sergio ILARRI*

---

## Remerciements

Je tiens à remercier Madame Josiane Mothe, professeur à l'université Toulouse 3 et responsable de l'équipe « Système d'Informations Généralisés » (SIG) de l'IRIT, pour m'avoir accueillie au sein de son équipe.

Je remercie ma directrice de thèse Madame Florence Sedes, professeur à l'université Toulouse 3, pour avoir encadré et dirigé mes recherches et pour m'avoir accordé non seulement ses conseils scientifiques précieux mais aussi son amitié et son support continu. Je le remercie pour la confiance qu'il m'a accordée, pour sa disponibilité, pour ses commentaires et ses remarques pertinentes. L'aboutissement de ce travail doit beaucoup à son recul, sa rigueur scientifique et son expertise. Je tiens ici à lui exprimer ma gratitude et mon profond respect et je l'assure que je ne l'oublierai jamais.

J'adresse mes sincères remerciements à Monsieur André Peninou, maître de conférences à l'université Toulouse Jean Jaures. Ses compétences scientifique et pédagogique ont fortement contribué à l'accomplissement de ce travail. Je le remercie pour sa disponibilité, ses précieux conseils, son soutien continu et ses relectures de mes travaux. Qu'il trouve ici ma profonde reconnaissance et mon grand respect.

Je tiens à remercier Monsieur Sergio Ilarri, maître de conférences à l'université de Zaragoza et à Monsieur Thierry Delot, professeur HDR à l'université de Valenciennes qui m'ont fait l'honneur d'être rapporteurs de ce mémoire et de faire partie du jury. Je remercie également à Monsieur François Bremond, professeur à l'INRIA Sophia Antipolis, à Monsieur Vincent Oria, maître de conférence à l'institut de Technologie du New Jersey, Newark, États-Unis et à monsieur Jean-François Sulzer de Thales d'avoir accepté d'examiner ce travail et de faire partie du jury. Je les remercie pour leur évaluation scientifique et leur travail de synthèse.

Je tiens également à remercier Monsieur Asanobu Kitamoto et Vincent Oria, professeurs à l'Institut National d'Informatique (NII) de Tokyo et à l'Institut de Technologie du New Jersey, Newark, États-Unis de m'avoir offert l'opportunité de faire partie de leurs équipes pour une période de trois mois. Leurs bons conseils, leur compétence et leur expérience m'ont apporté un grand enrichissement professionnel et personnel. Je leur remercie aussi pour leur gentillesse et pour leurs qualités humaines. Je garderai un très bon souvenir de ces mobilités scientifiques.

Je remercie les membres de l'équipe SIG de l'IRIT et à tout le personnel du laboratoire pour leur aide et leur gentillesse. Je tiens à remercier en particulier Marie Françoise Canut, Thierry Millan, Christian Percebois, Lynda Tamine-Lechani, Gilles Hubert, Karen Karen Pinel-Sauvagnat, Riad Mokadem... pour m'avoir fait partager leur expérience en tant qu'enseignants-chercheurs. Je les remercie pour la confiance qu'ils m'ont accordée, pour leurs conseils et pour leur aide.

Je remercie également tous les doctorants avec qui j'ai passé de bons moments pendant ces quatre années. Ainsi, je remercie en particulier mes collègues de bureau Madalina, Firas, Ali et Mohamed. Mes remerciements à tous mes collègues : Arlind, Adrian, Laure, Ana Maria, Lamjed ...

---

Je tiens à remercier Madalina Mitran pour tous les moments que nous avons partagés que je vais me rappeler toujours : les nuits blanches et les weekends de travail à l'IRIT, les pauses café, les conférences et les moments de joie.

Je remercie également à ma colocataire Cristina Stoican et à Annette Reyes et Maria Fernanda Reyes qui ont été à mes côtés tout au long de cette période. Je remercie mes amis : Mohamed, Andrei, Valentin, Jules, Thibaut, Sophie, Audrey, Ismail, Guillaume, Lourdes, Davide . . . Vous avez rendu ma vie tellement facile et je vous en suis reconnaissante.

Enfin et surtout, je remercie du fond du coeur et avec un grand amour toute ma famille, notamment mes parents, Ioan et Dorina et ma soeur Anda. J'avoue que je n'aurais jamais réussi à en arriver là sans leur amour et leur soutien inconditionnel. Ils m'ont toujours encouragé à atteindre mes ambitions et ils ont toujours su me motiver. Le vrai défis de ces années a été d'être loin d'eux.

---

## Résumé

Le nombre de caméras vidéos déployées de nos jours tant dans des contextes professionnels (e.g., dans le cadre des systèmes de vidéo surveillance urbaine) aussi bien que personnels (e.g., caméras des smartphones) augmente de façon exponentielle, générant des volumes de contenus considérables. Rendre le filtrage et la recherche de ces contenus plus efficace est une préoccupation inévitable, avec des exigences de vélocité et de mobilité des contenus liées aux nouvelles infrastructures qui obligent à revitaliser les techniques d'indexation "classiques". Les approches existantes pour répondre à ce besoin se focalisent sur deux axes : (1) la proposition d'outils d'analyse des contenus vidéos pour l'extraction automatique d'informations comme le contour des personnes ou la présence d'une activité "anormale" ; (2) l'indexation des vidéos en utilisant des métadonnées liées aux contenus (e.g., un texte descriptif, des tags, des données de géolocalisation). Nos travaux se situent dans ce second axe.

Le contexte d'application de notre travail est celui des systèmes de vidéosurveillance. Notre recherche a été guidée par différents projets de recherche en collaboration avec la Police Nationale, la SNCF, la RATP et Thales Sécurité. Dans ce contexte, les systèmes visés sont caractérisés par : (1) une grande "variété" des contextes d'acquisition des contenus (e.g., indoor, outdoor), (2) un très grand volume de données et un manque d'accès à certains contenus, (3) la multitude des formats fermés propriétaires et l'absence de standards, qui engendre une hétérogénéité des formats des données et des métadonnées issues de tels systèmes. De ce fait, d'une part, le développement d'outils d'analyse du contenu génériques et performants dans tous les contextes est très problématique compte tenu des diversités des contextes d'acquisition, des volumes à traiter et de l'inaccessibilité directe de certaines sources. D'autre part, l'absence de métadonnées ajoutées aux vidéos (tags, commentaires) rend quasi caduque l'utilisation des approches d'indexation classique.

La première contribution de ce mémoire est une conséquence directe de ce constat et consiste en un dictionnaire de métadonnées spécifique au contexte de la vidéosurveillance. Ce dictionnaire est structuré dans un format qui enrichit la norme ISO 22311 qui a comme objectif la facilitation de l'interopérabilité des systèmes de vidéosurveillance.

La seconde contribution concerne la recherche et le filtrage de vidéos basés sur des métadonnées spatio-temporelles. Nous avons réalisé une étude sur le traitement actuel des requêtes dans le cadre des systèmes de vidéosurveillance qui met en évidence que le point d'entrée de toute requête est une trajectoire reconstituée à partir des

---

positions d'une personne par exemple et d'un intervalle temporel qui est ensuite utilisée pour retrouver des extraits vidéos des caméras qui ont pu filmer une scène d'intérêt. De ce fait, la recherche de vidéos est positionnée comme un problème de modélisation des données spatio-temporelles. Dans ce cadre, nous avons proposé les éléments suivants :

- une définition du concept de requête trajectoire hybride qui est constituée des segments géométriques et symboliques exprimés par rapport à des systèmes de référence différents (e.g., système géodésique, réseau routier) ;
- un modèle de données multicouche qui intègre des données concernant : le réseau routier, le réseau de transport, le mouvement des objets et les changements de champs de vue des caméras ;
- des opérateurs qui, à partir d'une requête trajectoire et d'un intervalle temporel, sélectionnent les caméras fixes et mobiles dont le champ de vue est susceptible d'avoir "filmé" la trajectoire requête.

Nos contributions ont été validées dans le cadre d'un prototype mettant en oeuvre ces trois aspects. Il est basé sur l'API Google Maps pour construire des requêtes hybrides et utilise des data sets fournis dans le cadre de l'open data par différentes collectivités (Transport Collectif de Toulouse).

Notre travail donne lieu à des perspectives multiples qui concernent l'extension du modèle de requête de trajectoire hybride dans un environnement indoor, la participation dans un projet national de mise en place de démonstrateurs dans des situations réelles de vidéosurveillance afin de pouvoir tester le framework sur des données réelles ou le développement d'une architecture d'outil Forensic qui intègre des fonctionnalités de filtrage spatio-temporel et des modules d'analyse de contenu.

*Mots clés* : métadonnées, requêtes spatio-temporelles, système de vidéosurveillance, outil forensic

---

## Abstract

The number of video cameras deployed nowadays in both professional (e.g., urban videosurveillance systems) and personal (e.g., smartphone's cameras) contexts is growing exponentially, producing some considerable volumes of data. Driving the filtering and the retrieval of this content more effective is a major concern, driven by the content mobility and velocity requirements related to the utilization of new technologies, requirements that lead to the need to revitalize the classical indexing techniques. The actual approaches that aim to satisfy these requirements have a twofold orientation : (1) the proposition of video content based indexing tools that automatically extract information like a person's shape or the presence of an "abnormal" activity in the video ; (2) the video indexing based on metadata like textual descriptions, tags or geolocalisation data. Our work concern this second research direction.

The application context of our work is related to videosurveillance systems. Our research was guided by different research projets in collaboration with the National Police, SNCF, RATP and Thales Sécurité. In this context, the targeted systems are characterized by : (1) the big "variety" of content acquisition contexts (e.g., indoor, outdoor), (2) the big data volume and the lack of access to some content, (3) the multitude of system owners and the lack of standards, which leads to a heterogeneity of data and metadata formats generated by videosurveillance systems. Consequently, on one hand, the development of content based indexing tools generic and reliable in all contexts is problematic given the acquisition contexts diversity, the content volume and the lack of direct access to certain sources. On the other hand, the lack of metadata associated to the videos (tags, comments) makes the use of classical indexing approaches very difficult.

The first contribution of this report is a direct consequence of this assessment and consists of a metadata dictionary specific for the videosurveillance context. This dictionary is structured in a format that enriches the ISO 22311 standard whose objective is to facilitate the interoperability of videosurveillance systems.

The second contribution concerns the video filtering and retrieval. We did an analysis of the current query processing mechanism within the videosurveillance systems that highlighted the fact that the entry point of any query is a trajectory reconstituted based on a person's positions and a time interval. These elements are used to select the videos of the cameras that are likely to have filmed the scenery of interest. Consequently, the video retrieval is treated as a spatio-temporal data modelling problem. In this context, we have proposed the following elements :

- 
- a definition of the hybrid trajectory query concept, trajectory that is constituted of geometrical and symbolic segments represented with regards to different reference systems (e.g., geodesic system, road network);
  - a multilayer data model that integrates data concerning : the road network, the transportation network, the objects movement, the cameras fields of view changes ;
  - some operators that, based on a trajectory query and a time interval, select the fixed and mobile cameras whose field of view is likely to have filmed the query trajectory.

Our contributions were validated within a prototype that implement these three elements. This prototype is based on the Google Maps API in order to build the hybrid trajectories and uses the datasets provided by the opendata projects led by different communities (Public Transportation of Toulouse).

Our research contributions lead to many interesting future work perspectives like : extending the hybrid trajectory query model in an indoor environment, joining a national project that aims to set up some demonstrators in realistic videosurveillance contexts so that we could evaluate our framework on real data, developing a Forensic tool architecture that integrates both spatio-temporal filtering functionalities and video analysis modules.

*Keywords* : metadata, spatio-temporal queries, videosurveillance systems , forensic tool





# Table des matières

Résumé . . . . .	5
<b>1 Introduction</b>	<b>13</b>
1.1 Contexte et motivation . . . . .	13
1.1.1 Cas d'utilisation . . . . .	16
1.1.2 Traitement d'une "requête" aujourd'hui dans les systèmes de vidéosurveillance . . . . .	18
1.2 Problématique . . . . .	21
1.3 Contribution . . . . .	23
1.4 Plan . . . . .	24
<b>2 État de l'art</b>	<b>27</b>
2.1 Introduction . . . . .	27
2.2 La "vidéosurveillance intelligente" . . . . .	29
2.3 Modélisation des informations contextuelles . . . . .	33
2.3.1 Modélisation de la localisation et de la trajectoire des objets mobiles . . . . .	34
2.3.2 Modélisation des informations du contexte décrivant le champ de vue d'une caméra . . . . .	47
2.4 Types de requêtes supportés par les systèmes utilisant des meta-données spatio-temporelles . . . . .	49
2.4.1 Position queries (P) . . . . .	49
2.4.2 Range queries (R) . . . . .	50
2.4.3 Visibility queries (V) . . . . .	52
2.4.4 Nearest neighbour queries (NN) . . . . .	52
2.4.5 Nearest Surrounding Queries(NS) . . . . .	53
2.4.6 Trajectory queries(T) . . . . .	53
2.4.7 Predictive queries . . . . .	53
2.5 Bases de données spatio-temporelles . . . . .	54

2.6	Comparaison des approches . . . . .	55
<b>3</b>	<b>Métadonnées pour la vidéosurveillance</b>	<b>63</b>
3.1	Introduction . . . . .	63
3.2	Classification des métadonnées . . . . .	65
3.3	Caméras de vidéosurveillance . . . . .	66
3.3.1	Champ de vue d'une caméra . . . . .	67
3.3.2	Installation d'une caméra . . . . .	72
3.4	Proposition d'un dictionnaire de métadonnées pour la vidéosurveillance	73
3.4.1	Des métadonnées associées aux systèmes de vidéosurveillance .	74
3.4.2	Métadonnées scène observée (description de l'environnement de la scène filmée par la caméra) . . . . .	80
3.4.3	Métadonnées scénario (description des objets et des événements de la scène) . . . . .	82
3.5	Norme ISO 22311 . . . . .	86
3.5.1	Structuration des données vidéo . . . . .	86
3.5.2	Dictionnaire de métadonnées . . . . .	88
3.6	Proposition d'un méta modèle conforme à la norme . . . . .	91
3.7	Conclusion . . . . .	92
<b>4</b>	<b>Interrogation des métadonnées de vidéosurveillance à partir des tra-</b>	
	<b>jectoires hybrides</b>	<b>95</b>
4.1	Introduction . . . . .	95
4.2	Interrogation des données spatio-temporelles à partir d'une trajectoire composée de segments géométriques et/ou symboliques . . . . .	97
4.2.1	Cas d'utilisation de la vidéosurveillance . . . . .	97
4.2.2	Requête trajectoire hybride . . . . .	101
4.3	Modèle multi couches . . . . .	104
4.3.1	Couche Réseau Routier . . . . .	105
4.3.2	Couche Réseau de Transport . . . . .	107
4.3.3	Couche Objets . . . . .	111
4.3.4	Couche Caméras . . . . .	112
4.4	Spécification des opérateurs . . . . .	114
4.4.1	Caméras fixes . . . . .	115
4.4.2	Caméras mobiles . . . . .	116
4.5	Algorithmes de recherche . . . . .	117
4.5.1	Algorithme de sélection des caméras fixes . . . . .	118
4.5.2	Algorithme de sélection des caméras mobiles . . . . .	119

4.6	Exemples de requêtes . . . . .	122
4.6.1	Caméras fixes . . . . .	122
4.6.2	Caméras mobiles . . . . .	125
4.7	Conclusion . . . . .	127
<b>5</b>	<b>Application</b>	<b>129</b>
5.1	Introduction . . . . .	129
5.2	Architecture du framework proposé . . . . .	130
5.2.1	Le module de collecte de données . . . . .	132
5.2.2	Le module interface utilisateur . . . . .	135
5.2.3	Le module d'interprétation de la requête . . . . .	136
5.2.4	Le module de stockage et le module de recherche . . . . .	137
5.3	Exemple de cas d'utilisation réalisé . . . . .	139
5.4	Expérimentations concernant le réseau de transport en commun . . . . .	144
5.5	Conclusion . . . . .	149
<b>6</b>	<b>Conclusions générales et perspectives</b>	<b>153</b>
6.1	Synthèse des propositions . . . . .	153
6.2	Perspectives de recherche . . . . .	155
<b>A</b>	<b>Indoor Location Model</b>	<b>159</b>
A.1	Introduction . . . . .	159
A.2	Trajectory oriented query . . . . .	160
A.2.1	Video Protection Query . . . . .	162
A.2.2	Surveillance of a working area . . . . .	164
A.2.3	Requirements for a heterogeneous trajectory based query framework . . . . .	165
A.3	State of the art : Trajectories in indoor location models . . . . .	166
A.3.1	Geometric location models . . . . .	168
A.3.2	Symbolic location models . . . . .	168
A.3.3	Hybrid models . . . . .	170
A.4	Query template definition . . . . .	170
A.5	Geometrical-Symbolic information mapping . . . . .	172
A.6	Representing topology . . . . .	174
A.7	Indoor coordinate system . . . . .	176
<b>B</b>	<b>GTFS (General Transit Feed Specification) format</b>	<b>179</b>
	<b>Bibliography</b>	<b>179</b>

*TABLE DES MATIÈRES*

---

# Chapitre 1

## Introduction

### 1.1 Contexte et motivation

Le nombre de caméras de vidéosurveillance déployées dans les grandes villes censées assurer la sécurité des citoyens augmente régulièrement. Ce nombre est estimé en 2009 à 5 millions pour le monde entier et il augmente chaque jour [Keval, 2009]. Les chiffres actuels montrent par exemple qu'à Londres le nombre de caméras de vidéosurveillance installées est d'environ 400000. Toute personne passant une journée dans la capitale britannique risque d'être enregistrée en moyenne par plus de 300 caméras. En France, on cite l'exemple de deux systèmes déployés sur Paris. Le réseau de la RATP comprend 9000 caméras installées en stations et gares (ex : pôle Chatelet 350 caméras) et 19000 caméras embarquées ce qui génère environ 2PB de stockage sur une durée de 3 jours (700000 heures de vidéo). Celui de la SNCF (dans l'Île de France) est estimé à environ 4600 caméras fixes et 6600 embarquées dans des trains (e.g., Gare du Nord 551 caméras) et génère des volumes d'environ 1,7PB de stockage sur quelques jours. Selon la SNCF, en septembre 2013 la Police avait procédé à 3000 réquisitions sur Paris. C'est dans ce contexte que se situe notre travail, dans l'objectif de mettre en oeuvre la recherche de preuves a posteriori dans les collections vidéos issues des systèmes de vidéosurveillance.

Le schéma général d'un système de vidéosurveillance, tel qu'illustré en Figure 1.1, montre comment le contenu est acquis de façon distribuée par des caméras installées dans des lieux publics ou privés dans des contextes différents (extérieur ou intérieur des bus, des stations de métro, dans les rues, dans ou devant des centres commer-



FIGURE 1.1 – Schéma d'un système de vidéosurveillance

ciaux, etc.). Ces caméras sont fixes ou mobiles, ont des caractéristiques techniques hétérogènes et appartiennent aux différents systèmes (avec des propriétaires et dans des formats différents) qui ne sont pas interopérables.

Les vidéos sont stockées sur des serveurs et analysées depuis des centres de contrôle. Aujourd'hui cette analyse est purement manuelle et elle est réalisée par des opérateurs qui scrutent plusieurs écrans disposés en matrice et encodent si nécessaire via un clavier avec des touches dédiées : mur d'images ("video wall") de la Figure 1.1. Les images restituées via ces écrans ne sont pas organisées spatialement. Très peu d'informations ou métadonnées sont disponibles (un identifiant de la caméra, un "timestamp", éventuellement une localisation mais pas très précise) pour l'opérateur qui ne peut que se référer au numéro de la caméra et à son expertise personnelle pour situer celle-ci en fonction des autres. Sa tâche consiste donc à exploiter (visualiser à une vitesse entre ralenti, normal et accéléré) manuellement des flux vidéo mis côte à côte sans aucune organisation globale. Cette exploitation manuelle se trouve encore plus limitée avec les volumes croissants de données générées, avec une surcharge cognitive liée au nombre de caméras, à leur mobilité et à l'importance des flux à analyser.

Plusieurs études ont adressé ce problème de surcharge cognitive. [Keval, 2009] a réalisé une étude dans 14 centres de sécurité à Londres ("control rooms") et les résultats ont montré que les opérateurs doivent suivre dans certains cas jusqu'à quinze

écrans simultanément. Cela, au-delà des problèmes de concentration, de fatigue et d'ennui, pose le problème de l'acuité et de la pertinence de l'observation, génératrices d'erreurs ou d'approximations.

Les choses se compliquent beaucoup quand il faut faire analyser des vidéos issues de plusieurs systèmes par des opérateurs extérieurs, ce qui arrive souvent dans le cas des enquêtes judiciaires complexes. Le besoin des opérateurs est d'arriver à rapidement faire usage des données recueillies par les différents systèmes de vidéosurveillance couvrant une certaine zone (trajectoire) donnée. Pour cela il faut que les sorties des différents systèmes soient interopérables. L'interopérabilité des systèmes de vidéosurveillance concerne deux aspects : le format de communication des contenus vidéo et les métadonnées (caractérisant les conditions de prise de vue, les caractéristiques techniques de la caméra, les caractéristiques de la scène enregistrée, etc.) modélisant les annotations textuelles et les résultats des analyses du signal vidéo. Le besoin d'interopérabilité fait que le problème de la normalisation est aigu dans le domaine de la vidéosurveillance [Sedes et al., 2012]. En conséquence, les principaux acteurs de ce domaine en France ont réuni leurs efforts pour développer la norme ISO 22311<sup>1</sup> qui propose un format d'export (une structure et un dictionnaire) des données et des métadonnées issues des systèmes de vidéosurveillance. La norme se proposant d'offrir un cadre et non pas d'être exhaustive, des extensions peuvent être apportées.

Pour mieux illustrer la difficulté, nous allons nous focaliser sur l'analyse de la façon dont une recherche est effectuée dans un système gérant un réseau de vidéosurveillance. Dans le cas d'une recherche dans une collection d'enregistrements, l'objectif principal est de localiser (date et heure, lieu) les faits (e.g., vol du sac, agression), puis de reconstituer le cheminement avant et après les faits afin d'obtenir des éléments d'identification (vue de face / marque et modèle de véhicule, couleur et numéro de plaque). Le résultat d'une telle recherche consiste en une liste d'images preuves et de clips vidéo de qualité suffisante (qui permet la détection des faits ou l'identification des individus et donc qui puisse constituer preuve dans un procès). En conséquence, le point d'entrée de toute enquête (ce que nous allons appeler "requête") est constitué par les informations de localisation dans l'espace et dans le temps, notamment la description de la trajectoire de la victime (son déplacement dans un intervalle de temps).

Nous avons fait une analyse de l'exécution d'une recherche dans une collection vidéo d'un point de vue "système d'information" c'est-à-dire dans une vision globale d'in-

---

1. [http://www.iso.org/iso/fr/catalogue\\_detail.htm?csnumber=53467](http://www.iso.org/iso/fr/catalogue_detail.htm?csnumber=53467)

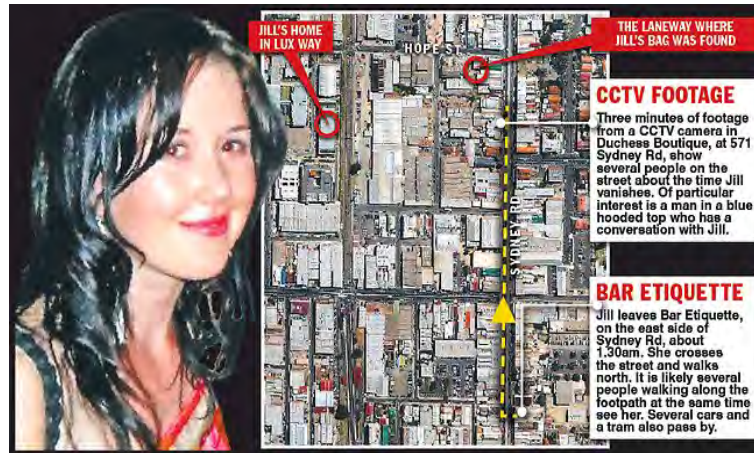


FIGURE 1.2 – Reconstitution de la trajectoire de la disparue de Sidney

terrogation d'un utilisateur et de restitution d'un résultat intelligible. Nous allons utiliser le terme générique "requête" pour désigner la formulation du besoin de l'utilisateur d'un système de vidéosurveillance dans une tâche de recherche et "d'interrogation". Par la suite, nous allons présenter des cas concrets d'utilisation du système gérant un réseau de vidéosurveillance pour des enquêtes judiciaires, en analysant la façon dont les opérateurs et les enquêteurs ont abordé la "requête".

### 1.1.1 Cas d'utilisation

Prenons l'exemple d'une disparition à Sidney<sup>2</sup> : les enquêteurs ont créé une page en ligne via laquelle toute personne pensant posséder une information utile pouvait participer. Ils ont ainsi pu reconstituer la trajectoire de la personne disparue (voir Figure 1.2) et finalement identifier, dans les enregistrements vidéo d'une caméra installée dans une boutique, l'agresseur (voir Figure 1.3).

Un autre cas est celui du tireur du siège de Libération à Paris le 18 novembre 2013. A partir des témoignages des policiers et d'autres personnes impliquées, les enquêteurs ont pu reconstituer le cheminement avant et après les faits (voir Figure 1.6). La description physique faite par des témoins amène à la conclusion que c'est la même personne qui avait menacé des employés de BFMTV trois jours avant. La

2. <http://www.news.com.au/national-news/victoria/one-of-six-people-seen-on-crucial-cctv-comes-forward-as-police-probe-abduction-theory-on-missing-jill-meagher-and-plea-for-witnesses/story-fndo4cq1-1226482146903>





FIGURE 1.3 – Images prises par la caméra installée dans la boutique

vidéosurveillance a été un outil essentiel dans l'enquête qui a mené à l'interpellation et à l'identification de l'agresseur<sup>3</sup>. Connaissant sa trajectoire et ses caractéristiques physiques, le but a été de trouver des images claires à partir desquelles on pourrait identifier la personne.

Voici le cheminement détaillé de l'agresseur :

Le 15 novembre 2013 à 06h42 une image capturée par une caméra de vidéosurveillance installée sur les quais de la station de Tramway de Porte d'Issy montre le suspect assis sur un des bancs (voir Figure 1.6).

Quelques minutes plus tard, à 06h53, Porte de Versailles, le même homme, armé d'un fusil à pompe menace deux journalistes de BFMTV. L'image (voir Figure 1.6) a été capturée par une caméra de vidéosurveillance de BFMTV.

Le 18 novembre 2013 vers 10h15, le suspect est filmé par une caméra du réseau public parisien place de la République (voir Figure 1.6).

2 minutes après, l'homme, armé d'un fusil à pompe, fait irruption dans le hall de Libération, rue Béranger, dans le 11e arrondissement de Paris. Il ouvre le feu et blesse grièvement l'assistant d'un photographe du quotidien avant de prendre la fuite.

Le suspect prend le métro pour se rendre dans le quartier de la Défense où à 11h40 il fait feu à au moins trois reprises dans la direction du bâtiment de la Société Générale.

Ensuite, l'agresseur va détourner une voiture qui empruntera l'avenue des Champs-

---

3. [http://mobile.lemonde.fr/societe/article/2013/11/19/traque-du-tireur-a-paris-la-police-peut-s-appuyer-sur-un-dense-reseau-de-videosurveillance\\_3516603\\_3224.html](http://mobile.lemonde.fr/societe/article/2013/11/19/traque-du-tireur-a-paris-la-police-peut-s-appuyer-sur-un-dense-reseau-de-videosurveillance_3516603_3224.html)

Elysées et qui va le déposer sur l'avenue George V près de la station de RER Pont de l'Alma.

La personne continue vers Place de la Concorde et entre dans la station de métro. A 12h30 il est filmé par une caméra installée dans la station de métro Concorde (voir Figure 1.5). C'est cette image vue de face qui permet son identification.

### **1.1.2 Traitement d'une "requête" aujourd'hui dans les systèmes de vidéosurveillance**

Dans le cadre du projet ANR METHODEO, à partir des entretiens menés avec des enquêteurs et des opérateurs de la Police Nationale et la RATP, nous avons pu analyser la façon dont une demande (la "requête") est traitée aujourd'hui dans un système de vidéosurveillance. Quand une personne (e.g., la victime d'une agression) porte plainte, on lui demande de compléter un formulaire pour décrire les éléments qui pourront aider les opérateurs à trouver les segments vidéos pertinents (voir Figure 1.4). Les principaux éléments d'un tel formulaire sont : la localisation, la date et l'heure, l'événement, la trajectoire de la victime avant et après l'événement et éventuellement des signes distinctifs qui pourraient être repérés facilement dans la vidéo (e.g., la couleur des vêtements). En conséquence, les requêtes ont des composantes spatiales et temporelles et seront mises en correspondance avec les métadonnées de géo-localisation et temporelles associées aux vidéos issues de différents systèmes de vidéosurveillance. Ce "matching" est réalisé manuellement par l'opérateur humain qui se base sur l'information spatiale et temporelle de la "requête" et sur sa propre connaissance de la localisation des caméras pour sélectionner les caméras qui pourront avoir filmé des éléments pertinentes pour la requête, pour les analyser ensuite.

En analysant les cas d'utilisation et le traitement d'une plainte portée par une victime, nous avons fait les observations suivantes : (1) le point de départ de toute recherche dans une collection de vidéo surveillance est l'information spatiotemporelle, (2) le trajet de la victime ou de l'agresseur avant et après les faits est reconstitué et en se basant sur celui-ci, on essaye d'extraire les images permettant d'identifier le(s) agresseurs, suspects, pick-pockets, etc.

Localisation : **Toulouse**  
Date et Heure : **7 Février 2014 entre 14h et 14h30**  
Evénement : **vol de sac**  
Cheminement : **Route de Narbonne : sortie du Metro Paul Sabatier -> intersection avec le Chemin de Maraichers**  
Signes distinctifs/Couleurs Vêtements/Portrait robot

FIGURE 1.4 – Exemple de requête



FIGURE 1.5 – Image prise par la caméra de vidéosurveillance installée dans la station de métro Concorde

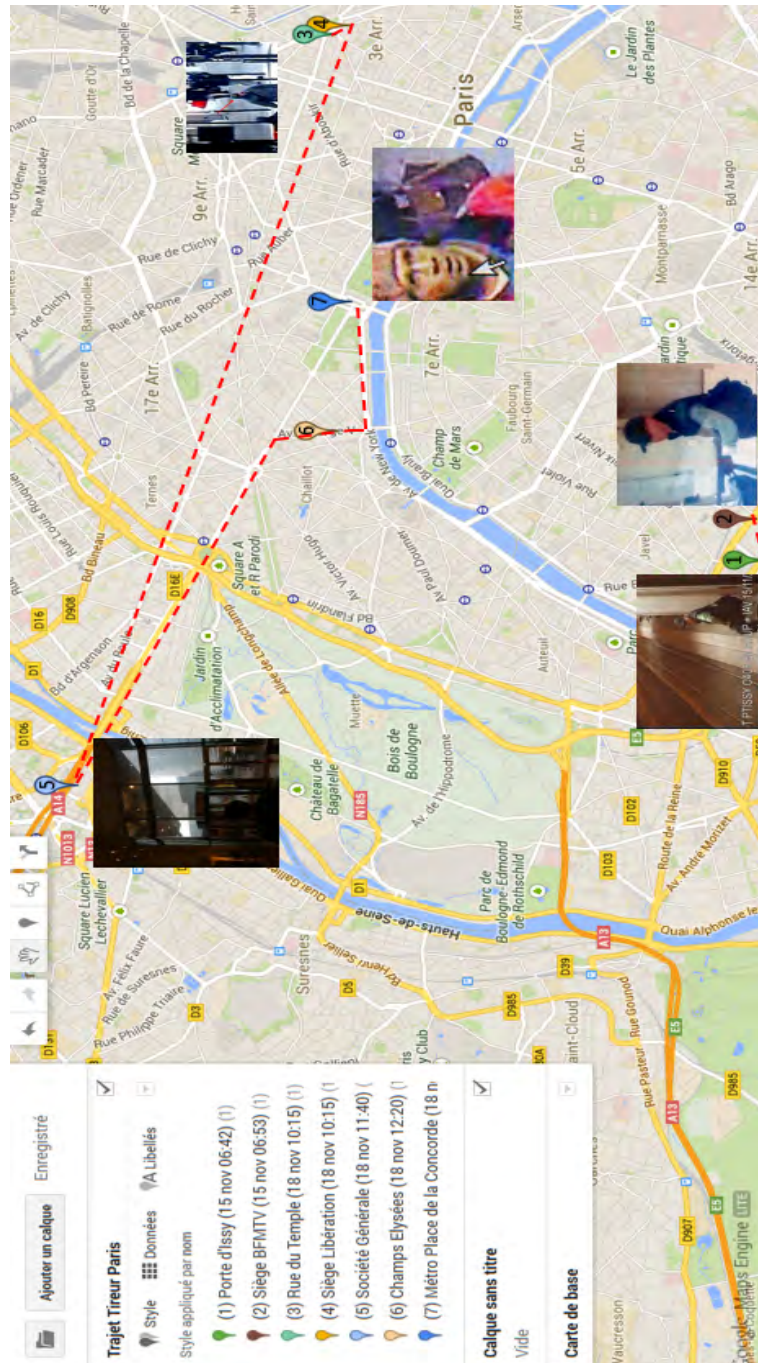


FIGURE 1.6 – Le trajet du tireur

## 1.2 Problématique

A partir de l'analyse du contexte d'usage de la vidéosurveillance, la principale question qui se pose est : quelle "boîte à outils" fournir aux opérateurs pour les assister dans leur tâche et les aider à réduire le temps de recherche, tout en favorisant l'interopérabilité des différents systèmes de vidéosurveillance ? C'est sous l'angle du système d'information, et non par une approche de traitement du signal pur, que nous abordons ce problème.

Un système d'informations est un ensemble de ressources matérielles, humaines et techniques [Denning, 2000] dont le rôle est la collecte, le stockage, le traitement et la diffusion de l'information. Dans le cas de la plupart des systèmes de vidéosurveillance, nous avons une collection de contenus vidéos qui sont visualisés par les opérateurs en utilisant des lecteurs qui permettent la visualisation à différentes vitesses. La préoccupation de créer des systèmes de vidéosurveillance dits intelligents est d'actualité parmi les chercheurs et industriels du domaine. Le but est de développer des outils basés sur les données et sur les métadonnées pour aider les opérateurs dans leur tâche. C'est la problématique à laquelle nous allons nous intéresser dans cette thèse.

Les systèmes de vidéosurveillance représentent un type de système d'information dont les particularités sont :

- Contrainte de temps de réponse très importante surtout dans le cas des situations d'urgence pour les opérateurs, ce qui rend le rappel beaucoup plus important que la précision (il est crucial de ne pas rater d'informations, quitte à en fournir trop) ;
- Grande diversité des contextes d'acquisition : type de caméra, installation de la caméra (fixe/mobile), luminosité, prise de vue, paramètres de compression, etc. ;
- Qualité très faible des enregistrements donc très souvent mauvaise performance des algorithmes d'analyse du contenu ;
- Inexistence des métadonnées comme celles disponibles pour les vidéos du web (e.g., texte autour des vidéos, titre, commentaires etc.) ;
- Manque d'une image "requête" dans beaucoup de situations (à part des portraits robot) au moins pour une première "itération" de la recherche ;
- Grand nombre de systèmes gérés par des entités différentes ayant leur propre format de données et de métadonnées ce qui a comme conséquence un manque d'interopérabilité.

Dans le cadre des systèmes d'informations multimédias, il y a deux types d'approches pour l'indexation des collections : (1) indexation basée sur des outils d'analyse du signal vidéo ou audio et (2) indexation basée sur l'intervention humaine ou sur d'autres

informations (e.g., générées par d'autres types de capteurs comme par exemple des capteurs GPS ou des boussoles, ou extraites directement de la caméra comme les paramètres de compression ou des indicateurs de qualité d'image).

En suivant la même classification, nous trouvons deux types d'approches d'interrogation des contenus stockés : (1) méthodes basées sur le contenu et (2) méthodes basées sur les métadonnées. Dans le premier cas, la requête est une image (ou une vidéo) qui est analysée par le système pour calculer des descripteurs qui seront comparés à ceux caractérisant les images/vidéos de la collection. Les objets les plus similaires sont retournés. Dans le deuxième cas, la requête est "textuelle" (les guillemets marque le fait que le texte de la requête peut désigner une localisation ou une géométrie décrite par des coordonnées GPS par exemple) et elle est comparée avec les métadonnées décrivant les contenus vidéos (e.g., des annotations textuelles manuelles, semi-automatiques ou automatiques, des descriptions extraites de la même page web ou du titre, des coordonnées de géolocalisation).

Beaucoup de recherches sont menées dans le domaine de l'analyse et de l'interrogation des collections issues des systèmes de vidéosurveillance en se basant sur le contenu vidéo ([Lew et al., 2006], [Snoek and Worring, 2009]). Les principaux problèmes qui se révèlent en essayant d'utiliser le premier type d'approche dans le cas des systèmes de vidéosurveillance sont : (i) difficulté de développer des algorithmes d'analyse du contenu vidéo performants dans toutes les conditions d'acquisition des caméras de vidéosurveillance (beaucoup de progrès ont été faits dans le développement d'outils d'analyse vidéo mais il reste difficile d'utiliser ces algorithmes hors des domaines spécifiques d'applications pour lesquels ils ont été entraînés comme les journaux télévisés ou les vidéos des sports), (ii) grand volume de données donc coût d'exécution élevé et (iii) manque d'accès à certaines données vidéo (e.g, des contenus vidéo issus du système de vidéosurveillance d'une banque).

Pour récapituler, les principaux problèmes générés par le traitement actuel (manuel) qui porte sur des données issues de plusieurs systèmes de vidéosurveillance différents sont : (1) l'hétérogénéité des formats et données provenant des différents systèmes, (2) le temps de réponse, long (jusqu'à plusieurs jours de traitement) à cause du grand volume de données vidéos et du manque d'interopérabilité des données et des métadonnées (3) le risque de perte de résultats à cause d'une mauvaise connaissance du terrain (appel à des opérateurs extérieurs, méconnaissance des caméras).

Dans ce contexte là, nous proposons de nous baser sur les métadonnées pour un filtrage du contenu et une exécution "intelligente" des algorithmes d'indexation sur des sous-parties de la collection vidéo. Notre approche est complémentaire des approches

d'analyse du contenu vidéo (qui sont hors du cadre de ce travail) et ensemble peuvent constituer une solution puissante de recherche et d'indexation des contenus vidéos.

## 1.3 Contribution

A partir de l'énoncé de ces problèmes, nous formulons les questions auxquelles nous répondons dans le cadre de notre contribution :

- Quelles sont les métadonnées utiles pour la vidéo surveillance ? Quel format d'annotation favorise l'interopérabilité des systèmes ? Ces informations sont-elles disponibles assez rapidement ?
- Comment réduire l'espace de recherche et implicitement le temps de réponse ?

Les efforts menés pour répondre à ces questions ont conduit à une double contribution qui sera exposée dans le cadre de cette thèse et validée par une expérimentation qui prouve la faisabilité et l'apport des modèles développés.

La contribution théorique est articulée en deux volets :

- Une étude des métadonnées utiles pour la recherche dans les collections de vidéos, dans le contexte très particulier de la vidéosurveillance, qui induit des contraintes spécifiques, nous a permis de dresser un inventaire (non-exhaustif) des métadonnées utiles dans le cadre de la vidéosurveillance. Les principales catégories sont : (1) métadonnées décrivant le système de vidéosurveillance, (2) métadonnées décrivant la scène (l'environnement et les objets et événements de contexte) et (3) métadonnées décrivant le scénario (les objets et les événements). Afin de définir une méthode réaliste pour passer de la description d'une scène de crime à des requêtes techniques sur les données rendues disponibles par les algorithmes de traitement, il faut définir un modèle générique qui va englober toutes les métadonnées définies auparavant dans un format qui facilitera l'interopérabilité des sorties des différents systèmes. Ceci nous a conduit à proposer un format de métadonnées pour la vidéosurveillance basé sur la norme ISO 22311.
- En se basant sur une partie de ces métadonnées (celles relevant de la composante spatiotemporelle), notre deuxième contribution a pour but d'aider les opérateurs humains de vidéosurveillance dans l'analyse manuelle d'extraits vidéos particuliers (e.g., agression, objet perdu) via l'identification automatique d'un ensemble de caméras susceptibles d'avoir filmé une scène recherchée. Nous proposons une modélisation multi-couche (caméras fixes et mobiles, réseau routier et de transport en commun) dont la contribution consiste à inclure des données de calcul de

la géométrie du champ de vue des caméras. Plutôt que de stocker cette géométrie en tant que telle, elle est calculée en dynamique en fonction de caractéristiques temporelles (e.g., angle de vue, distance focale). Nous proposons des algorithmes de sélection de caméras basés sur des critères spatio-temporels de calcul des intersections entre, d'une part, un parcours donné (cible e.g., personne agressée), et, d'autre part, les prises de vues des caméras fixes et les trajectoires des caméras mobiles. Les requêtes spatio-temporelles ainsi traitées sont directement exploitables dans le cadre des opérations d'extraction des images preuves pour les enquêtes judiciaires.

En se basant sur les contributions théoriques, nous pouvons valider la contribution applicative de notre travail qui est de montrer la faisabilité et les bénéfices de l'utilisation des informations (e.g., de géolocalisation, caractéristiques optiques de la caméra) associées aux contenus vidéo dans un modèle qui pourra constituer la base d'un système d'assistance aux opérateurs de vidéosurveillance. Toutes ces contributions assureront la concrétisation d'un outil Forensic [Sedes et al., 2012] (qui aide les opérateurs dans une enquête a posteriori) permettant à la Police d'effectuer des requêtes efficaces pour la recherche d'évènements dans les enregistrements vidéo sélectionnés par la première étape de filtrage spatio-temporel.

## 1.4 Plan

Le manuscrit est organisé en cinq parties :

- Un **état de l'art** qui commence par la présentation du domaine de la vidéosurveillance "intelligente" dans le contexte des travaux menés aussi dans la communauté de recherche que dans celle industrielle. Une fois le besoin identifié de s'appuyer sur des métadonnées décrivant le contexte d'acquisition des contenus vidéos pour le filtrage de ceux-ci avant une analyse automatique du contenu, nous allons présenter un résumé du domaine de la modélisation des informations décrivant d'un côté le contexte spatio-temporel des objets cibles et des capteurs vidéos (e.g., position et changement de la position dans le temps) et d'un autre côté l'aire d'action des capteurs vidéos (e.g., orientation, distance visible). Par la suite nous allons présenter un état de l'art des travaux qui ont utilisé ces informations pour filtrer, annoter, rechercher etc. des contenus multimédias ou des données issues des capteurs ou décrivant les capteurs.
- Le troisième chapitre constitue un zoom sur les **Métadonnées pour la vidéosurveillance** dans le contexte du besoin de l'interopérabilité de ces



métadonnées accentué par la publication de la norme ISO 22311. Nous présentons dans ce chapitre une proposition d'enrichissement de la norme 22311 que nous avons fait en se basant sur un étude des métadonnées utiles pour le contexte de la vidéosurveillance que nous avons réalisé dans le cadre du projet ANR METHODEO dans le cadre duquel nous avons participé en temps que partenaires.

- En se basant sur l'état de l'art présenté et sur l'étude du contexte de la vidéosurveillance étudié dans le cadre du projet nous allons décrire notre principale contribution concernant la **Modélisation spatio-temporelle dans le cadre de la vidéosurveillance**.
- L'implémentation du modèle de données et des algorithmes de recherche proposés et leur validation est décrite dans **Application et expérimentation**.
- Nous concluons par le bilan et les perspectives de notre travail.



# Chapitre 2

## État de l'art

### 2.1 Introduction

Le nombre de capteurs déployés dans les rues et à l'intérieur des stations de métro et des différents bâtiments augmente rapidement en constituant la base d'une palette très large d'applications comme : les applications géospatiales mobiles qui permettent de s'abonner à différents services qui se basent sur la localisation [Simon and Fröhlich, 2007] (e.g., recommandations des lieux, des chemins vers des points d'intérêt [Retscher, 2007]), d'annoter et de rechercher des images et des vidéos [Kim et al., 2010], [Debnath and Borcea, 2013], de mesurer certaines valeurs (e.g., température, consommation d'énergie), de détecter des intrusions, d'interroger la position d'un objet à un moment donné ou de suivre sa trajectoire (e.g., RFID<sup>1</sup>, Wifi [Beal, 2003], [Scuturici and Ejigu, 2006], GPS<sup>2</sup>), de surveiller certaines zones [Amriki and Atrey, 2014]. Dans ces types d'applications, les utilisateurs soumettent des requêtes (mots clés ou spatio-temporelles) afin de retrouver le chemin vers un point d'intérêt, retrouver les restaurants sur leur itinéraire, retrouver les objets ou les capteurs qui se trouvent dans une région ou sur une trajectoire, retrouver les images ou les vidéos qui ont "vu" une certaine région ou objectif, calculer la distance entre certains objets, etc.

Le concept de donnée spatio-temporelle est utilisé afin de désigner le changement d'une géométrie (de sa position ou sa forme) avec le temps. Le besoin croissant

---

1. <http://www.rfid-handbook.de/>

2. <http://www.gps.gov/>

de gérer et de retrouver des informations concernant le mouvement des différents objets a mené à l'apparition des concepts de base de données spatio-temporelle et de requête spatio-temporelle [Erwig and Schneider, 2002b]. Le passage d'une base de données classique (relationnelle) à une base de données spatio-temporelle a été réalisée par l'intégration des données spatio-temporelles en temps que types abstraits de données (e.g., point, ligne, polygone) dans les modèles de données déjà existants [Güting et al., 2000], [Sandu-Popa and Zeitouni, 2012] et le développement des opérateurs et des langages d'interrogation spécifiques à ces nouveaux types de données [Erwig and Schneider, 2002b]. Plus précisément, nous considérons comme requête spatio-temporelle toute interrogation de l'utilisateur comportant au moins un élément de géolocalisation ou de situation spatiale (exprimée en termes de coordonnées numériques ou de noms symboliques) et/ou une référence temporelle (date, heure, intervalle).

Les capteurs sont définis comme des objets physiques qui réalisent une observation<sup>3</sup>, plus précisément qui transforment un stimulus entrant (e.g., lumière, température, son) en une information, numérique la plupart du temps. Pour une description détaillée des réseaux de capteurs et des différentes catégories de capteurs, on se référera à [Compton et al., 2009]. Les capteurs utilisés dans les applications mentionnées plus haut sont classifiés dans plusieurs catégories : capteurs de localisation (ceux qui donnent la position des objets comme par exemple les capteurs GPS, RFID, Wifi), capteurs vidéo (e.g., caméras de vidéosurveillance), capteurs audio, capteurs qui retournent certaines valeurs mesurées (e.g., capteurs de température, de consommation d'énergie), capteurs qui donnent l'orientation comme la boussole ou qui donnent l'inclinaison comme l'accéléromètre. Les données générées et liées au premier type de capteurs sont stockées et gérées dans des modèles de données spatio-temporelles ou modèles de localisation [Afyouni et al., 2012]. En fonction des types d'objets et des capacités des capteurs associés aux objets, d'autres types d'informations peuvent être associés aux données spatio-temporelles (e.g., pour une caméra c'est intéressant d'associer les métadonnées décrivant son champ visuel comme expliqué dans le chapitre précédent ou des données techniques comme la variation de la qualité des images capturées). Toutes ces informations sont stockées et gérées dans des bases de données spatio-temporelles [Forlizzi et al., 2000], [Schneider, 2009] et des bases de données issues de capteurs [Bonnet et al., 2001] qui constituent le fondement des systèmes qui visent ajouter de "l'intelligence" à l'environnement [Sèdes et al., 2012].

Nous commençons ce chapitre par la présentation d'un type spécifique de systèmes, ceux relatifs à la vidéosurveillance car c'est ceux-ci que nous avons étudiés suite aux

---

3. <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/>

projets que nous avons menés. En premier lieu, nous présentons l'évolution dans la direction de ce qu'on appelle les systèmes de vidéosurveillance intelligents qui englobent beaucoup de technologies et techniques au niveau hardware (e.g., réseaux de caméras et des capteurs associés), software (e.g., techniques de filtrage de vidéos, algorithmes d'analyse des contenus) et social (e.g., applications de crowdsourcing). Par la suite, nous détaillons les différentes approches de modélisation des informations issues des différents capteurs qui peuvent être associées aux contenus vidéos capturés par les caméras de vidéosurveillance (nous allons utiliser le concept d'informations ou métadonnées contextuelles pour se référer à ceux-ci). Pour chaque type de métadonnées, nous présentons brièvement les principaux défis qu'il faut surmonter pour collecter ces métadonnées. Ensuite nous analysons les différents types de requêtes auxquelles on peut répondre en se basant sur les informations contextuelles. Nous concluons par une comparaison des différents systèmes qui s'appuient sur les métadonnées contextuelles pour des applications diverses.

## 2.2 La "vidéosurveillance intelligente"

La perception des systèmes de vidéosurveillance actuels a beaucoup changé depuis des événements malheureux comme les attaques terroristes de Madrid (en mars 2004), de Londres (en juillet 2005), l'assassinat du consultant du premier ministre italien Prof. Marco Biagi à Bologna (en 2002), ou le cas du tireur de Libération à Paris (en 2013), qui ont en commun le fait que les enregistrements vidéo issus des systèmes de vidéosurveillance ont joué un rôle crucial dans l'identification des coupables. En conséquence, les "attentes" concernant les systèmes de vidéosurveillance ont beaucoup augmenté, ces systèmes n'étant plus vus comme un ensemble d'enregistrements qui sont regardés par des opérateurs, mais comme des systèmes complexes composés des différents types de capteurs (vidéo, audio, GPS, Wi-fi, laser, etc.), de différents types de caméras (caméras fixes, caméras PTZ dont le champ de vue peut changer, caméras multi-spectrales, etc.) qui devraient être capables non seulement de visualiser du contenu vidéo, mais d'interpréter les différents types de données générées par les différents types de capteurs afin d'attirer l'attention des opérateurs vers certains événements anormaux, de déclencher des alarmes, etc.

En France, le besoin d'outils pour la vidéosurveillance en fonction des 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) sont [Dufour, 2012] :

– **la prévention et la sécurisation** dont le but est d'assurer la sécurité des lieux et

- des événements par l'intermédiaire de la vidéo surveillance en temps réel à savoir un besoin d'outils d'aide à la recherche en temps réel afin de focaliser l'attention des agents de sécurité sur les événements pertinents (e.g., alarmes),
- **le renseignement (temps réel et a posteriori)** qui se propose de recueillir des informations plus ciblées sur une personne ou une entreprise par exemple. Dans ce cas, il y a un besoin d'outils d'assistance comme des alarmes qui se déclenchent si des événements prédéfinis arrivent (e.g., la détection d'une personne dans une zone interdite) ou des outils d'analyse du contenu (lecture labiale, détection d'une personne, etc.), ou d'amélioration de la qualité de l'image,
  - **l'enquête** qui est la mission la plus complexe car elle implique le recoupement de beaucoup d'informations. La recherche de preuves dans les vidéos issues des caméras de vidéosurveillance nécessite une structure matérielle et applicative pour implémenter et exécuter des filtrages du contenu, des traitements d'images et de vidéo, permettre la gestion des résultats générés pour faciliter l'exploitation et permettre des recherches sémantiques et contextuelles sur les vidéos.

La plupart des efforts (de recherche et commerciaux) se concentrent sur le développement des outils concernant les premières deux missions de sécurité ([Girgensohn et al., 2006], [Kraus, 2012]),<sup>4</sup> les travaux concernant le développement des outils pour la recherche des preuves a posteriori dans les collections vidéos étant moins nombreux et moins mûrs.

L'intérêt pour le domaine évoqué est également illustré par le grand nombre de projets de recherche qui ont essayé de proposer des solutions efficaces au problème de la recherche des extraits vidéo "preuves" dans une collection de vidéosurveillance. La plupart des projets se focalisent sur le développement des outils d'analyse des contenus vidéo afin de détecter des activités [Girgensohn et al., 2006], [Chleq et al., 1999], des événements [Tian et al., 2008], détecter des "patterns" de trajectoires "normales" [Aravecchia et al., 2010], détecter des objets, détecter ou suivre des personnes [Cucchiara, 2005], [Cupillard et al., 2002], etc.

En essayant d'analyser de façon exhaustive les contenus vidéos issues des systèmes de vidéosurveillance, les problèmes rencontrés sont : (i) difficulté de développer des algorithmes d'analyse du contenu vidéo performants dans toutes les conditions d'acquisition des caméras de vidéosurveillance (beaucoup de progrès ont été faits dans le développement d'outils d'analyse vidéo mais il reste difficile d'utiliser ces algorithmes hors des domaines spécifiques d'applications pour lesquels ils ont été entraînés comme les journaux télévisés ou les vidéos des sports) [Chu and Smeulders, 2010], (ii) grand

---

4. <http://www.objectvideo.com/>

volume de données donc coût d'exécution élevé et (iii) manque d'accès à certaines données vidéo (e.g, des contenus vidéo issus du système de vidéosurveillance d'une banque) (iv) manque de la capacité à définir précisément ce que l'on cherche et d'une calibration (ou un apprentissage) pour chaque caméra.

Dans ce qui suit, nous allons présenter les principaux projets qui proposent des solutions de gestion des contenus vidéo issus des systèmes de vidéosurveillance en évoquant leur objectif principal et s'ils proposent un filtrage des contenus avant l'analyse basée sur les caractéristiques vidéo.

Le projet CANDELA propose une architecture distribuée générique pour l'analyse et la recherche des contenus vidéos [Merkus et al., 2004]. L'analyse exhaustive des contenus vidéos est réalisée au moment de l'acquisition. Le projet CARETAKER<sup>5</sup> a développé des techniques permettant l'extraction automatique des métadonnées sémantiques à partir du contenu vidéo. Aucun pré-filtrage du contenu n'est réalisé avant l'extraction des descripteurs vidéos.

Plus proche de nos travaux, le projet européen VANAHEIM<sup>6</sup>, en se basant sur des algorithmes de détection des activités anormales, propose une technique pour filtrer automatiquement (en temps réel) les vidéos visualisées par les opérateurs humains sur le mur d'images ("video wall"). Toutefois, le filtrage est basé sur des algorithmes d'apprentissage basés sur l'analyse du contenu qui nécessitent l'utilisation de beaucoup de données et qui semblent difficiles à implémenter à une large échelle.

Le projet SURTRAIN<sup>7</sup> a comme but d'assurer la sécurité des voyageurs par l'implémentation d'un système de transport intelligent. Cela implique l'installation d'un système d'enregistrement des vidéos et des sons à l'intérieur de la rame avec des composants logiciels d'analyse de l'image et du son pour en premier lieu détecter et localiser les cris, puis une fois que la caméra la plus proche de l'événement est sélectionnée et mise en premier plan, démarrer les outils de suivi et d'identification de personnes. Pourtant, l'approche proposée n'offre pas de solution de traitement des requêtes a posteriori.

Notre équipe a collaboré sur deux projets qui concernent la problématique de la vidéosurveillance. Nous allons citer dans ce chapitre le projet européen LINDO (Large scale distributed INDEXation of multimedia Objects) [Brut et al., 2011a]. La problématique centrale du projet LINDO était d'indexer et rechercher des segments

---

5. [http://cordis.europa.eu/ist/kct/caretaker\\_synopsis.htm](http://cordis.europa.eu/ist/kct/caretaker_synopsis.htm)

6. <http://www.vanaheim-project.eu/>

7. <http://www.agence-nationale-recherche.fr/informations/l-actu-en-video/surtrain-surveillance-des-transports-par-lanalyse-de-limage-et-du-son/>

pertinents de contenus multimédias stockés de façon distribuée. L'objectif principal du projet consistait à établir une architecture distribuée d'indexation des contenus multimédias, dans laquelle les algorithmes d'indexation sont déployés sur les différents sites distants où les contenus multimédias sont stockés, afin de réduire la quantité de données véhiculées sur le réseau. La problématique de l'inefficacité de l'indexation exhaustive des contenus acquis en utilisant tous les extracteurs disponibles a été adressée par la mise en place d'un processus de sélection de algorithmes à indexer en fonction du besoin de l'utilisateur [Codreanu et al., 2012].

Les éléments cités montrent que de nombreux travaux sont menés pour le développement de "systèmes de vidéosurveillance intelligents". Beaucoup de travaux visent le développement d'outils d'analyse du contenu [Cucchiara, 2005] mais les problèmes de volumétrie, d'hétérogénéité des contextes d'acquisition, de qualité très variable des enregistrements et de manque d'accessibilité à certains contenus (e.g., vidéos privées) rendent inutile voire impossible l'exécution des algorithmes d'analyse du contenu à cause des mauvaises performances ou de la faible qualité des résultats obtenus. En conséquence, après plus de vingt ans de recherche dans le domaine de l'analyse des contenus audio visuels, à l'heure actuelle en France il n'y a pas de logiciel d'analyse de contenu qui soit utilisé en conditions réelles dans l'ensemble d'un système de vidéosurveillance. Les solutions existantes se basent toujours sur d'autres types de capteurs qui détectent l'intrusion dans une zone interdite par exemple. Le besoin persiste donc de proposer des approches de filtrage, de modélisation, d'indexation et de recherche dans des collections de contenus vidéos sans avoir recours à une indexation exhaustive basée sur le contenu. L'idée principale est d'utiliser des informations provenant d'autres sources (e.g., données issues des capteurs, caractéristiques techniques, annotations sociales) pour générer des descripteurs ou des résumés. Ces informations peuvent être fixes ou peuvent évoluer avec le temps.

De ce constat naît le besoin de s'appuyer sur d'autres types d'informations, qui ressort aussi de la norme ISO 22311<sup>8</sup> publiée grâce aux efforts communs des principaux acteurs du domaine de la vidéosurveillance en France et au niveau international. La norme spécifie un format commun pour les données qui peuvent être extraites des systèmes de collecte de vidéosurveillance, par exemple à des fins d'enquête. Le dictionnaire de métadonnées proposé par la norme contient des éléments de localisation des caméras, description de la scène observée par chaque caméra, caractéristiques optiques de la caméra, etc.

Une généralisation de la mise en oeuvre de la norme, avec présence des métadonnées

---

8. [http://www.iso.org/iso/fr/catalogue\\_detail.htm?csnumber=53467](http://www.iso.org/iso/fr/catalogue_detail.htm?csnumber=53467)



normalisées, est envisagée grâce au développement des nouvelles technologies, qui ont rendu facile et peu onéreux le déploiement d'autres types de capteurs (e.g., capteurs GPS, boussoles, accéléromètres) associés aux caméras. Les travaux existants montrent qu'en se basant sur les métadonnées spatiales et les caractéristiques de la caméra, il est possible d'extraire des informations précieuses et précises concernant la scène filmée [Debnath and Borcea, 2013].

Dans le but final de "traduire" le problème de filtrage des contenus vidéos en un problème de modélisation et interrogation des données spatio-temporelles, nous allons présenter par la suite un état de l'art des travaux menés dans le domaine de la modélisation des informations liés aux objets ou capteurs fixes et mobiles. Nous allons utiliser la notion de contexte pour se référer à l'ensemble de ces informations. Nous allons utiliser le terme "objet" pour désigner aussi bien l'objet cible (e.g., personne, voiture) que le capteur (e.g., caméra, lecteur RFID, température) puisque la modélisation doit être uniforme pour les deux.

## 2.3 Modélisation des informations contextuelles

Un système comme celui que nous avons pris comme cible de notre étude est composé d'un ensemble de caméras fixes (qui peuvent quand même changer de champ de vue), de caméras mobiles (e.g., installées à l'intérieur ou à l'extérieur des bus), d'objets cibles (personnes, voitures, bâtiments) et de capteurs associés aux caméras ou aux objets. Les éléments de contexte liés à tous ces éléments que nous allons considérer sont :

- les informations liées à l'environnement (réseau routier ou réseau de transport) ;
- les informations de géolocalisation (la position et le changement de cette position dans le temps) ;
- les informations décrivant l'aire d'action du capteur vidéo (le champ de vue de la caméra) ;

Par la suite nous allons présenter la façon dont ces informations de contexte sont modélisées dans la littérature.

### 2.3.1 Modélisation de la localisation et de la trajectoire des objets mobiles

Une des plus importantes informations contextuelle et le point d'entrée de toute enquête est l'information spatio-temporelle liée aux objets. Avec l'apparition et le développement des outils géolocalisés (e.g., capteurs GPS, smartphones), il est devenu de plus en plus facile d'avoir des informations comme la position d'un objet mobile (e.g., personne, véhicule, bus). Nous considérons comme objet mobile toute entité capable ou munie des dispositifs capables de calculer et d'envoyer périodiquement sa position. Au moment où il faut stocker et gérer ce type de donnée à l'intérieur d'un système d'information, les premières questions qui se posent sont : Comment représenter la position d'un objet et sa modification dans le temps ? et quels sont les problèmes relatifs à l'acquisition de ces données ?

La représentation de la position d'un objet est de deux types [Leonhardt, 1998] :

- *position géométrique* : de la forme des coordonnées (x, y, z) par rapport à un système de référence localement/globalement défini (e.g., coordonnées GPS par rapport au système géodésique). Le principal avantage d'une telle position est la grande précision de l'information de localisation (la position est toujours réduite à un point). Le désavantage est représenté par le manque de sémantique d'une représentation purement géométrique qui n'est pas la façon dont les utilisateurs ont l'habitude d'exprimer leur position [Afyouni et al., 2012] ;
- *position symbolique* : de la forme d'une description de l'environnement en se basant sur des entités structurelles et/ou points d'intérêt (e.g., nom de la rue, du quartier, numéro de la rue, numéro de ligne de bus, station de bus, numéro de la pièce, étage, nom du bâtiment). La représentation symbolique de la position d'un objet vient adresser le problème de gap sémantique levé par la représentation géométrique mais a les inconvénients suivants : le besoin de définir un modèle précis de l'environnement par rapport auquel les symboles sont exprimés qui entraîne un temps de modélisation élevé, la précision de la position dépend du niveau d'affinage du modèle d'environnement sous-jacent (e.g., si on considère les localisations données par rapport à un système de lecteurs RFID la position est représentée par l'aire de couverture du lecteur qui est un cercle), le calcul des distances ne peut souvent être réalisé que de façon qualitative.

Nous nous intéressons à la représentation du mouvement des objets, ce qui est connu dans la littérature sous le nom de trajectoire de l'objet. Par la suite, nous allons présenter les différents types de trajectoires existants dans la littérature.

## Modélisation de la trajectoire des objets mobiles

Dans la littérature, on utilise souvent le terme de trajectoire de l'objet mobile pour se référer au changement de la position de l'objet. L'idée derrière le concept de trajectoire prend ses origines dans le besoin de capturer le mouvement d'un objet dans une certaine zone géographique pendant un certain temps. Le mouvement brut ou la trajectoire brute d'un objet est définie comme une séquence de positions spatio-temporelles (positions géométriques, plus précisément des coordonnées par rapport à un système de référence), c'est-à-dire une séquence de tuples (position, timestamp) [Parent et al., 2013]. En fonction du type d'objet, des types de capteurs associés à l'objet et de la nature de l'application cible, d'autres informations peuvent être associées à la trajectoire (e.g. vitesse, accélération pour les véhicules [Sandu-Popa and Zeitouni, 2012], orientation et champ de vue pour les caméras [Arslan Ay et al., 2010b]). Cette trajectoire brute peut ensuite être traitée pour en extraire plus de connaissances en fonction des applications envisagées. Par exemple, une trajectoire peut être segmentée en fonction de différents critères. Le segment d'une trajectoire est défini comme la sous-séquence maximale d'une trajectoire de façon que toutes les positions spatio-temporelles soient conformes à un prédicat donné (e.g., direction constante ou même moyen de transport) [Mountain and Raper, 2001].

Nous allons donner une définition formelle et générique d'une trajectoire et d'un segment de trajectoire. Une *trajectoire* (voir l'équation 2.1) est définie comme un tuple composé d'un identifiant de trajectoire, un identifiant d'objet, un ensemble de segments de trajectoires et un intervalle de temps global qui marque le "start" et le "end" de la trajectoire.

$$Tr = (tr_{id}, obj_{id}, \{u_k\}, [t_{start}, t_{end}]) \quad (2.1)$$

$$u_k = (u_{id}, \{position_i\} / 1 \leq i, [t_{kstart}, t_{kend}]) \quad (2.2)$$

La définition d'un segment de trajectoire est illustré par l'équation 2.2. Un segment de trajectoire est un tuple qui contient un identifiant de segment, un ensemble de positions homogènes et un intervalle de temps qui est optionnel ( $t_{kstart}$  et  $t_{kend}$  représentent le début et la fin de l'intervalle temporaire pour le segment k).

Beaucoup de travaux plaident pour le fait que la représentation purement géométrique des positions n'est pas appropriée pour l'utilisateur qui a l'habitude d'exprimer des localisations d'une façon plus sémantique (e.g. Rue Montes-

quieu no 15 ou station du bus numéro 2). Pour pouvoir fournir des services et des applications plus intéressantes à l'utilisateur il faut rajouter des informations du contexte aux trajectoires brutes [Spaccapietra et al., 2008], [Parent et al., 2013], [Spaccapietra and Parent, 2011]. En conséquence apparaît le processus d'enrichissement sémantique d'une trajectoire se basant sur des sources externes de données (e.g., bases de données géographiques, sites web), sur une annotation manuelle<sup>9</sup> ou sur des algorithmes d'apprentissage [Zheng et al., 2010]. L'enrichissement sémantique est réalisé par l'ajout d'attributs à certaines granularités de la trajectoire. Un exemple de types d'attribut peut être le moyen de transport. Si la trajectoire est stockée dans une base de données, ce type d'information peut être représenté comme une valeur textuelle, par exemple "bus numéro 2" rajoutée dans un champ *Commentaires* ou par un pointeur vers le champ d'une autre table si on a une table spécifique qui renseigne tous les moyens de transport, La trajectoire géométrique avec des annotations sémantiques devient donc une *trajectoire sémantique*. L'apparition de ce type de trajectoire montre le besoin d'avoir les deux types d'informations (géométrique et symbolique) dans le cadre de la même trajectoire.

Le mouvement des objets peut être effectué de façon libre (e.g., le déplacement de l'aire d'impact d'un ouragan) [Parent et al., 2006], [Guttting and Schneider, 2005] ou il peut être contraint par exemple par un réseau routier [Guttting et al., 2006] (e.g., le mouvement des voitures suit les rues d'une ville), par un réseau de transport [Booth et al., 2009] (e.g., le mouvement des bus suit des lignes prédéfinies) ou par l'architecture d'un bâtiment [Jensen et al., 2009] (e.g., le mouvement d'une personne à l'intérieur d'une station de métro est contraint par la disposition des différentes pièces et les connexions entre les pièces).

Si le mouvement de l'objet est réalisé dans un réseau routier ou dans un réseau de transport, les segments de trajectoires peuvent être projetés sur la topologie sous-jacente (voir Figure 2.1) par l'intermédiaire des algorithmes de map-matching [Brakatsoulas et al., 2005]. Le segment de trajectoire peut être défini comme l'évolution de la position de l'objet dans l'espace pendant un intervalle de temps [Spaccapietra et al., 2008], [Cao and Wolfson, 2005].

$$Tr : [t_{begin}, t_{end}] \rightarrow space \quad (2.3)$$

Par conséquent, dans la démarche de définition du mouvement d'un objet mobile, il faut prendre en compte la modélisation de l'environnement et la représentation

---

9. <http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/>

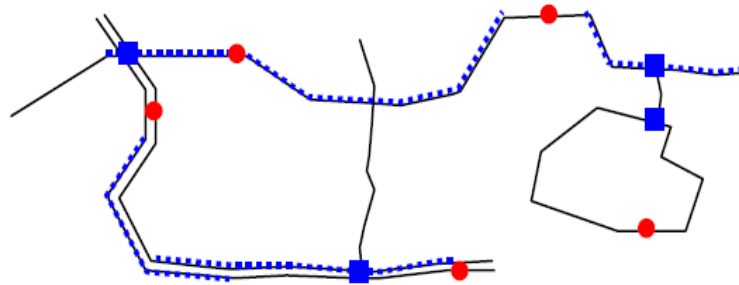


FIGURE 2.1 – Projection des segments de trajectoire (marqués en pointillés) sur un réseau sous-jacent [Guttting et al., 2006]

(géométrique ou symbolique) de la position et implicitement le représentation du mouvement de l'objet par rapport à cet environnement. Par la suite, nous allons passer en revue les principaux travaux de modélisation des données liées au réseaux routier et de transport en commun en environnement urbain.

### Modélisation du réseau routier

L'approche de modélisation du réseau routier la plus utilisée dans la littérature est celle basée sur la théorie des graphes [Guttting et al., 2006], [Frentzos, 2003], [Liu et al., 2012]. Un graphe est un tuple  $G=(V,E)$  composé d'un ensemble fini de noeuds  $V$  et un ensemble d'arêtes  $E \in V \times V$  qui constitue les liens entre les noeuds. Ces liens peuvent être orientés (si on prend en compte la direction) ou non-orientés. Le lien réflexif  $e=(v,v)$  s'appelle une boucle. Le réseau peut avoir plusieurs granularités car les noeuds du graphe peuvent être les changements de direction, les intersection des rues ou les points de début et de fin des rues [Sandu Popa et al., 2011]. Les arêtes du graphe peuvent se voir associés des poids en fonction de l'application envisagée (si on veut appliquer des algorithmes de calcul des chemins les plus courts, on peut associer des valeurs constantes représentant la distance [Pajor, 2009], si on veut développer une application de gestion du trafic on peut associer une fonction qui calcule la vitesse moyenne sur le segment de rue [Sandu-Popa and Zeitouni, 2012]).

Il y a des approches qui, en se basant principalement sur des techniques de traitement d'images, analysent l'intégralité du réseau routier cible et le stockent dans une base de donnée locale [Brinkhoff, 2002] (voir Figure 2.2). D'autres utilisent des bases de données extérieures comme celle de OpenStreetMaps [Shen et al., 2011]. Les réseaux routiers extérieurs peuvent être interrogés via des APIs publiques comme



FIGURE 2.2 – Exemples de graphes routiers correspondant à deux villes [Brinkhoff, 2002]

celle de OpenStreetMaps<sup>10</sup> ou de Google Maps<sup>11</sup>. Les services fournis par ces interfaces rendent possibles l'extraction des informations descriptives de la géométrie des segments des rues (les points GPS qui décrivent leur forme), l'extraction des informations relatives aux objets qui se trouvent dans une certaine aire (OpenStreet-Maps) et l'appellation des fonctions de geocode et reverse geocode<sup>12</sup> qui définissent un "mapping" entre les positions géométriques (coordonnées GPS) et les positions symboliques (exprimées en temps qu'adresses postales par rapport au réseau de rues).

Le mouvement des objets dans un environnement urbain est contraint aussi par le réseau de transport en commun. En conséquence, beaucoup de travaux se focalisent sur l'intégration de la modélisation du réseau routier et du réseau de transport en commun, sur la modélisation d'une trajectoire multi-modale (qui peut changer entre les deux) et sur la sécurité des moyens de transport en commun.

---

10. <http://wiki.openstreetmap.org/wiki/API>

11. <https://developers.google.com/maps/?hl=FR>

12. <https://developers.google.com/maps/documentation/geocoding/?hl=fr>

## Modélisation du réseau de transport en commun

Les réseaux de transport urbains deviennent de plus en plus complexes et avec l'exode de la population rurale, les villes deviennent très étendues et les gens passent jusqu'à plusieurs heures par jour dans les trajets urbains. De plus en plus de compagnies assurant le transport de personnes se préoccupent du confort et de la sécurité des passagers. En conséquence, le concept de système de transport intelligent (ITS- Intelligent Transport System) est de plus en plus utilisé. Il se réfère aux efforts faits en se basant sur les solutions technologiques actuelles (en termes de déploiement de réseaux de capteurs, collecte et modélisation de données, applications de types GIS, etc) pour améliorer la sécurité, le confort des passagers. La grande majorité de services proposés comprennent le planning et la recommandation des trajets pour les utilisateurs en fonction des différents critères (e.g., distance, coût, nombre de changements, incidents existant sur le trajet et même sûreté du trajet). Des solutions dans ce sens ont été proposées aussi bien par la communauté des chercheurs que par celle des commerciaux. [Zhang et al., 2011a], [Zhang et al., 2011b] et [Borole et al., 2013] proposent des applications de planification des trajets dans des réseaux multimodaux en Californie en tenant compte des informations comme les horaires des moyens des transport, des informations en temps réel et d'autres types d'informations comme les places de parking disponibles alentour. [Peng and Kim, 2008], [Beelen, 2004] se sont concentrés plutôt sur l'aspect architecture et ont proposé des approches distribuées pour l'assistance des passagers dans la planification de leur trajets. Des compagnies qui offrent des services de type GIS (Geographical Information Systems) ou des services de transport comme Google<sup>13</sup>, Trapeze<sup>14</sup>, Tisseo (Toulouse, France)<sup>15</sup>, Bahn (Germany)<sup>16</sup> ont développé leur propres produits.

Ces services sont basés sur des solutions de modélisation du réseau de transport qui doivent d'un côté intégrer toutes les informations concernant les contraintes spatiales (les lignes de transport en commun suivent des trajets prédéfinis passant par certaines stations) et temporelles (e.g., les horaires des différents bus en fonction de la date (e.g., les fréquences des bus diffèrent parfois pendant les périodes de vacances et en fonction du moment du jour) en répondant aux exigences de flexibilité, facilité de modification et rapidité en temps de calcul de requête.

La notion centrale dans la modélisation d'un réseau de transport est celle d'horaire

---

13. <http://maps.google.com/intl/fr/landing/transit/#dmy>

14. <http://www.trapezegroup.eu/solutions/public-transport/planning-and-analysis>

15. <http://www.tisseo.fr/calculateur-multimodal>

16. <http://en.swb-busundbahn.de/>

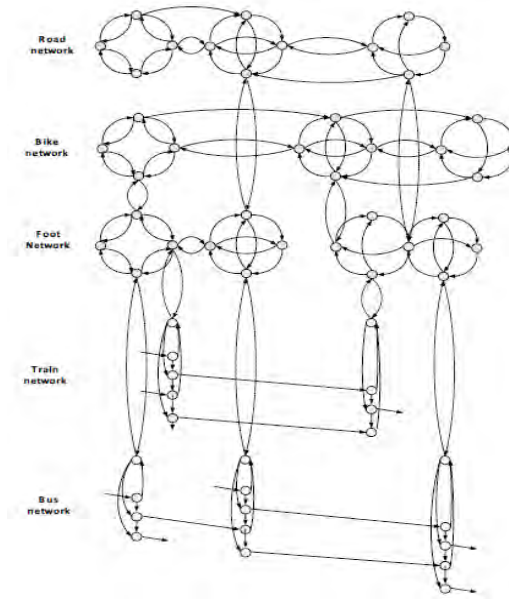


FIGURE 2.3 – Schéma du super-réseau proposé par [Zhang et al., 2011b]

de passage des bus, trains, etc. dans les stations. Par exemple, dans le cas d'un réseau de trains, un horaire est défini comme  $(C, B, Z, \Pi)$  où  $B$  est un ensemble de stations,  $Z$  un ensemble de trains,  $\Pi$  la périodicité de l'horaire et  $C$  un ensemble de connexions élémentaires. Chaque connexion élémentaire de  $C$  est définie comme le tuple  $c := (z, S_1, S_2, t_1, t_2)$  qui désigne le trajet du train  $z \in Z$  allant de la station  $S_1$  jusqu'à la station  $S_2$  entre  $t_1$  et  $t_2$  [Pajor, 2009]. Cette notion d'horaire est implémentée dans des modèles basés sur les graphes de deux façons :

- en modélisant pour chaque station (qui va être considérée comme super-noeud), autant de noeuds (appelés noeuds-événements) qu'il y a d'arrêts et de départs de la station respective; [Zhang et al., 2011b] utilisent cette approche et proposent une solution de planification des itinéraires dans un soit-disant super-réseau (qui englobe plusieurs modes de transport) comme celui illustré dans la Figure 2.3.
- en modélisant un seul noeud par station en associant une propriété (une fonction) aux liens entre les noeuds qui donne pour chaque point de départ la durée du voyage; [Booth et al., 2009] proposent une modélisation d'un réseau de transport en commun urbain ayant comme but le calcul des chemins possibles entre un point de départ et un point d'arrêt. Ils définissent le réseau comme un tuple  $U = (M, F, L, G)$ , où  $M$  est l'ensemble des modes de transport  $M = \{\text{pedestrian, auto, bus, urban rail, suburban rail}\}$ ,  $F$  un ensemble d'établissements (facilities) (e.g., centres



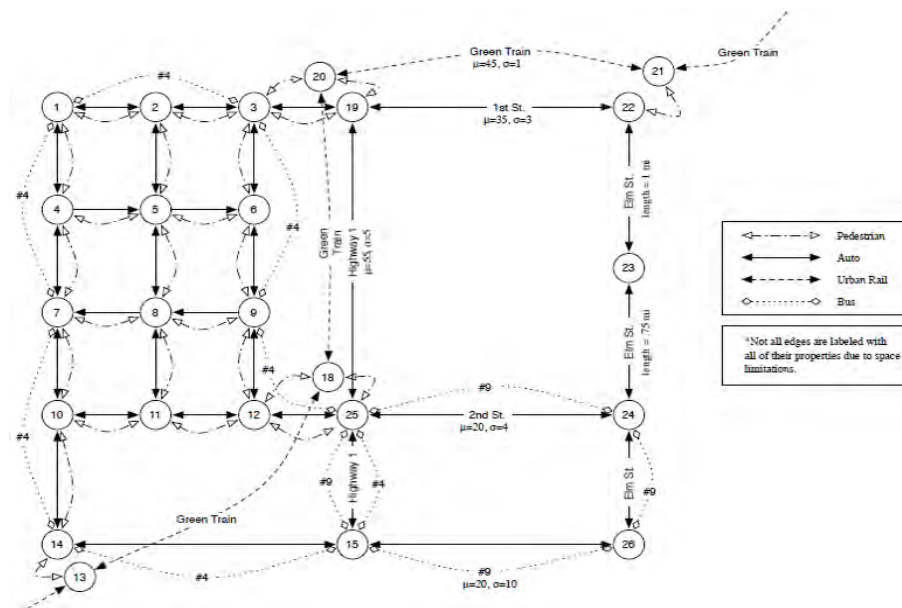


FIGURE 2.4 – Schéma du graphe proposé par [Booth et al., 2009]

commerciaux, stations d'essence, théâtres),  $L$  un ensemble d'attributs associé aux arêtes du graphe et  $G$  un multigraphe orienté. Comme on peut voir sur la figure 2.4, pratiquement on a un graphe pour chaque mode. Ces graphes sont unifiés dans un seul car ils partagent des nœuds. Les nœuds partagés sont ceux où on peut changer de mode.

[Pajor, 2009] utilise également cette approche pour développer une solution qui calcule des trajets dans un réseau multimodal composé d'un réseau routier, un réseau de trains et un réseau de vols.

Pour les deux types de modèles, les temps de transfert dans chaque station peuvent être pris en compte.

En se basant sur ces modèles de réseau, des algorithmes de calcul des chemins dans les graphes comme l'algorithme de Dijkstra [Dijkstra, 1959] ou l'algorithme Dynet [Tulp and Siklóssy, 1991] ont été modifiés pour "paramétrer" ou optimiser la recherche en fonction des critères demandés par l'application envisagée (e.g., temps, distance, nombre de moyens de transport changés, prise en compte ou pas des horaires).

Vue la multitude d'approches de modélisation des réseaux de transport dans le but de fournir les applications mentionnées, un autre aspect crucial qui a été pris en

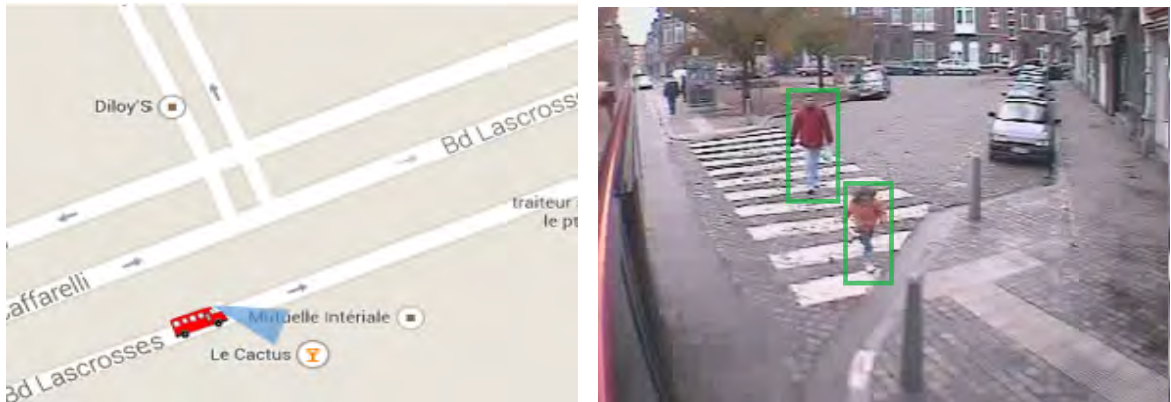


FIGURE 2.5 – Caméra installée à l'extérieur d'un bus

compte dans la démarche de développement des systèmes de transport intelligent est l'interopérabilité des différents modèles et systèmes proposés. Par conséquent, des ontologies [Houda et al., 2010] et des formats de descriptions des données liées aux réseaux de transport (e.g., le format de Google GTFS (General Transit Feed Specification)<sup>17</sup>) ont été proposés. [Wong, 2013] présentent la grande utilisation de GTFS au niveau mondial et ils utilisent le terme de "standard" pour s'y référer.

Assurer le confort des passagers concerne forcément leur sécurité. Comme nous l'avons déjà montré dans la première section de cet état de l'art, il y a des projets qui concernent l'équipement des moyens de transport en commun avec des composantes matérielles (e.g., capteurs, caméras) et logicielles (e.g., algorithmes de filtrage et d'analyse des contenus vidéos) afin d'assurer la sécurité des personnes [Amriki and Atrey, 2014], le projet SURTRAIN<sup>18</sup>. Des caméras de vidéosurveillance sont installées aussi bien à l'intérieur qu'à l'extérieur des bus. La Figure 2.5 montre une image capturée par une caméra installée à l'extérieur d'un bus.

Afin de pouvoir tester et valider les modèles présentés, le besoin est d'avoir des collections de données publiques et représentatives pour les problématiques considérées ce qui ne représente pas une tâche facile. Par la suite, nous allons nous intéresser à la question relative à l'acquisition des données de localisation des objets mobiles : par quel moyen et quels sont les problèmes rencontrés ?

17. <https://developers.google.com/transit/gtfs/reference?hl=fr-FR>

18. <http://www.agence-nationale-recherche.fr/informations/l-actu-en-video/surtrain-surveillance-des-transport-par-lanalyse-de-limage-et-du-son/>

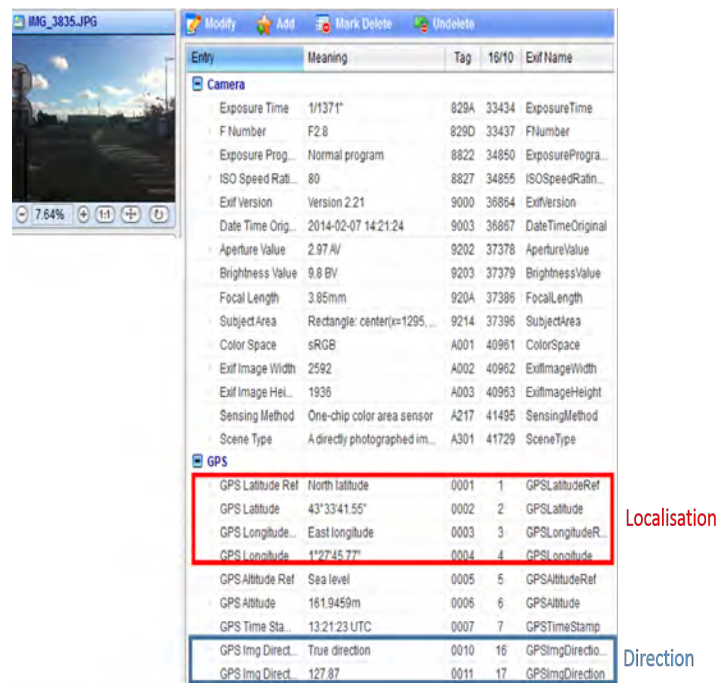


FIGURE 2.6 – Métadonnées spatiotemporelles associées à une image par un Iphone4

### Acquisition des données de localisation

Plusieurs solutions de générateurs de données de localisation des objets mobiles qui se déplacent dans des réseaux routiers urbains ont été proposées dans la littérature ([Saglio and Moreira, 2000], [Pfoser and Theodoridis, 2003]), le plus connu étant celui de [Brinkhoff, 2002]. Ce générateur est très utile pour tester des algorithmes et des structures d'index qui supportent une large palette de requêtes spatiales mais a les désavantages que les données générées sont assez limitées et le manque d'information de localisation réelle le rend très difficile à utiliser dans notre contexte.

Avec les technologies actuelles, les informations spatiotemporelles sont de plus en plus faciles à collecter. Dans l'espace outdoor : presque tous les smartphones et caméras de nouvelle génération sont dotés d'un capteur GPS, voire même d'une boussole. Leur position peut être identifiée de façon très précise en utilisant le GPS, les antennes WiFi ou les cellules GSM [Zandbergen, 2009], [Raimond et al., 2012]. La figure 2.6 montre un extrait des métadonnées associées à une image prise avec un Iphone4. Nous pouvons observer les coordonnées GPS et l'angle d'orientation de la caméra par rapport à la direction Nord.

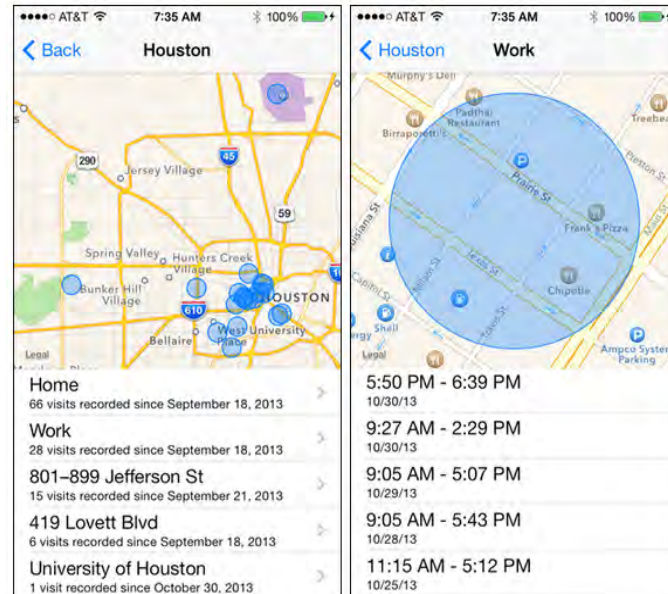


FIGURE 2.7 – Localisations fréquentes enregistrées par un iPhone4S

A partir de IOS7, les dispositifs iPhone permettent (si souhaité par l'utilisateur) de garder la trace des localisations de l'utilisateur (voir Figures 2.7 et 2.9<sup>19</sup>).

Récemment, Google a mis à disposition un site<sup>20</sup> où les utilisateurs peuvent se connecter avec leur compte et peuvent retrouver (s'il le souhaitent en choisissant de garder activé le service localisation sur leur dispositif mobile) l'historique de leur déplacements (Figure 2.8). Cet historique est construit à partir des localisations détectées par les capteurs GPS embarqués dans leur smartphone ou tout autre dispositif mobile.

Dans l'environnement "indoor", les choses sont plus compliquées car la précision des coordonnées GPS est très limitée. Beaucoup de travaux ont été menés pour essayer de trouver des solutions au problème de la localisation indoor. Les technologies utilisées varient beaucoup, des technologies radio (e.g., RFID, WLAN, Bluetooth) ou non (e.g., infrarouge et ultrason) [Kolodziej and Hjelm, 2006]. Les travaux qui ont essayé de proposer des systèmes de localisation uniformes (qui puissent intégrer différents types de modèles de localisation) sont peu nombreux et n'ont été implémentés

19. <http://blog.chron.com/techblog/2013/10/your-iphone-knows-where-youve-been-puts-it-on-a-map/>

20. <https://maps.google.com/locationhistory/>

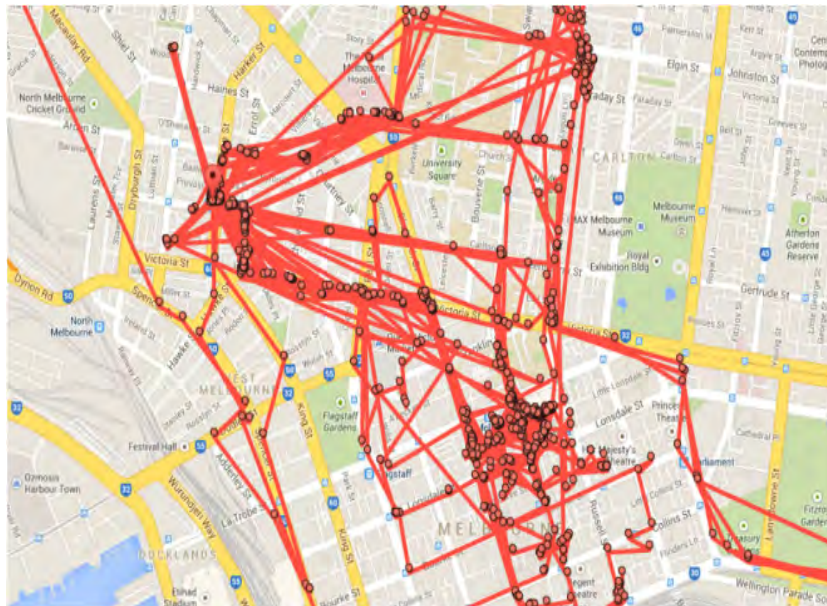


FIGURE 2.8 – Exemple d'historique Google de localisations

seulement dans des environnements restreints et contrôlés [Hansen et al., 2009]. La problématique de la modélisation des localisations des objets mobiles dans un environnement indoor est détaillée dans l'annexe A.

Comme expliqué avant, en général, en environnement urbain outdoor, la mobilité des objets (e.g., personnes, véhicules, bus) est contrainte par les réseaux routiers et de transport en commun. Avec le développement des différents outils disponibles en ligne comme les API de Google Maps<sup>21</sup>, de OpenStreetMaps<sup>22</sup> ou Bing Maps<sup>23</sup> il est facile de disposer des données concernant les réseaux routiers urbains et d'autres objets d'intérêt qui peuvent se trouver dans une ville comme les bâtiments, les parcs, les banques, etc.

Le terme de "open data" désigne les données rendues publiques dans un format lisible par un ordinateur qui permettent aux données d'être complètement accessibles et utilisables par les utilisateurs finaux<sup>24</sup>. Ce mouvement a gagné beaucoup de terrain ces dernières années parmi les agences fournissant des services de transport en

---

21. <https://developers.google.com/maps/?hl=FR>

22. <http://owl.apis.dev.openstreetmap.org/#map=5/51.500/-0.100>

23. <http://www.microsoft.com/maps/choose-your-bing-maps-API.aspx>

24. <http://opendatafrance.net/>



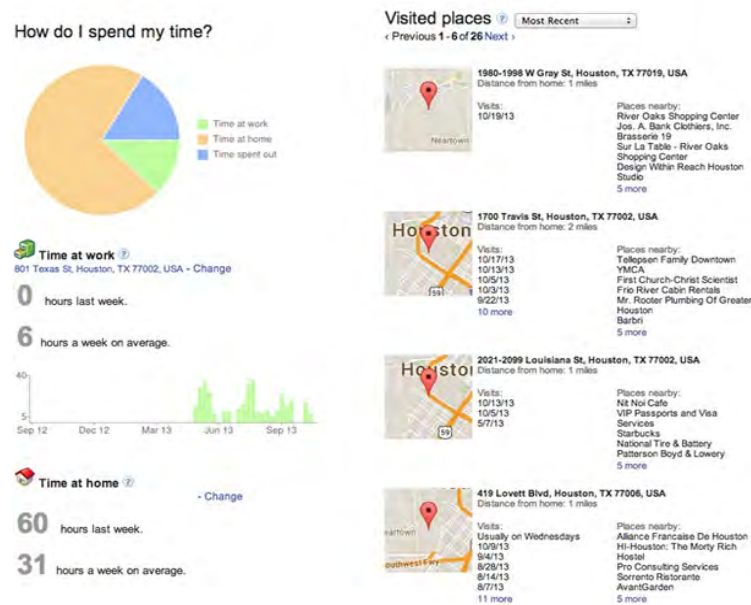


FIGURE 2.9 – Statistiques sur les localisations enregistrées par un Iphone4S pendant un mois

commun dans les villes. En conséquence, en suivant le trend Open Data, sont mises à disposition des données décrivant le réseau de transport (stations, horaires) et même des APIs qui permettent l'interrogation des horaires en temps réel. Nous allons citer l'exemple du projet Toulouse Open Data<sup>25</sup> qui met à disposition des utilisateurs des collections de données et des APIs qui leur permettent de récupérer des données offline et en temps réel concernant le réseau des bus et du métro toulousain avec les horaires de fonctionnement et les positions des bus en temps réel (la collection mise à disposition est basée sur le format d'échange des données liées au réseau de transport urbain GTFS publié par Google (voir Annexe B)).

Par la suite, nous allons présenter le troisième type de données contextuelles qui se réfèrent à la modélisation des informations décrivant l'aire d'action d'un capteur, dans notre cas d'un capteur vidéo.

25. <http://data.toulouse-metropole.fr/>

## 2.3.2 Modélisation des informations du contexte décrivant le champ de vue d'une caméra

### Représentation des données

Une caméra située à une position  $P$  et orientée dans une direction  $\vec{d}$  observe une aire qui est appelée champ de vue, scène observée ou scène visible et qui dépend de l'installation et des propriétés optiques de la caméra. [Arslan Ay et al., 2008] ont proposé de modéliser la scène visible 2D par une caméra à partir de quatre éléments :

- Position  $P$  donnée par un capteur GPS ;
- Direction de la caméra  $\vec{d}$  donnée par une boussole ;
- Angle de vue  $\theta$  qui est calculé à partir des caractéristiques optiques de la caméra (distance focale de la lentille, taille du capteur et niveau de zoom [Graham et al., 1965]) ;
- Distance visible  $R$  qui est la distance maximale à laquelle un objet est reconnu. Cette distance peut être calculée en se basant sur le contenu de l'image et dépend de l'application (e.g., pour une reconnaissance du visage, la distance est plus petite que pour une reconnaissance de silhouette).

En conclusion, le champ de vue d'une caméra à un instant de temps  $t$  peut être représenté comme un tuple de 4 éléments illustré par l'équation 2.4 :

$$FOVScene(P, \vec{d}, \theta, R) \quad (2.4)$$

En fonction des informations disponibles, le champ de vue peut être réduit à un point  $PointScene(P)$ , un cercle de centre  $P$  et de rayon  $R$   $CircleScene(P, R)$  ou une section d'un cercle comme dans la figure 2.10. Une analyse de toutes les caractéristiques des caméras qui influencent son champ de vue (horizontal et vertical) est présentée dans le chapitre 3 de ce manuscrit.

Ensuite nous allons aborder un problème important, celui de l'acquisition de ces données.

### Acquisition des données liées au champ de vue d'une caméra

Collecter des données relatives au changement du champ de vue d'une caméra dans le temps n'est pas une tâche facile car les données proviennent de capteurs différents

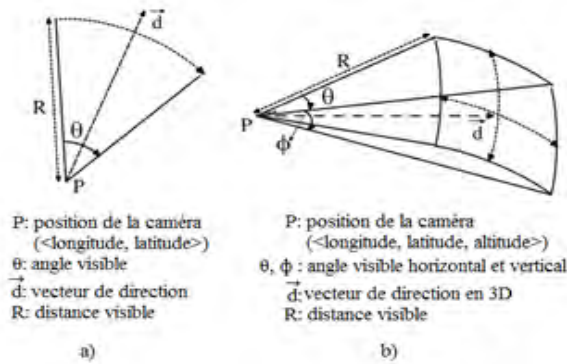


FIGURE 2.10 – Champ de vue 2D et 3D d'une caméra modélisés dans [Arslan Ay et al., 2008]

engendrant des problèmes de synchronisation et de gestion des erreurs. Et si on considère en outre le réseau de transport, le problème de trouver un dataset est impossible à résoudre.

Les rares approches qui ont traité ce problème ont créé leur propre dataset en associant un capteur GPS et une boussole à une caméra et en collectant simultanément des données provenant de ces capteurs [Arslan Ay et al., 2008] ou en utilisant les capteurs intégrés dans les smartphones [Debnath and Borcea, 2013].

[Arslan Ay et al., 2010a] sont partis des observations faites à partir des données réelles qu'ils ont générées en utilisant le prototype présenté dans [Arslan Ay et al., 2008] pour proposer un générateur paramétrable du mouvement et de la rotation des caméras. Ils définissent un *Camera Template* qui représente un schéma des caractéristiques qui influencent le mouvement de la caméra. En partant de l'idée que le comportement d'une caméra est influencé par le but et le contexte de l'enregistrement vidéo, les auteurs proposent un "template" contenant des caractéristiques paramétrables par l'utilisateur et qui va constituer l'entrée du générateur. Les éléments qui influencent le comportement d'une caméra sont :

- (1) La trajectoire : trois cas sont considérés :
  - (a) la caméra se déplace dans un réseau routier et les paramètres qui influence son mouvement sont : le réseau, la vitesse moyenne sur chaque type de rue (e.g, boulevard, rue secondaire), une accélération, une décélération, la probabilité d'un arrêt, le temps estimé d'un arrêt ;
  - (b) la caméra se déplace librement et les paramètres qui influencent son mouvement sont : une vitesse moyenne et une distribution initiale des caméras ;



- (c) la caméra a un mouvement mixte qui est influencé par : une vitesse moyenne, une probabilité de , une durée moyenne d'un arrêt et la distribution et le pourcentage du mouvement libre ;
- (2) La rotation.

Nous avons traité l'aspect modélisation des données spatio-temporelles. Nous allons dans ce qui suit présenter les types de requêtes qui peuvent être traités en s'appuyant sur les modèles mentionnés.

## 2.4 Types de requêtes supportés par les systèmes utilisant des métadonnées spatio-temporelles

Comme présenté dans l'introduction de ce chapitre, les applications s'appuyant sur des modèles de données de localisation et des informations issues de capteurs sont très diverses. Dans cette section nous présentons les types de requêtes qui sont supportés par ces modèles.

Nous considérons comme requête spatio-temporelle toute interrogation de l'utilisateur comportant au moins un élément de géolocalisation ou de situation spatiale (exprimée en termes de coordonnées numériques ou de noms symboliques) et/ou une référence temporelle (date, heure, intervalle) [Ilarri et al., 2010]. Les algorithmes derrière les requêtes spatio-temporelles sont basés sur les prédicats spatio-temporels décrits dans [Erwig and Schneider, 2002a].

Avec l'apparition des approches qui considèrent le mouvement des objets dans le cadre des réseaux routiers et de transport en commun, des solutions de traitement et d'indexation des données spatiales adaptées ont été proposées (la nouvelle sous-branche du domaine des bases de données spatiales est référée dans la littérature comme SNDB (Spatial Network Databases [Papadias et al., 2003]). En conséquence, chaque requête spatiale considérée dans l'espace euclidien à une correspondance dans l'espace d'un réseau routier ou de transport.

### 2.4.1 Position queries (P)

Les requêtes de type position retournent la localisation des objets statiques et mobiles, localisation qui est calculée en se basant sur un modèle géométrique ou symbo-

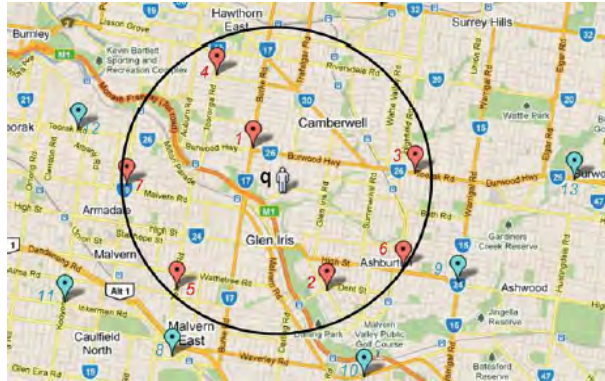


FIGURE 2.11 – Illustration d’une requête de type région [AL-Khalidi et al., 2013]

lique de localisation. Ce type de requête est à la base de toute autre requête spatio-temporelle qui utilise l’information de localisation des objets [Afyouni et al., 2012].

## 2.4.2 Range queries (R)

Les requêtes de type région sont des plus populaires dans les applications basées sur la localisation (LBS- Location Based Systems) et elles répondent à des questions comme : ”Quels sont les restaurants ou les stations d’essence situées dans un rayon de 500m de ma position” ou ”Quelles sont les voitures qui sont passées par ce périmètre” ou ”Quelles sont les voitures situées dans un rayon de 1km d’une voiture de police en mouvement”. Ce type de requêtes retourne des informations concernant les objets situés dans une certaine aire ou périmètre défini par l’utilisateur (d’habitude cette aire est définie par rapport à un objet de référence par exemple le cercle de rayon 1km ayant le centre dans la position de l’objet). Étant donnée une géométrie requête  $R$ , la requête est traduite par une fonction qui calcule l’ensemble d’objets  $O$  qui intersectent la géométrie au moment  $t$  (un exemple de géométrie de requête est illustré dans la Figure 2.11). Comme montré par les exemples, les requêtes de type région sont traitées dans plusieurs cas selon la mobilité de la requête et des objets cibles [Stojanovic et al., 2005] : région requête statique, objets cibles statiques ou mobiles et région mobile, objets cibles statiques ou mobiles.

Les travaux concernant ce type de requête se concentrent surtout sur la proposition des méthodes optimales de traitement de la requête et des techniques et des structures d’indexation [Papadias et al., 2003]. Traiter une requête spatiale de façon naïve implique le parcours séquentiel de la collection d’objets cibles et l’évaluation de

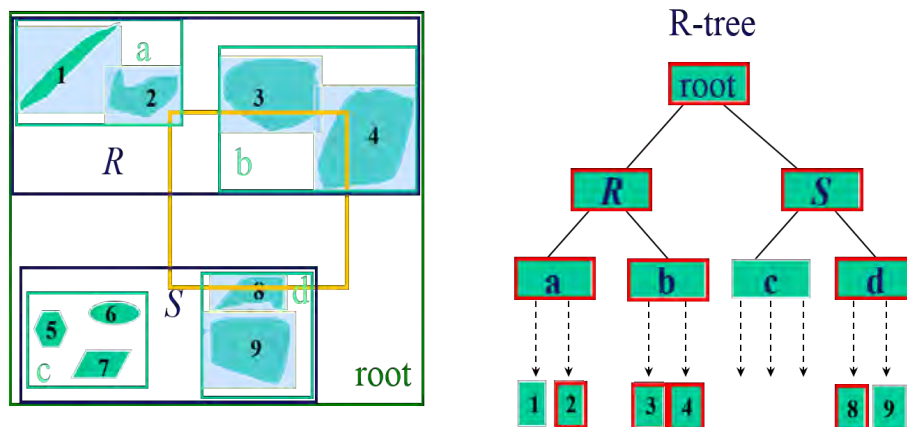


FIGURE 2.12 – Exemple d’une structure d’un index de type R-tree

l’intersection entre la géométrie de la requête et la géométrie de chaque objet. Dans ce cas, le temps d’exécution peut exploser surtout quand la région de la requête ou les objets cibles sont mobiles. En conséquence, la communauté de recherche s’est beaucoup concentrée sur le développement de solutions d’optimisation.

La technique d’optimisation la plus utilisée dans le traitement de requêtes spatiales est basée sur l’utilisation des MBRs (Minimum Bounding Rectangle) structurés dans un arbre d’indexation appelé R-tree (illustré dans la Figure 2.12). Le traitement d’une requête spatiale en se basant sur un tel index est réalisé en deux étapes : une première étape de filtrage suivie d’une étape d’affinage. Dans l’étape de filtrage, le R-tree est parcouru à partir de la racine et à chaque niveau les MBRs qui intersectent la région de la requête sont sélectionnés. Le résultat de cette étape est constitué par un ensemble de feuilles du R-tree (Figure 2.12). Ces feuilles contiennent aussi des pointeurs vers les géométries proprement dites des objets qui seront utilisées dans la deuxième étape pour réaliser une comparaison précise avec la région requête et un affinage de la liste des objets résultats.

[Papadias et al., 2003] considère les requête de type région à l’intérieur d’un réseau routier. Dans ce cas, les objets retournés sont ceux qui se trouvent dans la région donnée et en plus qui intersectent les segments du réseau routier. L’algorithme se base sur des structures d’indexations indépendantes pour le réseau et chaque ensemble d’objets cibles (e.g., hotels, stations d’essence).

A cause de la complexité des requêtes de type région et du temps de réponse qui peut être très grand, la solution de l’approximation a été exploitée

[AL-Khalidi et al., 2013]

### 2.4.3 Visibility queries (V)

Les requêtes de visibilité rajoutent la notion de direction et de champ de vue aux requêtes de type région. Pour une position et une modélisation du champ de vue données, les objets "vus" sont retournés. [Simon and Fröhlich, 2007] proposent une modélisation 2,5D de l'environnement outdoor d'un contexte urbain qui considère la position, la géométrie et la hauteur des objets qui composent l'environnement (e.g., bâtiments). Le champ visuel de l'objet "observateur" est calculé en fonction de la position (latitude, longitude, altitude) et l'orientation de l'observateur. En se basant sur les deux modèles les objets "vus" par l'observateur sont retournés.

### 2.4.4 Nearest neighbour queries (NN)

Les requêtes NN retournent l'objet le plus proche géographiquement par rapport à un objet donné par l'utilisateur. Étant donné un espace métrique  $(\mathbf{R}^n, d)$ , un ensemble de points candidats  $P$  et un ensemble de points requêtes  $Q$ , la requête est traduite par une fonction  $f : Q \rightarrow P$  qui pour chaque point de  $Q$  cherche dans  $P$  le point le plus proche par rapport à la distance  $d$  [Pajor, 2009] (la formalisation du problème de recherche du voisin le plus proche est donné par l'équation 2.5). L'algorithme naïf de résolution de ce problème consiste dans le calcul, pour chaque point requête, de la distance par rapport à chaque point candidat puis du choix de celui situé à la distance minimale. Des approches d'optimisation utilisent des index pour structurer l'ensemble des points candidats [Moore, 1991]. D'autres approches ont proposé des algorithmes optimisés pour la recherche des  $k$  voisins les plus proches (kNN query), où  $k$  est paramétrable [Benetis et al., 2006].

$$f(q) := p \Leftrightarrow \forall p' \in P : p \neq p' \Rightarrow d(p', q) \geq d(p, q) \quad (2.5)$$

Les approches concernant ce type de requête se déclinent en fonction de la mobilité des objets requêtes et cibles et à l'espace métrique de calcul des distances. La plupart des travaux considèrent l'espace métrique euclidien (2D). Dans cet espace, des approches considèrent que des objets statiques (requêtes et cibles) [Cheung and Fu, 1998], d'autres des objets requêtes mobiles [Song and Roussopoulos, 2001], [Feng et al., 2007]. Des algorithmes de recherche

considérant la double mobilité des points requêtes et cibles ont été proposés [Benetis et al., 2006]. Des propositions d'algorithmes qui considèrent la distance calculée par rapport au graphe d'un réseau routier ont également été faites [Jensen et al., 2003].

### 2.4.5 Nearest Surrounder Queries(NS)

Similaire à la démarche derrière les requêtes de visibilité, [Lee et al., 2010] rajoutent une couche de précision aux requêtes de type NN en prenant en compte la direction dans laquelle les objets résultat se trouvent par rapport à l'objet requête. Plus précisément, à partir de la position de l'objet requête, la liste des objets situés dans le voisinage le plus proche de l'objet requête (i.e., il n'existe pas d'autres objets entre l'objet requête et les objets du résultat) est générée avec les angles entre lesquels chaque objet du résultat est le NN de l'objet requête.

### 2.4.6 Trajectory queries(T)

Les requêtes de type trajectoire ne considèrent pas la position ou la région autour d'un objet à un moment donné mais le suivi (spatial) du mouvement de l'objet dans le temps. Nous avons expliqué en détail le concept de trajectoire dans la section 2.3.1 de ce chapitre. [Tao et al., 2002] considèrent le mouvement d'un objet requête comme un segment reliant tous les points de son mouvement et recherchent le voisin le plus proche de chaque point du segment. Étant donné un espace métrique  $(\mathbf{R}^n, d)$ , un ensemble de segments requêtes  $Q$ , chaque  $q \in Q$  est un segment  $= [s, e]$ . Le résultat de la requête est un ensemble de  $\langle R, T \rangle$ , où  $R$  est un point résultat et  $T$  un intervalle spatial du segment requête pour lequel  $R$  est le voisin le plus proche [Tao et al., 2002].

### 2.4.7 Predictive queries

Les types de requêtes présentés concernent les positions passées ou en temps réelles des objets requêtes et cibles. Il y a beaucoup de travaux qui se sont concentrées sur l'analyse des mouvements passés des objets afin de prédire et de répondre aux requêtes concernant les positions et trajectoires futures des objets([Hendawi and Mokbel, 2012],[Zheng et al., 2009]).

## 2.5 Bases de données spatio-temporelles

Jusqu'à maintenant dans ce chapitre nous avons passé en revue les principales approches définies dans la littérature pour la modélisation des données contextuelles liées au mouvement des objets et des capteurs mobiles et à l'aire d'action d'un capteur vidéo. Nous allons nous intéresser par la suite de l'implémentation de ces modèles dans des bases de données relationnelles donnant naissance aux bases de données spatio-temporelles.

L'implémentation des approches de modélisation présentées des informations spatio-temporelles associées aux objets mobiles dans des bases de données est un domaine qui a été beaucoup exploité et des propositions ont été faites pour :

- des modèles conceptuels et logiques développés au dessus du modèle relationnel [Parent et al., 2006],
- des nouveaux types de données (les plus importants étant (*point*, *line*, *region*) avec leur équivalents mobiles et leur dérivés pour le mouvement contraint d'un réseau routier) [Forlizzi et al., 2000], [Gutting and Schneider, 2005], [Parent et al., 2006],
- des nouveaux prédicats topologiques (Meets, Adjacent, Touches, Crosses, Overlaps, etc.) [Erwig and Schneider, 2002a],
- des nouveaux langages d'interrogation basé sur SQL [Erwig and Schneider, 2002b]

Des extensions des systèmes de gestion des bases de données commerciaux pour gérer les objets spatiaux ont atteint déjà leur maturité et sont utilisés à large échelle : Oracle Spatial<sup>26</sup>, PostGIS<sup>27</sup>, Extension Spatiale de MySQL<sup>28</sup>.

Il y a des travaux qui ont proposé des extension temporelles aux bases de données spatiales en implémentant des types de données spatio-temporels. Par exemple, dans [Zhao et al., 2011], les auteurs présentent une extension spatiotemporelle nommée STOC (un package PL/SQL) de Oracle Spatial. L'extension contient des types de données (voir tableau 2.1) et des opérations.

L'extension proposée dans [Zhao et al., 2011], permet de répondre à différents types de requêtes spatio-temporelles : (1) Spatio-Temporal Range Query (e.g., Quelles sont les voitures qui ont traversé la région X dans la période de temps Y?), (2) Spatio-temporal Distance Query (e.g., Quelles sont les paires de voitures qui n'ont pas été

---

26. <http://www.oracle.com/fr/products/database/options/spatial/index.html>

27. <http://postgis.net/>

28. <http://dev.mysql.com/doc/refman/5.6/en/spatial-extensions.html>

Type de données	Représentation interne
Moving(Bool), Moving(String) Moving(Number)	<value, [from, to]> <t_type, num, chR, [from,to] >, où t_type est 0 pour des données discrètes et 1 pour des données continues, num est la valeur au début de l'intervalle et chR est le taux de changement de num
Moving(Point) Moving(Region)	<point, crX, cRY, [from, to]> < region (SDO_GEOMETRY), crX1, cRY1, crX2, cRY2, [from, to] >

TABLE 2.1 – Résumé des implémentations des types spatiotemporels dans [Zhao et al., 2011]

à plus de 10m l'une de l'autre?), (3) Spatio-temporal Topology Query (e.g., Quelles sont les paires de voitures qui se sont croisées?), (4) Spatio-Temporal Aggregate Query (e.g., Quels carrefours ont été visités par le plus grand nombre de voitures?). Cette approche ne prend pas en compte le réseau routier ou celui de transport en commun. De plus, pour exécuter des requêtes elles utilisent des opérateurs développés et qui ne sont pas détaillés.

Ces bases de données spatio-temporelles représentent la structure sous-jacente pour beaucoup d'applications qui visent des objectifs différents. Nous allons présenter une comparaison des certaines de ces applications avec un focus sur celles qui traitent la problématique de la recherche du contenu vidéo.

## 2.6 Comparaison des approches

Plusieurs approches ont proposé d'utiliser des informations spatiotemporelles liées aux contenus vidéo ou aux objets dans le cadre des systèmes d'informations. Ces différentes approches se différencient par :

- l'objectif principal de l'application (e.g. : annotation des vidéos et des images avec des tags textuels, développement des systèmes d'aide à la décision basée sur l'interrogation des informations géospatiales, développement des systèmes de gestion du trafic);

- les métadonnées sur lesquelles ces systèmes se basent : position des objets, géométrie de la scène observée, temps, caractéristiques techniques de la caméra ;
- le type de positions traité (géométrique, symbolique) ;
- la façon dont ces informations sont représentées dans chaque modèle de données : les données sont continues/discrètes, le champ de vue de la caméra est représenté comme une région mobile (une géométrie) calculée pour chaque frame/minute ou seconde, etc. ;
- le(s) type(s) de requête auxquels le système permet de répondre (e.g., key words(KW), range based(R), nearest neighbor(NN), visibility(V), trajectory(T)).

Par la suite, nous allons présenter des approches qui utilisent des données spatio-temporelles pour des applications diverses en mettant en évidence les critères de comparaison que nous venons de préciser. Un tableau synthétique est présenté en fin de ce chapitre.

Dans [Zhao et al., 2011], les auteurs présentent une extension spatiotemporelle nommée STOC (un package PL/SQL) de Oracle Spatial. Les régions mobiles sont représentées comme des géométries (SDO\_GEOMETRY<sup>29</sup>) qui bougent dans le temps. Le cas d'utilisation présenté est un système de gestion des informations concernant le trafic et qui permet de répondre à des questions comme : "Quels sont les véhicules qui ont traversé une région donnée ?".

Dans [Liu et al., 2009], les auteurs proposent un système (SEVA) qui annote chaque frame d'une vidéo par la localisation, le timestamp et les objets présents dans le frame. Le système est composé de : (1) une caméra vidéo, (2) une boussole numérique, (3) un système de localisation, (4) une radio wi-fi associée à la caméra. Les auteurs partent de l'hypothèse que tous les objets susceptibles d'être capturés sur les vidéos sont dotés d'un système qui leur permet de transmettre leur localisation (qui sera captée par la radio wi-fi). A partir de la localisation des caméras et de la localisation des objets, les images (frames de la vidéo) sont annotées par les objets qu'elles sont susceptibles de contenir. Des opérations d'interpolation, extrapolation, synchronisation temporelle et filtrage basées sur le champ de vue de la caméra sont réalisées pour affiner les annotations. Les auteurs partent de l'hypothèse forte que tous les objets sont munis d'un capteur qui permet au système d'avoir sa localisation à chaque moment, ce qui n'existe que dans des environnements contrôlés. Ils construisent également la géométrie du champ de vue pour chaque seconde de vidéo.

Dans [Shen et al., 2011], une approche similaire à SEVA est présentée, avec les différences suivantes : (1) les objets ne doivent pas transmettre leur position et (2)

---

29. [http://docs.oracle.com/cd/B19306\\_01/appdev.102/b14255/sdo\\_objrelschem.htm](http://docs.oracle.com/cd/B19306_01/appdev.102/b14255/sdo_objrelschem.htm)



leur géométrie est prise en compte, et pas uniquement le point de localisation. Pour chaque seconde de la vidéo, les auteurs calculent le champ de vue associé à la caméra et interrogent deux bases de données extérieures (OpenStreetMaps et GeoDec) afin d'extraire les objets qui se trouvent dans la scène filmée. La liste des objets est affinée en éliminant les objets qui ne sont pas visibles (en calculant une visibilité horizontale et verticale). Pour chaque objet, une liste des tags est calculée à partir des ressources extérieures (e.g., localisation, mots clefs, tags extraits de la page wikipedia associée). Un ranking des tags est effectué en se basant sur des critères spatiaux (la distance entre la caméra et le centre de l'objet, l'aire de l'objet dans le frame annoté, etc.). Ce système permet ensuite de retrouver des segments de vidéo à partir des requêtes textuelles en calculant une similarité entre les mots de la requête et les tags associés à chaque segment de vidéo de la base.

Dans [Shahabi et al., 2010], les auteurs présentent un framework d'aide à la décision basée sur l'information géospatiale. Leur principale contribution est d'avoir défini une architecture trois tiers qui, en se basant sur une base de données qui intègre des informations provenant de plusieurs sources (images satellitaires, cartes, GIS datasets, données temporelles, flux vidéo) pour répondre à des requêtes spatiotemporelles. Un grand effort a été fait pour le développement d'une interface qui facilite une bonne visualisation et interaction (leur proposition intègre les solutions de visualisation existantes (Google Maps<sup>30</sup>, Google Earth<sup>31</sup>) en améliorant l'interaction par le rajout d'une barre temporelle). Toutefois, le modèle de données et la façon dont les sources de données sont intégrées (e.g., vidéo streams) ne sont pas clairement définis. Le système ne prend pas non plus en compte la géométrie des champs de vue des caméras, mais uniquement les positions.

Dans [Debnath and Borcea, 2013], une approche d'annotation des images en se basant sur la localisation et l'orientation de la caméra est présentée. L'originalité réside dans le fait qu'entre la localisation et des caractéristiques optiques de la caméra (angle de vue), le système proposé (TagPix) calcule une distance entre l'utilisateur et différents objets situés dans l'aire de visibilité de la caméra afin de choisir le tag le plus pertinent. Les principaux points de similarité avec notre approche résident dans le calcul du champ de vue et la distance vus par la caméra sans avoir accès au contenu. TagPix vise l'annotation des photos donc ne considère pas la mobilité des objets et des caméras ni les requêtes de type trajectoire.

---

30. <https://www.google.fr/maps/preview>

31. <http://www.google.com/earth/>

Dans [Hwang et al., 2003], les auteurs proposent un schéma de métadonnées basé sur le modèle MPEG7<sup>32</sup> qui permet de stocker des informations de géolocalisation liées aux objets présents dans les scènes d'une vidéo. Un système a été développé (VideoGIS) qui permet à partir d'un objet de la vidéo sélectionnée par l'utilisateur d'avoir la localisation de l'objet et à partir d'un objet sélectionné sur une carte d'avoir toutes les vidéo dans lesquelles l'objet apparaît. L'approche proposée se rapproche de la nôtre par l'utilisation de la localisation et de la direction de la caméra pour lier des objets du monde réel aux scènes d'une vidéo, mais aucune explication détaillée n'est fournie concernant l'utilisation de l'information spatiale et temporelle.

Dans [Simon and Fröhlich, 2007], les auteurs présentent une analyse des besoins que les applications géospatiales utilisant les divers capteurs intégrés dans les dispositifs mobiles doivent accomplir. La plus importante conclusion de leur étude est que les requêtes les plus pertinentes pour un contexte d'application mobile sont celles de visibilité (en se basant sur la localisation de l'utilisateur et sur une modélisation de son champ visuel le système doit retourner les objets potentiellement "vus" par l'utilisateur). En se basant sur une modélisation 2,5D qui associe à chaque objet spatial du monde réel un polygone (la surface que l'objet occupe) et une valeur qui constitue la hauteur de l'objet (e.g., bâtiment, monument) et à partir de la localisation de l'objet, l'application présentée intersecte le cercle ayant comme centre le point de localisation de l'utilisateur et un rayon  $R$  paramétrable avec les objets situés dans le rayon et visibles par l'utilisateur (en fonction de leur disposition et leur hauteur). Les points d'intérêt qui composent le résultat sont restitués dans un format XML appelée "Visibility model" qui associent à chaque point d'intérêt les attributs : identifiant, nom, description, url, longitude, latitude, altitude, distance (par rapport à l'utilisateur), orientation (par rapport à l'utilisateur) et une largeur du champ de vue calculée.

---

32. <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>

Approche	Application	Réseau R/T	Requête	Cameras mobiles		Position	
				D,C	Représentation	Géo.	Sym.
[Liu et al., 2009]	Annotation	-/-	KW	D	Geom(t, p, $\alpha$ , d, R)	+	-
[Zhao et al., 2011]	Gestion du trafic	-/-	R, NN	-	-	+	-
[Shen et al., 2011]	Annotation	-/-	KW	D	Geom(t, p, $\alpha$ , d, R)	+	-
[Shahabi et al., 2010]	Aide à la décision	+/-	R, NN, V	-	-	+	-
[Debnath and Borcea, 2013]	Annotation	-/-	MC	D	Geom(p, $\alpha$ , d, distance())	+	-
[Hwang et al., 2003]	Schéma de métadonnées	?/?	?	?	?	+	-
[Simon and Fröhlich, 2007]	Application mobile geospatiale	-/-	V	D	Geom(p, $\alpha$ , d, distance())	+	-

TABLE 2.2 – Comparaison des approches

## Conclusion et objectifs

Après analyse de l'état de l'art, nous pouvons conclure que :

- à cause du volume croissant de contenus vidéo acquis par le grand nombre de capteurs vidéos déployés dans le cadre des systèmes de vidéosurveillance implémentés dans les rues, dans les moyens de transport et à l'intérieur des bâtiments, des stations de train ou de métro, etc., et en général dans tous ce qui touche la vie quotidienne (dispositifs mobiles, voitures, etc.), il y a un besoin de plus en plus grand de s'appuyer sur des éléments décrivant le contexte (e.g., géolocalisation, orientation, installation, contexte technique) des capteurs vidéos pour développer des méthodes et des outils de filtrage du contenu vidéos. La plupart des approches existantes se focalisent sur le développement des outils d'analyse du contenu vidéo qui puissent détecter de façon automatique des activités, personnes, événements avec des performances qui dépendent du cas d'utilisation et de la variabilité de la qualité du contenu, sans considérer un pré-filtrage des contenus, en se basant sur des éléments du contexte ;
- dans la démarche de changement de paradigme par la "traduction" de la problématique de filtrage des contenus vidéos issus des systèmes de vidéosurveillance dans un problème de gestion et interrogation des données spatio-temporelles, nous avons analysé les modèles de données spatio-temporelles existantes et les algorithmes et méthodes d'accès associées et avons fait les observations suivantes :
  - la plupart des approches qui traitent le mouvement des objets dans des réseaux routiers ou de transport supposent l'existence d'une étape en amont de numérisation qui génère le graphe routier (les noeuds du graphe qui sont les intersections des routes et éventuellement des points intermédiaires où la courbure de la route change), ce qui constitue une hypothèse assez forte ; numériser, stocker et gérer un réseau routier urbain représente une tâche lourde ;
  - le besoin de l'utilisation des modèles de données standardisés (pour la description des réseaux routier et de transport) n'est pas considéré.
- les approches qui se basent sur des données spatio-temporelles pour différentes applications comme la gestion du trafic ou le filtrage et la recherche des contenus vidéos ne proposent pas de modèles de données qui intègrent des informations concernant tous les éléments auxquels nous nous intéressons : réseau routier, réseau de transport, objets mobiles et caméras.
- dans la plupart des travaux existants les géométries des champs de vue des caméras sont construites (au moment de l'entrée dans le système) pour chaque frame et sont

stockées telles quelles ; dans le cas d'un système de vidéosurveillance cela peut vite apporter une surcharge importante.



# Chapitre 3

## Métadonnées pour la vidéosurveillance

### 3.1 Introduction

La définition la plus générique des métadonnées est "des données sur des données". Une définition plus précise des métadonnées est " l'information structurée qui décrit, explique, localise la ressource et en facilite la recherche, l'usage et la gestion. " <sup>1</sup>. Donc, le principal but des métadonnées est de fournir des informations complémentaires ou extraites automatiquement du contenu multimedia qui soient directement exploitables par des machines dans le cadre des systèmes d'informations multimedia afin de faciliter la gestion et la recherche des contenus multimedia. Beaucoup d'études sur les métadonnées multimedia [Manzat, 2013], [Raphael Troncy, 2011] concernent surtout les domaines de l'archivage, des journaux télévisés, des médias sociaux etc. Les travaux qui se concentrent sur les métadonnées pour la vidéosurveillance sont beaucoup moins nombreux et n'ont pas le même niveau de maturité car la "numérisation" de ces systèmes (par l'utilisation des caméras IP) est un processus relativement récent.

Les particularités des systèmes de vidéosurveillance qui posent problèmes dans la démarche de définition d'un format (structure + dictionnaire) de métadonnées sont les suivantes :

– peu de requêtes et beaucoup de contextes (la plupart des "requêtes" vise la

---

1. National Information Standards Organization, 2004. Understanding metadata.

- détection ou l'identification d'une personne ou d'un véhicule dans des collections de vidéos prises dans des contextes très hétérogènes) ;
- grand volume : impossibilité de tout indexer (à cause du risque d'avoir des index aussi volumineux voire plus que les données) ;
- un grand nombre de systèmes ayant des différents propriétaires, utilisant leurs propres formats de données et de métadonnées ce qui a comme conséquence un manque d'interopérabilité entre les différents systèmes (e.g., c'est difficile pour les agents de la RATP d'avoir une utilisation directe des contenus issues des systèmes de la SNCF).

Comme expliqué dans l'introduction de ce mémoire, à partir de l'analyse du contexte d'usage de la vidéoprotection, la principale question qui se pose est : quelle « boîte à outils » fournir aux opérateurs pour les assister dans leur tâche et les aider à réduire le temps de recherche dans des situations d'urgence avec un impact direct sur la sécurité sociétale, tout en favorisant l'interopérabilité des différents systèmes de vidéosurveillance ? Dans ce sens, le standard international ISO22311 vise à fournir un profil d'interopérabilité qui constitue un format d'échange de données. Le besoin de s'appuyer sur d'autres types d'informations dans un contexte de filtrage et de recherche d'extraits vidéo dans une collection de vidéosurveillance (par exemple dans le cadre d'une enquête judiciaire) ressort de la norme ISO 22311 publiée grâce aux efforts communs des principaux acteurs du domaine de la vidéosurveillance en France et au niveau international. La norme spécifie un format commun pour les données qui peuvent être extraites des systèmes de collecte du domaine de la vidéosurveillance, par exemple à des fins d'enquête. Le dictionnaire de métadonnées proposé dans la norme contient des éléments de localisation des caméras, description de la scène observée par chaque caméra, caractéristiques optiques de la caméra, etc. Il définit également les exigences techniques minimales garantissant que le contenu numérique de vidéosurveillance exporté soit compatible avec les systèmes de relecture. Enfin, il cherche à établir un niveau de qualité approprié pour les données et de contenir toutes les informations de contexte (métadonnées) nécessaires pour leur traitement. Ce dictionnaire est pourtant non-exhaustive. En plus, le texte de la norme ne contient pas d'éléments concernant la modalité de renseignements ou de mise en oeuvre du standard. En conséquence, par la suite nous proposons un étude des métadonnées utiles dans le contexte de la vidéosurveillance que nous avons réalisé dans le cadre du projet ANR METHODEO, avec un focus sur les caractéristiques des conditions de prise de vue des caméras de vidéosurveillance. La deuxième partie de ce chapitre est consacré à la structuration du dictionnaire de métadonnées dans un format qui vise à enrichir la norme ISO22311.



L'utilisation croissante de la vidéosurveillance dans les lieux publics conduit à l'élaboration de systèmes de surveillance complexes comprenant un grand nombre de caméras et qui ont à gérer des quantités massives de données. Les systèmes de surveillance doivent satisfaire un ensemble d'exigences fonctionnelles allant de la transmission vidéo en temps réel à la récupération des données archivées, en passant par les outils d'analyse automatique et par l'annotation des données vidéos en se basant sur les métadonnées.

Historiquement, les systèmes de surveillance ont été considérés comme des solutions fermées et sur mesure loin d'être normalisées. Mais l'utilisation intensive de la vidéo surveillance au cours des dernières années a accéléré la demande de la normalisation et d'utilisation des outils d'analyse du contenu dans ce domaine d'application. Deux aspects sont à considérer dans la normalisation : le format de communication des informations vidéo et les métadonnées modélisant les informations liées aux contenus (e.g., données techniques ou de localisation liées à la caméra) et les résultats des analyses effectuées sur les contenus. Ces métadonnées doivent être indépendantes des algorithmes et des systèmes d'analyse. Elle peuvent être utilisées pour réaliser différentes tâches liées à l'exploitation du contenu (e.g., filtrage "négatif" du contenu, évaluation des algorithmes d'analyse du contenu, recherche d'information, archivage), à l'installation des caméras (e.g., paramétrage du système) et au contrôle du système de vidéosurveillance.

Par la suite, nous allons présenter les critères de classification pertinents pour notre analyse.

## **3.2 Classification des métadonnées**

Plusieurs auteurs [Bohm and Rakow, 1994], [De Sutter et al., 2006], [Lassila, 1998],[Pereira et al., 2008] ont essayé de réaliser une classification des métadonnées en fonction de différentes caractéristiques comme l'objectif principal de la description ou le niveau sémantique concerné par les métadonnées. Dans notre contexte, nous avons retenu trois critères de classification des métadonnées caractérisant les conditions de prise de vue dans les systèmes de vidéosurveillance : (1) l'objet de la description fournie par les métadonnées à plusieurs niveaux de granularité (métadonnées associées au systèmes, métadonnées associées aux scènes filmées par les caméras qui composent le système et métadonnées associées aux scénarios qui se suivent dans des scènes), (2) la mutabilité des métadonnées

(métadonnées qui changent ou pas avec le temps), (3) la modalité de renseignement des métadonnées (métadonnées automatiques, semi-automatiques et manuelles).

Critère de classification	Types de métadonnées	Exemples
L'objet décrit	<p>Métadonnées du système de vidéosurveillance</p> <p>Métadonnées de la scène</p> <p>Métadonnées du scénario</p>	<p>Localisation dans le monde (GPS) et par rapport au système des caméras (carte des caméras); Description technique des caméras.</p> <p>Coordonnées de la scène observée par chaque caméra, conditions météo, type de scène</p> <p>Description des objets et des événements</p>
La mutabilité	<p>Métadonnées statiques</p> <p>Métadonnées dynamiques</p>	<p>Propriétaire, Type de caméra.</p> <p>Position d'une caméra mobile</p>
La génération	<p>Métadonnées manuelles</p> <p>Métadonnées semi-automatiques</p> <p>Métadonnées automatiques</p>	<p>Annotations textuelles</p> <p>Utilisation des outils d'analyse du contenu avec ou sans intervention humaine.</p>

TABLE 3.1 – Classification

La section suivante représente un zoom sur les caméras de vidéosurveillance et tous les paramètres associés qui influencent les conditions de prise de vue.

### 3.3 Caméras de vidéosurveillance

Les performances de tout système de vidéosurveillance résident dans les performances des caméras qui le composent et de leur installation (position, réglages, etc.).

Avec le développement technologique, l'installation des caméras (avec d'autres capteurs intégrés comme le GPS ou la boussole) peut se faire de manière efficace du point de vue du coût et de la consommation d'énergie sur des plateformes fixes ou sur tout dispositif mobile. Comme conséquence, des caméras sont installées dans les rues, dans des bâtiments, sur des véhicules publics comme les bus, les voitures de police et les ambulances. La caméra embarquée s'est déjà largement généralisée en Russie, en Asie et aux États-Unis (depuis le 1er novembre 2012, tous les véhicules neufs aux États-Unis sont équipés d'une caméra embarquée) et commence à prendre de l'ampleur en France aussi. A Las Vegas, elle est même obligatoire dans les taxis et les transports publics. En Russie, où des piétons se jetaient volontairement sous les roues des voitures pour ensuite accuser les automobilistes, les dashcams (des petites caméras que l'on fixe à l'aide d'une ventouse sur le pare-brise des voitures) se sont vendues en très grand nombre<sup>2</sup>.

Les scènes capturées par la caméra changent avec la rotation de la caméra et avec son déplacement. Les technologies modernes développées autour de ces caméras facilitent l'extraction d'une multitude de métadonnées qui permettent sans même avoir accès à l'image d'extraire des informations précises sur la scène filmée par la caméra. Ces métadonnées peuvent être sauvegardées avec l'image ou séparément dans des différents formats [Sèdes, 1998], [Manzat, 2013].

Dans les sections suivantes, nous allons présenter ces métadonnées en expliquant leur utilité pour l'exploitation des contenus issus des systèmes de vidéosurveillance.

### 3.3.1 Champ de vue d'une caméra

Une caméra située à une certaine position, avec une certaine orientation et installation peut filmer une certaine aire. Le champ de vue représente l'aire de la scène surveillée par une caméra. Le champ de vue est vertical et horizontal. Cette aire est influencée par un nombre de paramètres spécifiques à chaque caméra. La Figure 3.1 illustre le champ de vue et ses paramètres (présentés dans le Tableau 3.2 et expliqués par la suite).

Les capteurs d'image de la caméra sont des matrices de cellules photosensibles qui récupèrent la lumière et la transfèrent sous forme de tension électrique à un convertisseur qui se charge ensuite de recomposer tout ça en pixels. Il y a deux types de

---

2. <http://www.ladepeche.fr/article/2014/02/08/1813711-une-camera-dans-toutes-les-voitures.html>

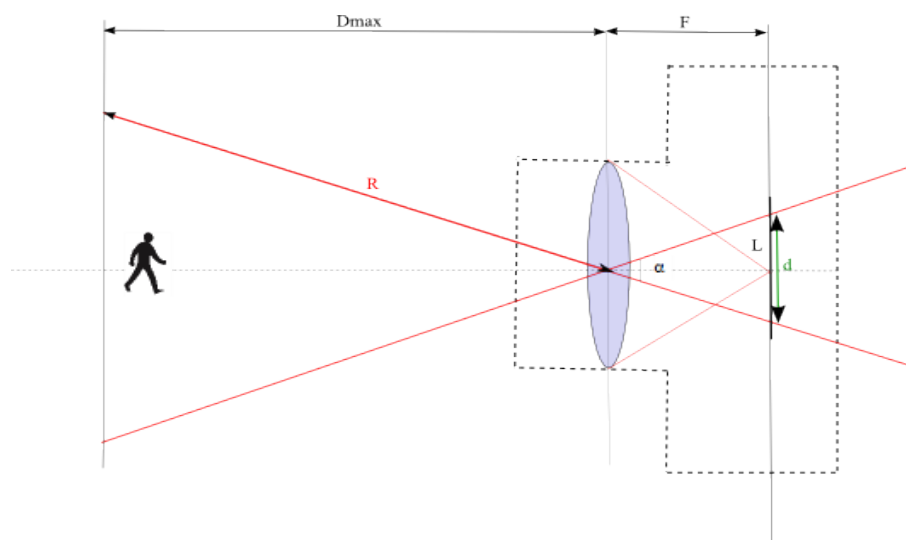


FIGURE 3.1 – Champ de vue d'une caméra

capteurs basés sur deux technologies : CCD et CMOS<sup>3</sup>, les capteurs CCD ayant une meilleure qualité de l'image que les capteurs CMOS, qui s'explique principalement par une meilleure sensibilité à la lumière.

La dimension classique du film sur lequel est "imprimée" l'image est de 24 mm x 36 mm (qui est appelé souvent 24x36 ou film 35mm). Aujourd'hui, la dimension des capteurs existant vont de 40% à 100% de cette dimension. Les dimensions des capteurs les plus utilisés aujourd'hui sont : 1/4" (3,2mm x 2,4mm) , 1/3.6"(4mm x 3mm), 1/3"(4,8mm x 3,6mm) , 1/2.5"(5,76mm x 4,29mm), 1/2"(6,4mm x 4,8mm), 2/3"(8,8mm x 6,6mm) et 1"(12,8mm x 9,6mm). Les différentes tailles du capteur engendrent des champs de vue correspondants différents en relation aussi avec l'angle de vue au moment de la capture et la distance focale de la lentille utilisée par la caméra.

La relation entre l'angle de vue, la taille du capteur et la distance focale de la lentille est donnée par la formule 3.1 [Hecht, 2001]. Pour la même distance focale, si la taille du capteur diminue, le champ de vue devient plus étroit. Autrement, pour la même taille du capteur, si la distance focale de la lentille est plus petite, le champ de vue devient plus large mais moins profond (voir Figure 3.2<sup>4</sup>). La relation entre l'angle de vue et R est donnée par l'équation 3.2. La valeur de R est celle pour laquelle

3. [http://www.axis.com/fr/files/whitepaper/wp\\_ccd\\_cmos\\_40722\\_en\\_1010\\_lo.pdf](http://www.axis.com/fr/files/whitepaper/wp_ccd_cmos_40722_en_1010_lo.pdf)

4. [http://www.honeywellvideo.com/documents/L-CAMTECGDB\\_D.pdf](http://www.honeywellvideo.com/documents/L-CAMTECGDB_D.pdf)

Paramètre	Description
$\alpha$	Angle de vue (horizontal ou vertical)
d	Taille du capteur
F	Distance focale de la lentille
$D_{max}$	Distance de visibilité maximale
$D_{min}$	Distance de visibilité minimale
	(apparaît avec l'inclinaison de la caméra (voir section suivante))
R	le coté du triangle formé par le champ de vue de la caméra
L	Taille du film classique (35mm)
H	La hauteur d'installation de la caméra
$\theta$	L'angle d'inclinaison de la caméra
$H_{min}$	La hauteur de l'objet cible

TABLE 3.2 – Paramètres du champ de vue d'une caméra

l'objet est "visible" dans l'image. Cette valeur est différente en fonction de la taille de l'objet cible et de l'application (e.g., détection, reconnaissance, identification).

$$\alpha = 2 \arctan \frac{d}{2F} \quad (3.1)$$

$$\alpha = 2 \arctan \frac{d(R \cos(\frac{\alpha}{2}) - F)}{2FR \cos(\frac{\alpha}{2})} \quad (3.2)$$

$$\cos \frac{\alpha}{2} = \frac{D_{max}}{R} \quad (3.3)$$

Prenons l'exemple de la caméra d'un Iphone4 (que nous avons également utilisé dans nos tests). La caméra principale d'un Iphone4 a les caractéristiques suivantes :

- **Type de capteur** : CMOS
- **Taille du capteur** : 4.54 x 3.39 mm (diagonale 5.67 mm)
- **Résolution** : 5.0 Mpixels (2592 x 1936) 1/3.2"
- **Distance focale** : 3.85 mm

En conséquence l'angle de vue horizontal est donné par le calcul 3.4 et l'angle de vue vertical est donné par le calcul 3.5.

$$\alpha = 2 \arctan \frac{4.52}{2 * 3.85} = 60.8^\circ \quad (3.4)$$

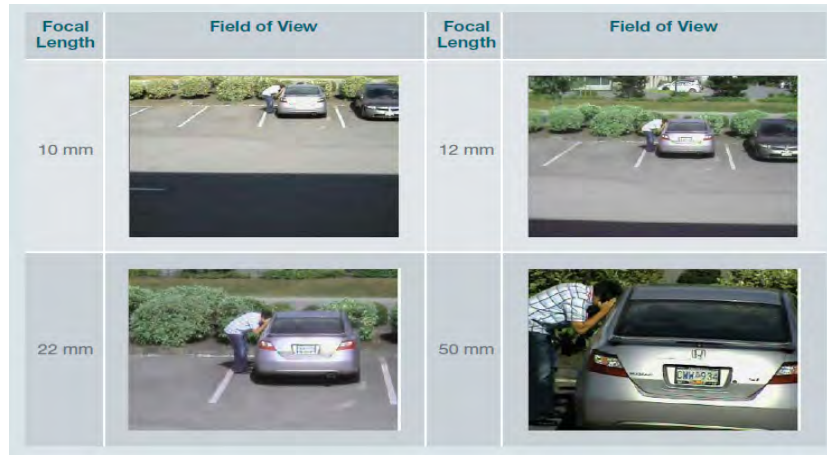


FIGURE 3.2 – Champ de vue d’une caméra pour des différentes distances focales

$$\alpha = 2 \arctan \frac{3.39}{2 * 3.85} = 47.5^\circ \quad (3.5)$$

La position de la caméra et les caractéristiques optiques de son champ de vue ne sont pas suffisantes pour déterminer précisément la scène ” vue ” par une caméra à un moment donné (voir Figure 3.3). Des caméras situées dans une même position peuvent ”regarder” dans des directions différentes. L’orientation d’une caméra est importante pour définir la direction de vue.

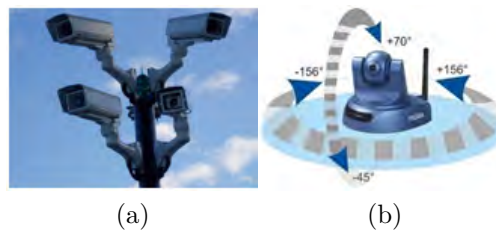


FIGURE 3.3 – Des caméras ayant la même position mais des champs de vue différents

L’orientation d’une caméra est définie comme l’angle horizontal de déviation (donc celui de déviation de l’axe z de la figure 3.4(a)) par rapport à une référence qui peut être le Nord Géographique ou le Nord Magnétique (angle appelé aussi azimut). La formule 3.6 montre que l’azimut par rapport au Nord Magnétique ( $Z_v$ ) est égal à l’azimut par rapport au Nord Géographique ( $Z_c$ ) plus une déviation. La déviation

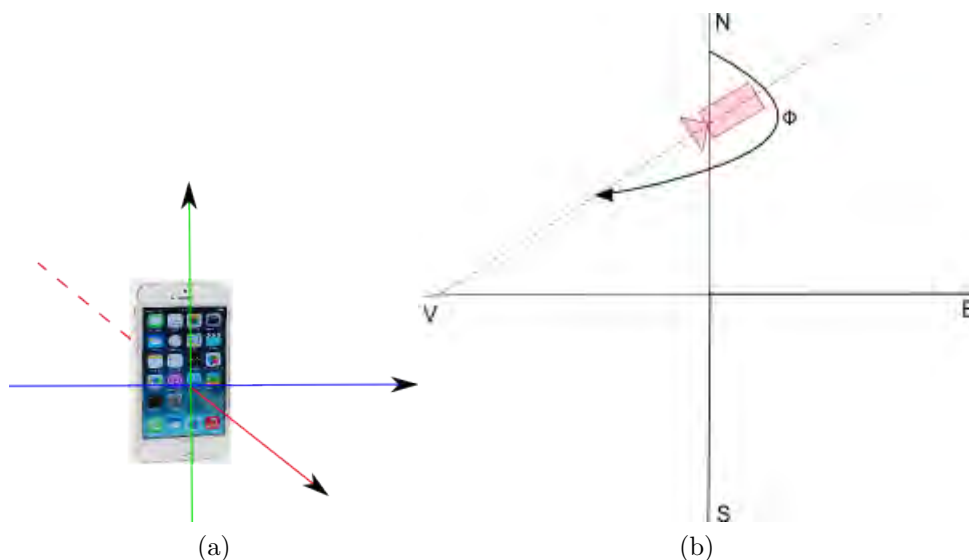


FIGURE 3.4 – Direction de vue d'une caméra

(cf. la formule 3.7) est l'angle entre le Nord Géographique et celui Magnétique ( $D$ ) et une déviation interne de la boussole ( $I$ ) (due à d'autres champs magnétiques présents dans le voisinage de la boussole par exemple). Dans nos travaux nous allons considérer  $I=0$ .  $D$  dépend de la position GPS et de la date et peut être trouvé sur des sites comme celui de l'organisme américain NOAA (National Geophysical Data Center)<sup>5</sup>.

$$Z_v = Z_c + W \quad (3.6)$$

$$W = D + I \quad (3.7)$$

A part la localisation, les caractéristiques optiques et l'orientation d'une caméra, ses paramètres d'installation impactent aussi son champ de vue. Par la suite nous allons présenter ses paramètres d'installation.

---

5. <http://www.ngdc.noaa.gov/geomag-web/#declination>

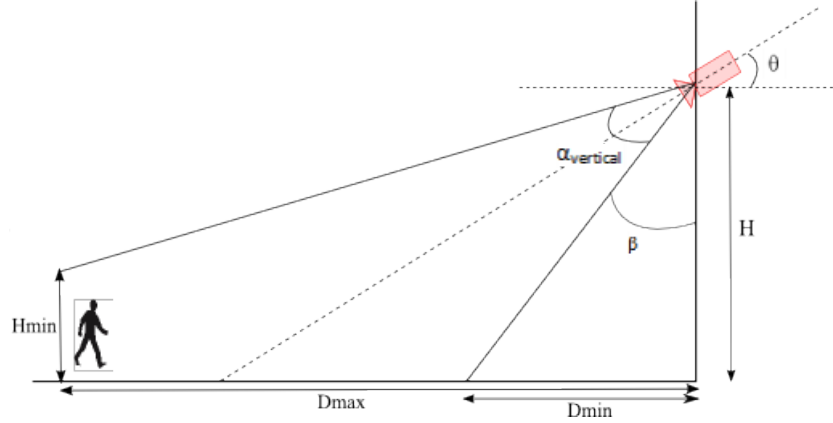


FIGURE 3.5 – Installation d’une caméra et ses paramètres

### 3.3.2 Installation d’une caméra

Pour une meilleure précision du champ de vue, l’installation de la caméra peut être prise en compte. La hauteur et l’angle d’inclinaison de la caméra déterminent la distance maximale visible par la caméra ( $D_{max}$ ) et la zone d’ombre ( $D_{min}$ ) qui n’est pas visible par la caméra. L’installation peut changer avec le temps donc nous allons définir l’installation d’une caméra comme le triplet (inclinaison, hauteur, t).

La figure 3.5 montre l’installation d’une caméra avec les paramètres qui influencent l’aire vue par la caméra. Les équations 3.8, 3.9, 3.10 illustrent les formules utilisées pour le calcul de  $D_{max}$  et  $D_{min}$ .

$$\beta = 90^\circ - \theta - \frac{\alpha_{vertical}}{2} \quad (3.8)$$

$$\tan \beta = \frac{D_{min}}{H} \quad (3.9)$$

$$\tan(\delta) = \frac{D_{totale}}{H} = \frac{D_{totale} - D_{max}}{H_{min}} \quad (3.10)$$



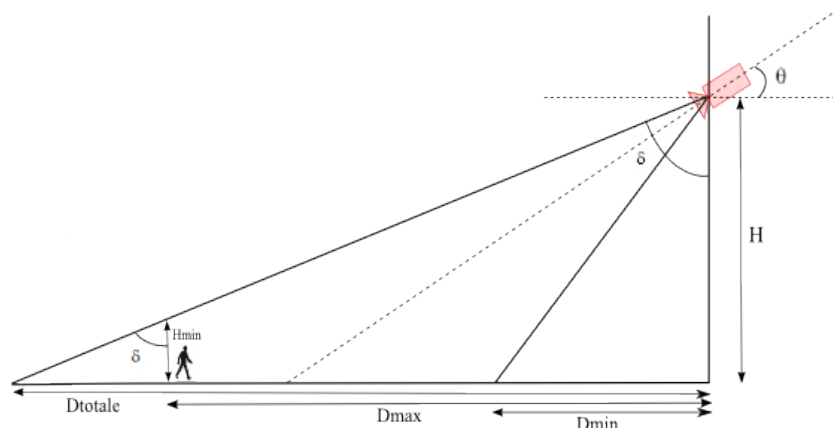


FIGURE 3.6 – Installation d’une caméra : Distance de visibilité maximale

### 3.4 Proposition d’un dictionnaire de métadonnées pour la vidéosurveillance

Par la suite, nous allons présenter, en respectant la classification proposée dans la section 3.2 le dictionnaire de métadonnées utiles dans le cadre de la vidéosurveillance. Le dictionnaire est structuré en trois grandes catégories : (1) Métadonnées décrivant le système de vidéosurveillance, (2) Métadonnées décrivant la scène observée par chaque caméra du système et (3) Métadonnées décrivant des scénarios d’intérêt ayant lieu dans les scènes filmées. Chaque catégorie contient des sous-catégories correspondant aux métadonnées statiques et dynamiques. Pour chaque type de métadonnée, nous allons donner une courte description montrant son utilité et la façon dont elle est renseignée (manuellement, semi-automatiquement, automatiquement). Un tel dictionnaire peut servir à plusieurs tâches concernant les organismes (publiques ou privés) qui exploitent au quotidien les systèmes, les installateurs qui doivent paramétrer les systèmes, les autorités de contrôle des systèmes mises en place et des enquêteurs qui doivent exploiter les vidéos issues de plusieurs systèmes.

### 3.4.1 Des métadonnées associées aux systèmes de vidéosurveillance

#### Métadonnées Statiques

Les informations génériques importantes relatives aux systèmes de vidéosurveillance sont : le propriétaire (e.g., la RATP, un centre commercial, une banque) et si celui-ci est un client privé (e.g., un centre commercial) ou une collectivité publique (e.g., la mairie d'une ville) ce qui induit des contraintes (législatives) différentes sur l'utilisation des données (voir tableau 3.3).

Libelle	Commentaire	Modalité de renseignement
<b>Propriétaire</b>	Chaîne de caractères (e.g., nom de l'opérateur de transport, d'un centre commercial)	Manuelle
<b>Privacité</b>	Public/Privé (il est important de disposer de cette information pour savoir si les données sont accessibles ou non)	Manuelle

TABLE 3.3 – Métadonnées statiques associées aux systèmes de vidéosurveillance

Chaque système de vidéosurveillance est composé de plusieurs caméras caractérisées par des descriptions statiques (voir tableau 3.4).

Libelle	Commentaire	Modalité de renseignement
<b>Type de caméra</b>	Couleur/Noir et Blanc, Jour/Nuit avec ou sans spot infrarouge, Thermique, Dôme	Manuelle/Automatique
<b>Fabricant</b>	Chaîne de caractères (e.g., Axis, Canon)	Manuelle/Automatique
<b>Modèle de caméra</b>	Chaîne de caractères (identification de l'appareil utilisé)	Manuelle/Automatique
<b>Type de transmission</b>	Analogique/IP	Manuelle/Automatique
<b>Format de données</b>	Chaîne de caractères	Manuelle

TABLE 3.4 – Métadonnées statiques associées aux caméras

Chaque système de vidéosurveillance peut se voir associé aussi un ensemble d'**Outils d'analyse vidéo**. Il est essentiel de connaître les fonctionnalités et les contraintes d'exécution des outils d'analyse vidéo pour pouvoir mettre en place une indexation intelligente des contenus vidéo. Par exemple, un outil peut détecter et classifier les objets présents dans la scène. La classification donne les performances attendues si les objets ont une densité de pixels horizontale de 250 pixels au mètre. Du coup la caméra doit être installée à une distance/hauteur/inclinaison appropriée par rapport à la scène observée.

Les informations concernant les contraintes d'exécution des outils sont renseignées par les développeurs. Dans [Codreanu et al., 2011] nous présentons une proposition de modèle de description des outils d'analyse des contenus multimédias et de formulaire (conforme au modèle) en ligne qui est renseigné par les développeurs des outils et sauvegardé dans une base de données. Dans [Brut et al., 2011c] et [Brut et al., 2011b] nous illustrons l'utilisation du modèle de description pour la sélection des outils en fonction du besoin de l'utilisateur (e.g., si on veut détecter un sac perdu nous n'allons pas exécuter d'algorithmes qui détectent des personnes) et des éléments du contexte (en fonction des performances des algorithmes dans différents contextes comme la luminosité, les conditions météo).

Les deux éléments du modèle présenté dans [Codreanu et al., 2011] obligatoires pour

le contexte de la vidéosurveillance sont la fonctionnalité de l'outil (e.g., reconnaissance, détection de personne, extraction de couleurs) qui sera comparée avec le besoin de l'utilisateur (e.g., l'enquêteur) et les conditions (ou contraintes) d'exécution qui représentent les conditions de format de données, de qualité d'enregistrement, de résolution, de taille maximum et minimale des objets à détecter, etc. qui doivent être remplies pour que l'outil fonctionne comme prévu (voir tableau 3.5). Dans l'analyse du besoin les fonctions d'analyse vidéo identifiées à partir des entretiens avec les opérationnels dans le cadre du projet METHODEO et l'étude de l'existant sont : détection du mouvement, détection de la dégradation ou du mauvais fonctionnement des caméras, monitoring du trafic routier, comptage de personnes, monitoring de groupes de personnes, reconnaissance de personnes, détection et suivi de personnes et de groupes de personnes, reconnaissance de visages, détection de visages, détection d'intrusions, détection et suivi de véhicules et de groupes de véhicules , détection de bagages abandonnés, reconnaissance de gestes, reconnaissance d'activités de personnes et de groupes de personnes (la liste n'est pas exhaustive). Comme perspective nous envisageons de définir un dictionnaire de fonctionnalités qui pourra être utilisé dans le renseignements des métadonnées.

Le besoin d'avoir ce type de contrôle est montré par l'apparition des normes techniques comme celles publiées dans l'arrêté d'août 2007<sup>6</sup> ou dans la norme EN50132-7<sup>7</sup>. Ces normes ont identifié différentes grandes catégories de tâches qui doivent être accomplies par les systèmes de vidéosurveillance et avec des exigences en termes de qualité d'image correspondantes [AN2V, 2014].

La densité de pixels au mètre est relative car pour que les pixels soient exploitables (jour et nuit) plusieurs éléments doivent être pris en compte par les spécialistes : contraste, luminosité, vitesse d'obturation, champ de vue etc.

## **Métadonnées Dynamiques**

Les métadonnées dynamiques décrivant les caméras qui composent un système de vidéosurveillance concernent : le champ de vue et les caractéristiques optiques des caméras (voir tableau 3.7), les informations de localisation (voir tableau 3.8) et des caractéristiques de qualité d'image (voir tableau 3.9)

---

6. <http://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000649127>

7. [http://www.en-standard.eu/csn-en-50132-7-ed-2-alarm-systems-cctv-surveillance-systems-for-use-in-security-applications-part-7-application-guidelines/?gclid=CjwKEAiAt4mlBRDXwt\\_m9ICU4DcSJAAS\\_X0WqxuzN2ouEvRFO3pwIen0DBvY\\_4Dop7cPVKpYo-6eSBoCVdLw\\_wcB](http://www.en-standard.eu/csn-en-50132-7-ed-2-alarm-systems-cctv-surveillance-systems-for-use-in-security-applications-part-7-application-guidelines/?gclid=CjwKEAiAt4mlBRDXwt_m9ICU4DcSJAAS_X0WqxuzN2ouEvRFO3pwIen0DBvY_4Dop7cPVKpYo-6eSBoCVdLw_wcB)

Libelle	Commentaire	Modalité de renseignement
<b>Fonctionnalité</b>	Chaîne de caractères (e.g., reconnaissance faciale, reconnaissance de plaque, détection d'intrusion, vol, bagarre, regroupement, passage, objet suspect)	Manuelle
<b>Conditions d'utilisation</b>	Paire paramètre - seuil min/max (e.g., frame-rate, résolution, distance maximum de détection, taille min/-max des objets)	Manuelle

TABLE 3.5 – Métadonnées statiques associées aux outils d'analyse du contenu

Catégories (EN50132-7)	Pixel par mètre	Objectifs	Applications types
<b>Exploitation</b>	N/A	Caméra qui n'est pas destinée à une surveillance mais à l'accomplissement d'une autre tâche	Comptage des personnes, flux de trafic
<b>Surveillance/Contrôle</b>	>10	Permet d'avoir une compréhension sommaire d'un événement afin de déclencher ou non une action	Gares, Zone interdite, détection périmétrique
<b>Détection</b>	>30	Permet de faire une levée de doute, de différencier des individus entre eux et de comprendre leur interaction afin de déclencher ou non une action	Parking, Zones industrielles, Ports, Centres commerciaux, Événements
<b>Reconnaissance</b>	>100	Permet de reconnaître un individu ou un objet qui a été déjà détecté auparavant	Rayons, Points de vente, Agences, Couloirs, Escalators, Visualisation de plaques d'immatriculation
<b>Identification</b>	>400	Permet d'identifier un individu ou un objet avec une grande probabilité	Entrées, portes de services, Escalators
<b>Inspection</b>	>800	Permet de distinguer	Entrées, portes de services, Escalators

TABLE 3.6 – Synthèse des exigences en termes de qualité d'images en fonction des applications cibles des caméras de vidéosurveillance [AN2V, 2014]

Libelle	Commentaire	Modalité de renseignement
<b>Taille du capteur</b>	Double X Double :la taille (horizontale et verticale) (en mm) du capteur utilisé par le modèle de caméra choisi (comme expliqué dans la section 3.2)	Manuelle/Automatique
<b>Distance focale</b>	Double :la distance focale en mm caractéristique au modèle de caméra (comme expliqué dans la section 3.2)	Manuelle/Automatique
<b>Angle de vue (horizontal ou vertical)</b>	Double (en degrés)	Automatique (Calculé comme expliqué dans la section 3.2)
<b>Pan, Tilt, Zoom, Roll</b>	Valeur (Double) ou fourchette de valeurs	Automatique(Calculé à partir d'un outil de calibration)
<b>Hauteur d'installation</b>	Valeur (Double)	Manuelle

TABLE 3.7 – Métadonnées dynamiques concernant le champ de vue d'une caméra

Libelle	Commentaire	Modalité de renseignement
<b>Système de référence</b>	Chaîne de caractères	Manuelle/Automatique
<b>Position Géométrique (2D/3D)</b>	Coordonnées 2D/3D	Manuelle/Automatique
<b>Position Symbolique</b>	Chaîne de caractères ou valeur numérique	Manuelle/Automatique

TABLE 3.8 – Métadonnées dynamiques concernant la localisation d'une caméra

Libelle	Commentaire	Modalité de renseignement
<b>Type de compression</b>	Les paramètres de compression (taux, codec, etc.)	Automatique (extraction du codec)
<b>Qualité de l'enregistrement :</b>	Utile d'avoir ces informations pour calculer une mesure "d'utilisabilité" du contenu vidéo	
<b>Évolution du framerate/perte du signal vidéo</b>	Double	Automatique
<b>Le masquage</b>	Chaîne de caractères (Oui/Non)	Automatique(Analyse de l'évolution des contraste) Automatique(Détection de mouvement de camera (stabilisation / suivi de point d'intérêt image/Calcul de la vitesse de la caméra à partir de ses positions))
<b>Le flou/le mouvement</b>	Pixel/Seconde	
<b>Densité de pixels</b>	Pixels/Mètre	Automatique
<b>Contrôle de l'exposition</b>	WDR (Wide Dynamic Range)/BLC (Back Light Compensation)	Manuelle/Automatique
<b>Résolution de l'image</b>	Largeur x Hauteur	Automatique (Extraction du codec)

TABLE 3.9 – Métadonnées dynamiques concernant la qualité des images issues d'une caméra de vidéosurveillance

### 3.4.2 Métadonnées scène observée (description de l'environnement de la scène filmée par la caméra)

Les métadonnées décrivant l'environnement de la scène filmée par une caméra sont **dynamiques** et sont présentées de façon synthétique dans le tableau 3.10.



Libelle	Commentaire	Modalité de renseignement
<b>Situation de la caméra</b>	Indoor/Outdoor	(Semi)-Automatique (Détecteur / Apprentissage sur vecteur de descripteurs images)
<b>Type de scène</b>	Chaîne de caractères (gare, station métro, couloir, carrefour, rue, route, zone piétonne, parc, parking, monument, magasin, bus, train)	Manuelle/(Semi-)Automatique (Apprentissage sur vecteurs de descripteurs images)
<b>Présence d'objet qui masque une partie de la scène</b>	Liste de points (Contour)/Chaîne de caractères/Identifiant	Manuelle/Détection/ Apprentissage
<b>Jour/Nuit</b>	Chaîne de caractères	(Semi)-Automatique (apprentissage sur vecteurs de descripteurs images)
<b>Météo</b>	Chaîne de caractères	(Semi)-Automatique(apprentissage sur vecteurs de descripteurs images)
<b>Luminosité</b>	Double [lux]	Automatique (mesurée à l'aide du luxmètre ou calculée à partir des descripteurs)
<b>Activité particulière de la scène</b>	Chaîne de caractères	Manuelle (voir liste en dessous)/ Apprentissage sur descripteurs locaux : mouvement, forme / couleur / fréquence apparition, déformation / rigidité

TABLE 3.10 – Métadonnées dynamiques associées à la scène filmée par la caméra

Avoir des informations concernant une activité particulière dans une scène peut être utile car cela influence la qualité de l'image et donc le score "d'utilisabilité" de celle-

ci et donc le choix des outils d'amélioration de la qualité de l'image et de détection d'objets ou d'événements. Une liste non-exhaustive des activités pertinentes pour le contexte de la vidéosurveillance est :

- ouverture/fermeture des portes/barrières ;
- surveillance d'une porte avec de la lumière du jour dehors et un environnement plus sombre à l'intérieur ;
- surveillance des véhicules qui entrent dans un parking ou un tunnel avec de la lumière du jour dehors et une illumination faible à l'intérieur ;
- véhicules avec des phares puissants ou avec des clignotements de feux de signalisation ;
- environnements avec beaucoup de lumière reflétée comme les bâtiments avec beaucoup de miroirs ;
- mouvements de végétation ;

### **3.4.3 Métadonnées scénario (description des objets et des événements de la scène)**

Le fait d'avoir distingué les métadonnées décrivant la scène et celles décrivant le scénario nous amène à séparer les entités objet en objets de contexte (e.g., banques, arbres, objets sur les quais du métro) et objets d'intérêt (e.g., personne, véhicule, groupe de personnes, groupe de véhicules) et les entités événement en événements de contexte (e.g. ouverture d'une porte ou d'une barrière) et événements d'intérêt (e.g., bagarre, vol de sac). Les entités peuvent être de contexte ou d'intérêt en fonction du contexte d'utilisation des métadonnées respectives (e.g. un véhicule stationnant au bord d'une rue peut faire partie du contexte si on cherche à détecter des personnes ou d'intérêt si on cherche à détecter des véhicules).

Les métadonnées concernant les objets consistent généralement en une localisation de l'objet dans l'image (e.g., un point, une liste de points, une boîte englobante) qui se voit associé différents attributs génériques (valables pour tous les objets) et/ou spécifiques au type d'objet décrit et à l'application ou à la tâche visée. Par la suite nous allons présenter des tableaux avec les caractéristiques génériques des objets (voir tableau 3.11) et avec les descriptions spécifiques aux objets de type personne (voir tableau 3.12) et aux objets de type véhicule (voir tableau 3.13).

Type	Classe	Exemples
Objets d'intérêt	Personne	Adulte, Enfant
	Groupe de personnes	
	Véhicule	Voiture, Camion, Train, Rame de métro
	Groupe de véhicules	
	Autres	Animal, Insecte, Sac à main, Bagage, Visage, Bras, Signes distinctifs
<b>Objets de contexte</b>		Porte, Arbre, Barrière
<b>Événements d'intérêt</b>		Vol du sac à l'arrachée, Personne qui entre/sort d'une voiture
<b>Événements de contexte</b>		Ouverture d'une porte, d'une barrière

Libelle	Commentaire	Modalité de renseignement
<b>Localisation de l'objet dans l'image</b>	Point (x,y), Liste de points, Bounding box	Manuelle, (Semi)-automatique (interpolation entre 2 positions définies sur 2 images clés initialisées manuellement ou via un détecteur)
<b>Identifiant</b>	Chaîne de caractères	Manuelle, (Semi)-automatique (algorithmes de suivi d'un signature visuelle)
<b>Type d'objet</b>	Chaîne de caractères	Manuelle, (Semi)-automatique (détection/apprentissage)
<b>Taille de l'objet</b>	Double x Double (Largeur x Hauteur)	Automatique (A partir des boîtes englobantes et d'une calibration de la caméra)
<b>Position par rapport à la caméra</b>	Chaîne de caractères (Face/Dos/Profile)	Manuelle, (Semi)-automatique (apprentissage, à partir des boîtes englobantes, du mouvement, d'un détecteur)

TABLE 3.11 – Métadonnées génériques associées aux objets

Libelle	Commentaire	Modalité de renseignement
<b>Localisation du visage</b>	Point (x,y), Liste de points, Bounding box	(Semi-)Automatique (Algorithme de détection (e.g., AdaBoost ou HOG))
<b>Orientation du visage</b>	Double (Angle de rotation)	(Semi-)Automatique (Détecteur 3D)
<b>Texture du visage, Couleur de la peau et des cheveux</b>	Vecteur de caractéristiques (e.g., histogramme, coefficients de Gabor, etc.)	Automatique
<b>Couleur des habits</b>	Vecteur de caractéristiques (e.g., histogramme de couleur, matrice de cooccurrence, covariance, etc.)	Automatique
<b>Position (debout, assis, accroupi, allongé)</b>	Chaîne de caractères	(Semi-)Automatique (détecteur/apprentissage)
<b>Comportement, démarche, gestuelle</b>	Vecteur de caractéristiques	Automatique (analyse des trajectoires, déformation, détection des bras)
<b>Silhouette</b>	Liste de points (forme) et une liste de caractéristiques (histogramme de gradient, de mouvement ou de couleur)	Automatique (extracteur de caractéristiques : SIFT, SURF, STIP, HOG, HOF, HSV)
<b>Signes distinctifs : lunettes, tatouage, type de vêtement (pull, blouson, short, pantalon, etc.), marque de vêtement, chapeau, casquette, casque, sac à dos / valise, parapluie</b>	Liste de paires : forme-texte	Manuelle, (semi-)automatique (détecteur de formes, apprentissage)

TABLE 3.12 – Métadonnées génériques associées aux objets de type personne

Les différents objets présents dans les scènes participent aux événements. Les métadonnées qui décrivent les événements sont dynamiques et sont présentées dans le tableaux 3.14.

Libelle	Commentaire	Modalité de renseignement
<b>Couleur</b>	Vecteur de caractéristiques(histogramme, etc.)/Chaîne de caractères	Automatique (extracteur de descripteurs/détecteur)
<b>Marque</b>	Chaîne de caractères	Manuelle, (semi-)automatique(apprentissage, OCR, interrogation automatique de la préfecture à partir de la plaque d'immatriculation)
<b>Plaque d'immatriculation</b>	Chaîne de caractères	Automatique (détecteur de plaques plus OCR)
<b>Signes distinctifs : carrosserie accidentée, autocollants</b>	Liste de paires : forme-texte	Manuelle, (semi-)automatique (détecteur de formes, apprentissage, OCR)

TABLE 3.13 – Métadonnées génériques associées aux objets de type véhicule

Libelle	Commentaire	Modalité de renseignement
<b>Identifiant</b>	Chaîne de caractères	Automatique (Attribué à chaque nouveau événement ou utilisation d'un dictionnaire ou d'une ontologie [SanMiguel et al., 2009])
<b>Date début/fin</b>	Date	Manuelle, (semi-)automatique (Détecteur)
<b>Type/Description</b>	Chaîne de caractères (agression, vol, bagarre, présence, regroupement, personne qui court)	Manuelle, (Semi-) Automatique (Utilisation d'un dictionnaire ou d'une ontologie)
<b>Objets impliqués</b>	Chaîne de caractères (identifiants des objets impliqués)	Manuelle, (Semi-) Automatique

TABLE 3.14 – Métadonnées génériques associées aux événements

## 3.5 Norme ISO 22311

La norme ISO 22311 vise à faciliter l'interopérabilité des systèmes de vidéosurveillance en définissant un format d'export des données et des métadonnées issues de tels systèmes.

### 3.5.1 Structuration des données vidéo

Les données issues d'un système de vidéo surveillance doivent être structurées selon le format de la Figure 3.7.

Les données sont organisées de manière hiérarchique dans des fichiers, dossiers et groupes de dossiers. Le rangement en fichiers et dossiers est réalisé conformément à des créneaux temporels de répertoire (DTS) et créneaux temporels de fichiers (FTS) (le temps est coordonné utilisant le temps universel (UTC)). Chaque fichier contient des données (audio/video) provenant de plusieurs sources (caméras), un index par contenu (audio/video) permettant l'accès précis à toute image et tout moment (de la vidéo) spécifiques et des métadonnées pour chaque source

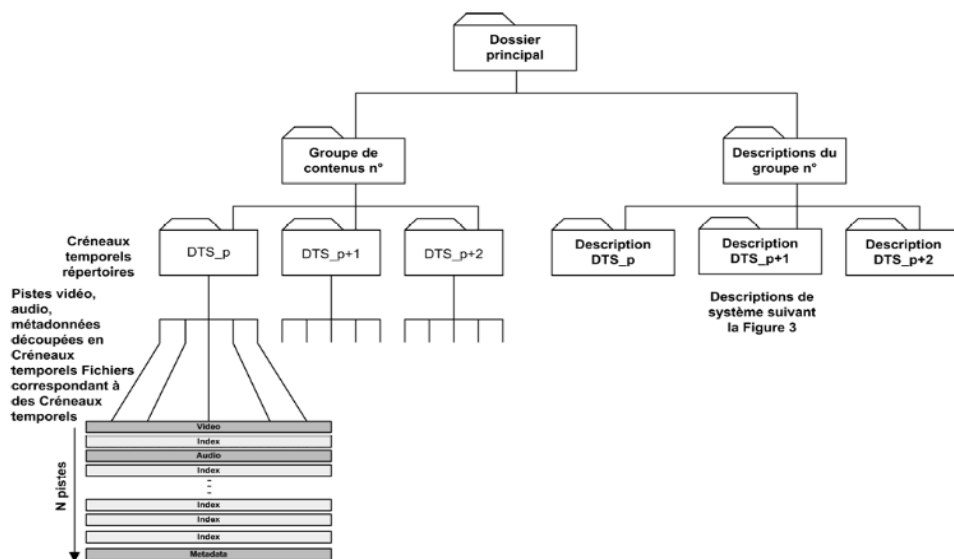


FIGURE 3.7 – Format du paquet audio-vidéo exporté d’un système de vidéosurveillance

et dossier (Figure 3.8). Les fichiers vidéos de chaque dossier sont nommés selon la règle :  $\langle \text{track\_nom} \rangle\_video\_ \langle \text{type\_of\_encoding} \rangle\_ \langle \text{timestamp} \rangle\_ \langle \text{extension} \rangle$ , où  $\langle \text{track\_nom} \rangle$  est le nom de la piste vidéo,  $\langle \text{type\_of\_encoding} \rangle$  désigne le type d’encodage de la vidéo,  $\langle \text{timestamp} \rangle$  représente l’estampille temporelle du début de la vidéo et  $\langle \text{extension} \rangle$  est l’extension du fichier.

Pour un minimum d’interopérabilité (niveau 1 de la norme), les fichiers de description pour chaque dossier sont des fichiers XML et contiennent les éléments suivants :

- La liste des FTS avec leur heure de début
- ID/Détails de chaque source (caméra)
  - Type : video/audio/métadonnées
  - Codec
  - Détails relatifs au contenu (nombre de pixels de l’image, échantillonnage de l’audio, etc.)
  - Type d’encapsulation (formats spécifiés dans la norme)
- Données de localisation de la scène observée
  - Localisation du centre de la scène (coord GPS)
  - Positions des coins de la scène (coord GPS)

Ces éléments ne sont pas suffisants et pour une exploitation plus rapide et efficace des contenus vidéo, la norme propose des exigences relatives aux métadonnées supplémentaires. Dans la section suivante nous présentons les dictionnaires de métadonnées (concernant les capteurs et les événements) et leur structuration.

### **3.5.2 Dictionnaire de métadonnées**

La norme propose des dictionnaires non exhaustifs et extensibles pour les descriptions des capteurs et des événements structurés conformément aux schémas présentés dans les figures 3.9 et 3.10. La description des capteurs (voir Figure 3.9) comprend une partie statique qui contient des éléments de description générale (ID et modèle du dispositif, informations sur la plate-forme transportant le dispositif, etc.) et une partie dynamique qui contient : l'information temporelle, la localisation du dispositif (absolue et relative), des informations concernant le champ de vue du capteur (distance focale, angle horizontal et vertical), des informations sur la scène observée par chaque capteur et des "champs libres" (le champ "Métadonnées expérimentales"). Les métadonnées liées au capteur (à la caméra) proposées par la norme mettent en évidence non seulement le besoin d'interopérabilité mais aussi le fait que la numérisation des systèmes de vidéosurveillance rend accessible une multitude de caractéristiques qui peuvent être utilisées aussi pour des objectifs de recherche et filtrage du contenu que pour du paramétrage du matériel et contrôle du système. Cependant, le dictionnaire des métadonnées concernant les capteurs proposé par la norme reste incomplet (e.g., il n'y a pas de descripteurs de qualité d'image) et ne contient pas d'éléments liés à la mise en place ou à la façon dont les métadonnées seront renseignées ou extraites. Par exemple, chaque caméra se voit associée une localisation absolue (latitude, longitude, altitude) et une localisation relative (X,Y,Z) qui contient les coordonnées de translation de la caméra par rapport à une position absolue de référence mais le dictionnaire de la norme ne donne pas d'éléments concernant les systèmes de référence qui sont utiles pour définir des positions symboliques utilisées spécialement en milieu indoor. En plus, la norme ne parle pas de la façon dont ces localisations seront extraites.

La description des événements (voir Figure 3.10) est dynamique, elle peut être représentée par rapport à un dictionnaire d'événements (e.g., une ontologie) et doit contenir la délimitation temporelle et spatiale. On peut également ajouter des nouveaux éléments dans le champ libre "Métadonnées expérimentales".



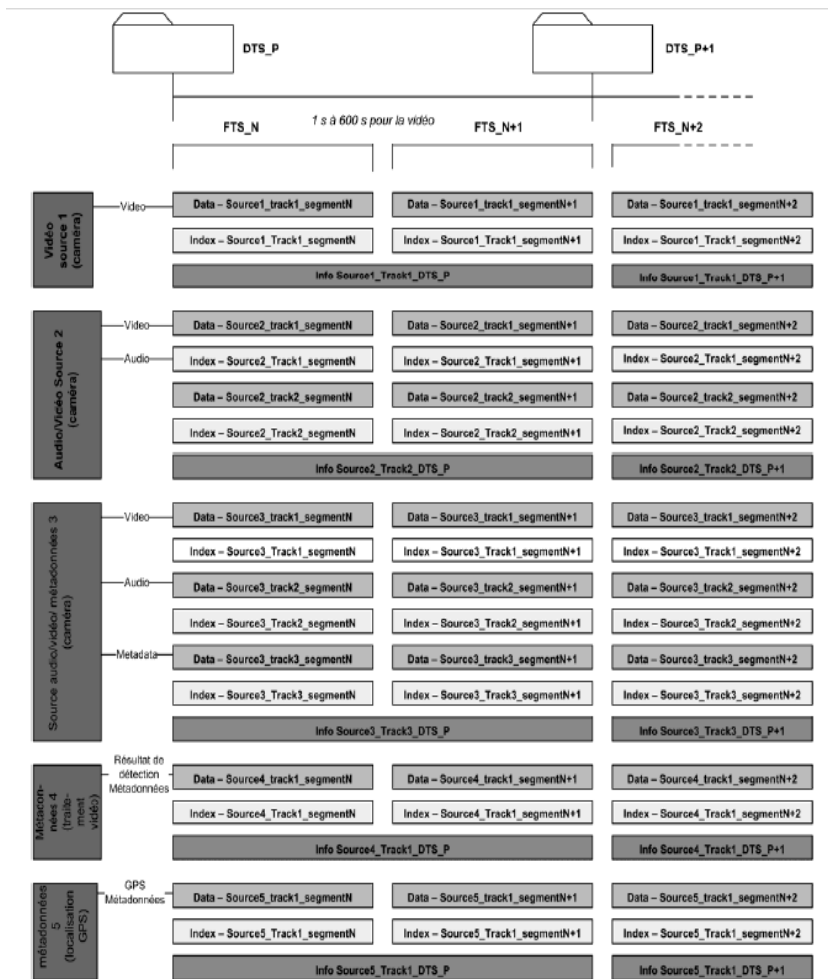


FIGURE 3.8 – Structure données, index et métadonnées dans les DTS et FTS

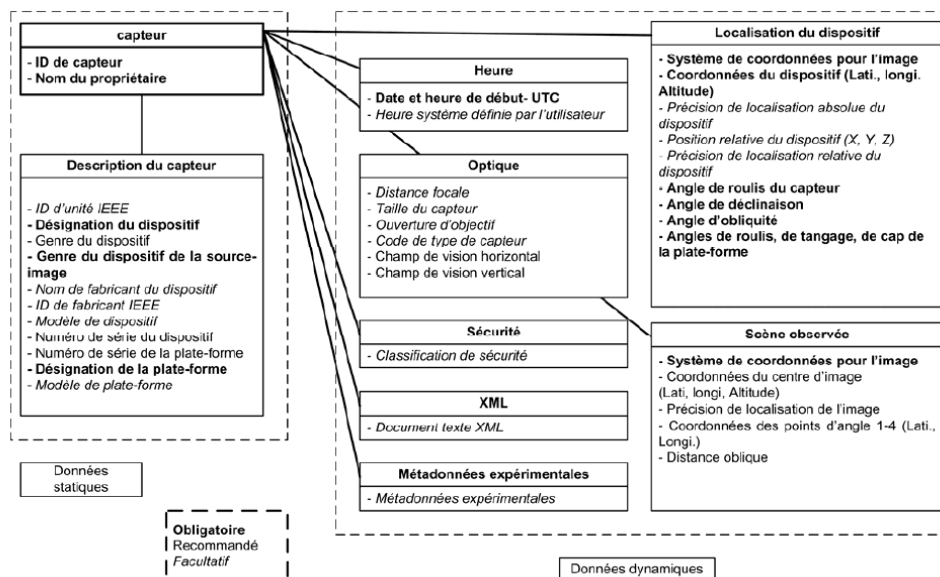


FIGURE 3.9 – Description du capteur

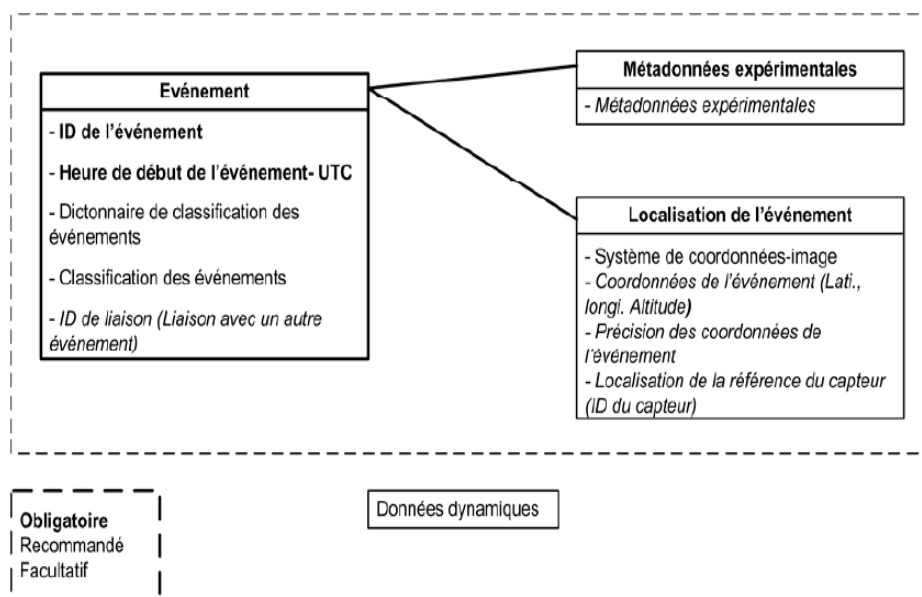


FIGURE 3.10 – Description de l'événement

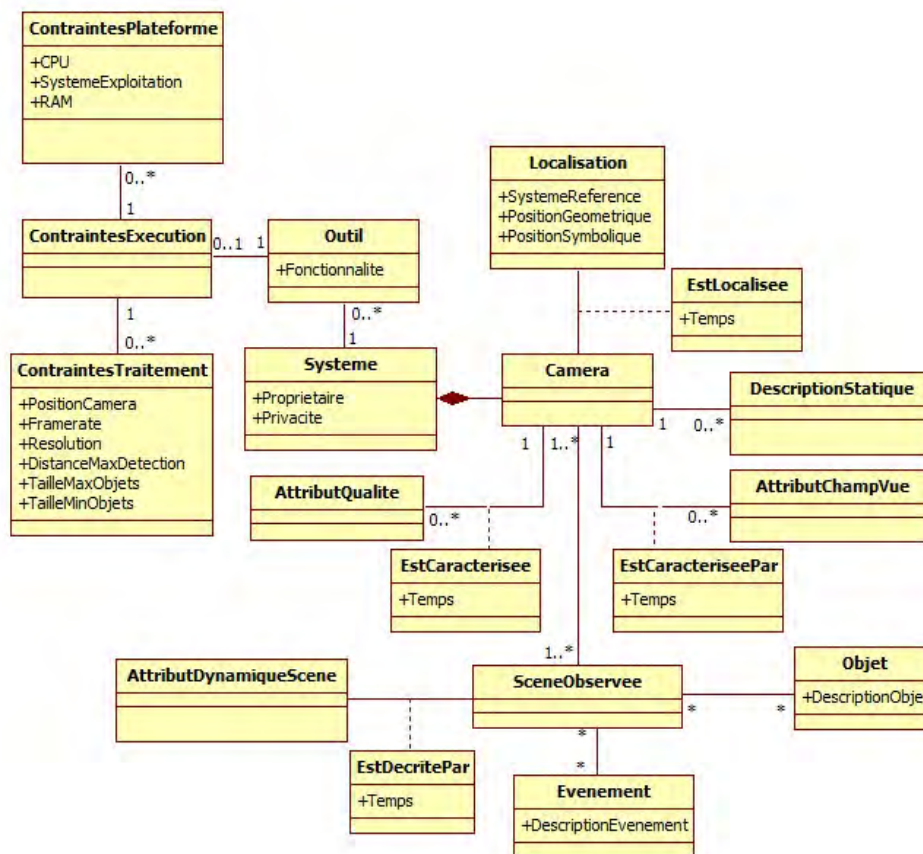


FIGURE 3.11 – Méta modèle de métadonnées vidéo pour la vidéosurveillance

### 3.6 Proposition d'un méta modèle conforme à la norme

Le méta modèle représente un "schéma" générique qui est conforme à la norme 22311 et qui offre une structure pour le dictionnaire proposé dans la section 3.4. En fonction du besoin (e.g., tâche d'installation des caméras, de contrôle de la qualité des images issues d'un systèmes, de recherche des extraits vidéos d'une personne ou d'une voiture d'intérêt dans le cadre d'une enquête) des instances du méta modèle seront générés dans des modèles de métadonnées. La façon dont un méta-modèle contraint la définition des modèles est similaire à celle dont la grammaire d'un langage de programmation contraint le développement du code dans ce langage. Grâce au

méta modèle, les métadonnées générées respectent et enrichissent la structure de la norme ISO 22311 pour favoriser l'interopérabilité ayant comme but final de définir un format générique de stockage des contenus et des descriptions des contenus vidéos issues des systèmes de vidéosurveillance. Dans la Figure 3.11 nous présentons le méta format d'annotation vidéo qui respecte, utilise et rajoute des éléments par rapport à la norme. En plus, comme perspective nous proposons d'enrichir le dictionnaire de métadonnées de la norme avec plus d'éléments pertinents pour la description des collections vidéos et concernant la description du réseau de caméras.

La description du système est associée à plusieurs annotations caméras. Pour chaque caméra (capteur dans la norme) on renseigne des informations techniques statiques et dynamiques (e.g., le modèle, le codec, la qualité de l'enregistrement), la localisation (par rapport à différents systèmes de référence) et on associe une scène observée, en décrivant les caractéristiques dynamiques de la scène (e.g., le type de scène, le contexte de la scène). Dans les vidéos correspondant à chaque scène on va annoter des objets et des événements (frame par frame (e.g., format PETS) ou par instance d'objet ou événement (e.g., format Viper GT)). Les classes d'association modélisent la dynamique de certains éléments qui seront représentés par rapport au temps.

## **3.7 Conclusion**

L'analyse du besoin effectuée dans le cadre du projet METHODEO a montré l'évolution des systèmes de vidéosurveillance et l'augmentation du nombre de paramètres liés notamment aux caméras qui composent de tels systèmes qui doivent être configurés, vérifiés ou pris en compte dans des tâches comme la recherche et le filtrage des contenus, l'installation, la mise en place et le contrôle des systèmes.

Notre contribution dans le cadre du projet a consisté dans la réalisation d'une étude des métadonnées utiles pour le contexte de la vidéosurveillance et des différentes tâches liées à l'installation, à l'exploitation, au contrôle et à la favorisation de l'interopérabilité de tels systèmes. Suite à cette étude, nous avons proposé un dictionnaire de métadonnées qui contient une description générique du chaque élément et des notions concernant sa mutabilité et la façon dont il peut être renseigné. Une structuration de ce dictionnaire est proposée suivant une déclinaison des métadonnées en trois grandes catégories : (1) métadonnées liées au système de vidéosurveillance (métadonnées génériques, métadonnées liées à la caméra et métadonnées liées aux outils d'analyse vidéo), (2) métadonnées liées à la scène, (3) métadonnées liées au scénario (aux objets et aux événements qui composent la scène).

L'approche de méta-modélisation des métadonnées vidéo que nous proposons se positionne sur un même plan que les standards et les normes qui cherchent à définir des éléments et des relations entre ces éléments génériques pour constituer des descriptions des contenus vidéo. C'est important que le méta modèle reste extensible (les utilisateurs puissent définir des nouveaux éléments). Notre proposition de méta modèle utilise des éléments de la norme et assurera le fait que toutes les instances de modèles de métadonnées seront conformes avec la norme ISO 22311. Suite aux résultats obtenus sur le projet METHODEO, plusieurs propositions de projets ont été faites qui visent à faire évoluer la norme ISO22311 et à donner des éléments de généralisation de la mise en place de celle ci.



# Chapitre 4

## Interrogation des métadonnées de vidéosurveillance à partir des trajectoires hybrides

### 4.1 Introduction

Dans le premier chapitre de ce mémoire nous avons illustré le contexte de la vidéosurveillance par des exemples de cas d'utilisation. Ces cas d'utilisation montrent que les principales requêtes dans ce domaine sont exprimées sous forme de trajectoires, trajectoires dont les segments sont décrits par des positions géométriques (des points) ou par des positions symboliques (descriptions qui peuvent être ramenées à un point ou une géométrie) par rapport à des différents systèmes de référence outdoor ou indoor comme le système géodésique, le réseau routier ou de transport, le réseau de lecteurs de ICCARD, etc. Le but final de toute "requête" envoyée à un système qui gère un réseau de vidéosurveillance est d'arriver à trouver les séquences vidéos (d'une qualité d'image suffisante) qui contiennent les objets (e.g., personne, véhicule, bagage abandonné) ou les événements cibles. L'espace de recherche étant énorme (parfois des dizaines d'heures de vidéo) et l'objet ou l'événement cible pas bien défini (au moins au début), le développement des outils de filtrage "négatif" des vidéos (en se basant sur différents critères) et d'assistance des opérateurs dans leur tâche représente une préoccupation importante des commerciaux et des chercheurs intéressés par ce domaine. L'étude de ces cas d'utilisation a aussi montré une double mobilité dans le cadre des systèmes, des objets cibles et des caméras, mobilité qui

est contrainte par le réseau routier et celui de transport en environnement outdoor et par la structure du bâtiment en environnement indoor.

Tous ces éléments nous ont amené à reformuler le problème de filtrage de contenus vidéos en un problème de filtrage spatio-temporel des collections vidéo en se basant sur l'interrogation à partir des requêtes spatio-temporelles des métadonnées associées aux contenus vidéo. L'utilité des requêtes spatio-temporelles comme les requêtes de localisation (position queries), les requêtes de zone (region queries), les requêtes de recherche des K voisins les plus proches (K nearest neighbours (KNN) queries) a été montrée par la multitude des travaux de recherche et commerciaux existants (voir le chapitre état de l'art). Toutefois, les travaux qui considèrent des requêtes basées sur des trajectoires sont beaucoup moins nombreux. A notre connaissance il n'y a pas d'approche basée sur une trajectoire composée de segments hétérogènes (géométriques et symboliques exprimés par rapport à des différents systèmes de référence). En conséquence nous proposons un framework qui permet d'interroger les métadonnées spatio-temporelles associées aux caméras de vidéosurveillance à partir des requêtes trajectoires hybrides. A part le nouveau concept de trajectoire hybride que nous avons défini, notre approche est originale par rapport aux très peu de travaux qui ont proposé de tels frameworks car :

- comme montré dans la section 2.6 du chapitre 2 les approches existantes ne proposent pas de modèles de données qui intègrent des informations concernant tous les éléments auxquels nous nous intéressons : réseau routier, réseau de transport, objets mobiles et caméras ;
- les approches existantes proposent des solutions qui supposent une étape de traitement du réseau routier ou de transport pour un but de stockage en local, cela ne favorise pas l'interopérabilité et suppose une charge de travail en amont non négligeable ;
- dans la plupart des travaux existants les géométries des champs de vue des caméras sont construites (au moment de l'entrée dans le système) pour chaque frame et sont stockées telles quelles ; dans le cas d'un système de vidéosurveillance cela peut vite apporter une surcharge importante ;

Nous nous sommes concentrés dans ce chapitre sur le traitement des trajectoires des objets mobiles dans un environnement outdoor. Dans une perspective de définition d'un modèle de données qui puisse supporter la représentation de façon uniforme pour l'utilisateur d'une trajectoire hybride indoor-outdoor, un travail en cours concernant un modèle de trajectoire hybride indoor est présenté dans l'annexe A. Dans ce but, nous avons défini le concept de *trajectoire hybride*. Dans la suite de ce chapitre nous allons présenter en détail la notion de trajectoire, segment de trajectoire et



celle de requête trajectoire hybride. Ensuite nous allons présenter notre modèle de données dans le contexte de la vidéosurveillance. Finalement nous allons détailler les opérateurs que nous avons définis et qui permettent de sélectionner de façon automatiques les caméras susceptibles d'avoir filmé une certaine trajectoire.

## **4.2 Interrogation des données spatio-temporelles à partir d'une trajectoire composée de segments géométriques et/ou symboliques**

Comme présenté dans le chapitre d'état de l'art, il y a deux façons d'exprimer la position d'un objet par rapport à des différents systèmes de référence : géométrique (coordonnées par rapport à un système de référence global ou local) et symbolique (description plus sémantique par rapport à des points d'intérêt, réseaux de transport, architecture d'un bâtiment etc.). Au moment du traitement de ces positions, traitement qui consiste dans la transformation des positions afin de tout exprimer par rapport à une référence commune, la principale différence entre les deux types de positions est représentée par la précision, une position géométrique pouvant être exprimée par un point tandis qu'une position symbolique peut être représentée par la géométrie d'une pièce d'un bâtiment ou celle d'une cellule RFID. Nous allons définir dans notre travail le concept original de *trajectoire hybride*, qui sera une séquence de segments de trajectoire dont chaque segment peut être composé de positions géométriques ou de positions symboliques. Nous allons illustrer ce concept à partir d'un cas d'utilisation du domaine de la vidéosurveillance que nous présentons dans la section suivante en faisant ressortir les défis qu'un système gérant ces cas devrait relever.

### **4.2.1 Cas d'utilisation de la vidéosurveillance**

#### **Contexte outdoor**

Prenons l'exemple suivant illustré dans la Figure 4.1. Une femme se fait voler son portefeuille dans les rues de Toulouse. La dernière fois qu'elle a vu son portefeuille était juste avant de prendre le bus numéro 16 de la station Jeanne d'Arc. La dame se rappelle l'heure exacte (10 :19) car elle avait regardé les horaires du bus affichés dans la station. Elle est descendue à la station Jean Jaurès et elle a marché vers la

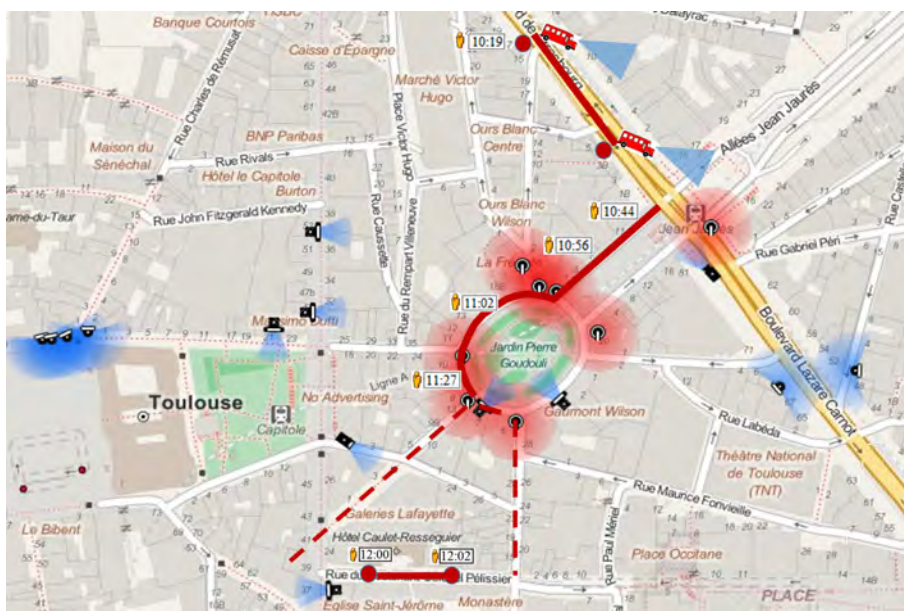


FIGURE 4.1 – Exemple de trajectoire reconstituée par les policiers

Place Wilson. Le smartphone (un Iphone 4S) de la personne avait activé le service de localisation ce qui a permis aux enquêteurs de récupérer ses coordonnées du moment où elle est descendue du bus jusqu'au moment où elle s'est rendue compte que son portefeuille a disparu (environ une heure plus tard). Son téléphone est resté sans batterie après. La femme se rappelle qu'un jeune très suspect s'était approché d'elle dans le bus. Le jeune est descendu du bus, et elle se rappelle l'avoir vu deux fois sur la Rue du Lieutenant Colonel Pélissie, devant l'entrée des Galleries Lafayette (numéro 5) et devant l'entrée dans un autre magasin situé au numéro 10 dans son trajet vers l'Espace Saint Georges. La trajectoire illustrée dans la Figure 4.1 contient des segments de trajectoire géométriques (les coordonnées récupérées du dispositif) et des segments symboliques (exprimés par rapport au réseau de transport en commun et par rapport au réseau de rues de la ville). La ligne marquée en pointillée sur la figure marque un "trou" dans la trajectoire (la victime ne se rappelle plus avec exactitude son trajet et son téléphone avait cessé de fonctionner). Les trajectoires possibles de la personne pendant le "trou" peuvent être calculées en se basant sur le réseau routier ou le réseau de transport.

Le principal défi relevé par la démarche de développement d'un système qui puisse traiter une trajectoire comme celle illustrée dans la Figure 4.1 est de gérer l'hétérogénéité des différents segments. Cette hétérogénéité provient de deux sources :

(1) le type de la position (géométrique vs. symbolique) et (2) le système de référence (e.g.,  $P1=(43.561,1.468)$  et  $P2=14$  sont deux positions géométriques mais la première est exprimée par rapport au système géodésique (WGS84) et la deuxième est exprimée par rapport à un système de référence local associé à une rue).

## Contexte indoor

Un autre cas d'utilisation très populaire de nos jours est celui d'un bâtiment "intelligent". Chaque utilisateur a un smartphone qui est suivi par des systèmes de localisation basés sur différents capteurs de localisation (e.g., RFID, WiFi). D'autres types de capteurs surveillent la consommation d'énergie et la température dans les différentes parties du bâtiment. La Figure 4.2 montre un exemple de trajectoire d'une personne à l'intérieur du bâtiment. La trajectoire est composée de quatre segments ( $u_1, u_2, u_3, u_4$ ). Il est possible d'avoir une perte d'information de localisation pour une période de temps donnée (une personne a fermé son smartphone et aucune information de localisation n'existe). La ligne en pointillés représente un "trou" dans la définition d'une trajectoire qui peut être complétée utilisant le plan du bâtiment (la représentation des différentes pièces et les connexions qui existent entre celles-ci) [Jensen et al., 2009]. En se basant sur cette trajectoire, on veut appliquer différents opérateurs comme l'agrégation de la consommation de l'énergie sur la trajectoire ou l'identification des capteurs situés sur la trajectoire ou à une certaine distance de la trajectoire.

Un autre exemple de scénario est le suivant. Il y a une alarme qui se produit dans une aire du bâtiment. La trajectoire de la personne qui se trouvait au moment de l'incident dans la zone respective est générée en se basant sur la localisation de son smartphone détectée par les cellules Wifi situées dans les alentours. Cela génère des coordonnées (x,y) exprimées par rapport au plan du bâtiment. A un moment donné le signal de son smartphone est perdu et la seule information que nous avons sur la personne que nous avons est qu'elle a été détectée par un lecteur de cartes (elle a badgé en entrant dans une salle). Dans ce cas la trajectoire de la personne va être celle de la figure 4.3. Les premiers deux segments de la trajectoire sont  $u_1$  et  $u_2$ , il y a une période où la localisation est inconnue et la dernière partie de la trajectoire est  $u_3$  qui est représentée par la géométrie de la chambre  $C_1$ . Dans ce cas, nous nous intéressons à trouver les caméras vidéo qui ont pu filmer sa trajectoire afin de visualiser les enregistrements.

Comme dans le cas du cas d'utilisation en environnement outdoor, le défi à surmonter

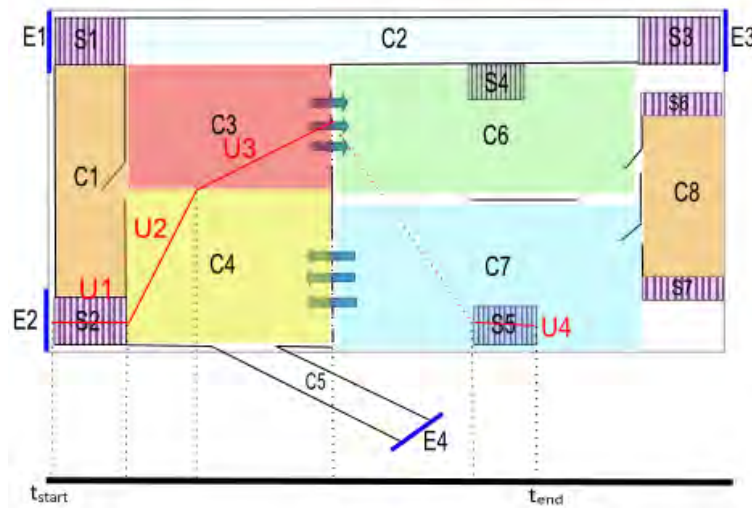


FIGURE 4.2 – Exemple de trajectoire à l'intérieur d'un bâtiment

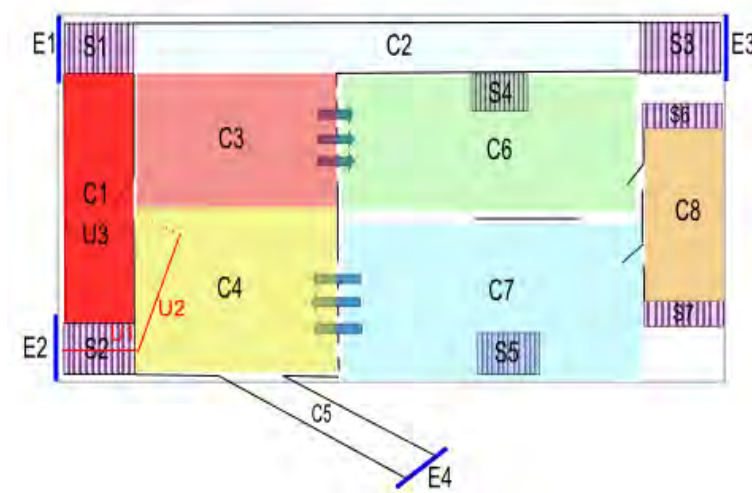


FIGURE 4.3 – Exemple de trajectoire hybride

réside dans l'hétérogénéité des types de positions à traiter : le système de localisation basé sur les cellules Wi-Fi donne des positions précises (des points) tandis que les positions exprimées par rapport au réseau de lecteurs ICCARD peuvent être ramenées à l'aire d'une pièce du bâtiment.

Les cas d'utilisation présentés relèvent les besoins suivants concernant l'approche de modélisation de données (e.g., réseau routier, plan du bâtiment, trajectoire) et les algorithmes à développer pour l'interrogation de ces données :

- le modèle de données :
  - doit pouvoir supporter plusieurs systèmes de référence par rapport auxquels on peut exprimer aussi des positions géométriques que des positions symboliques ;
  - doit être assez générique pour pouvoir s'appliquer aussi au contexte indoor que outdoor afin de pouvoir envisager une trajectoire uniforme outdoor-indoor dont le traitement soit inconnu à l'utilisateur ;
  - doit être un modèle d'encapsulation qui se base sur des standards afin d'assurer une interopérabilité des différents systèmes qui gèrent des réseaux de vidéosurveillance.
- les algorithmes :
  - doivent se baser sur des techniques en deux pas (filtrage/affinage) afin de pouvoir répondre au besoin de réponse rapide (même si moins précise) dans une situation d'urgence ;
  - doivent prendre en entrée des segments de trajectoires de l'utilisateur ;
  - doivent être suffisamment rapides pour répondre au besoin opérationnel.

A notre connaissance, les approches existantes de modélisation de trajectoires traitent des trajectoires géométriques ou symboliques mais il n'y a pas d'approche qui considère des trajectoires hybrides. Par la suite nous allons donner la définition et le template d'une requête trajectoire hybride. Dans ce chapitre nous allons nous concentrer sur le contexte outdoor.

## 4.2.2 Requête trajectoire hybride

A partir des exemples présentés nous donnons la définition d'une *requête trajectoire hybride*. Comme expliqué dans l'état de l'art une trajectoire est une séquence de segments de trajectoire et un intervalle de temps. Un segment ( $u_k$ ) de trajectoire est défini comme une séquence de positions. L'innovation que nous introduisons dans la définition d'une trajectoire est la prise en compte du fait que chaque position est exprimée par rapport à un système de référence (e.g., géodésique, réseau routier).

Les positions sont homogènes à l'intérieur du segment (exprimées par rapport à un seul système de référence) mais les différents segments peuvent contenir des positions hétérogènes comme dans les exemples présentés dans la section 4.2.1.

On définit une requête trajectoire hybride comme un tuple illustré ci dessous :

$$Tr = (tr_{id}, \{u_k\}, [t_{start}, t_{end}]) \quad (4.1)$$

ou  $tr_{id}$  est l'identifiant de la trajectoire,  $\{u_k\}$  est l'ensemble des segments spatiaux constituant la trajectoire et  $[t_{start}, t_{end}]$  représente l'intervalle de temps de la requête trajectoire.

La définition d'un segment de trajectoire est la suivante :

$$u_k = (u_{id}, refS_{id}, \{position_i\}) \quad (4.2)$$

ou  $u_{id}$  est l'identifiant du segment,  $refS_{id}$  est l'identifiant du système de référence par rapport auquel les positions sont définies et  $\{position_i\}$  est l'ensemble des positions qui compose le segment de trajectoire.

Pour pouvoir gérer la hétérogénéité des segments de trajectoire nous définissons la position d'un objet comme étant composée de deux grandes parties : la partie qui désigne le système de référence et la partie qui désigne la localisation par rapport au système de référence. L'idée se rassemble avec celle proposée par [Jiang and Steenkiste, 2002] qui définissent la syntaxe d'une localisation indoor hybride basée sur la syntaxe d'un URI (Universal Resource Identifier). Toutefois la notion de chemin qu'ils utilisent comprend juste l'aspect hiérarchique (une chambre fait partie d'une aile d'un bâtiment) et non pas celui de trajectoire (qui implique la connectivité directe entre les points successifs). Aussi ils ne considèrent pas différents types de systèmes de référence mais seulement le plan d'un bâtiment.

A partir des observations faites nous présentons un exemple d'une *requête trajectoire* (voir figure 4.4) contenant des segments géométriques et symboliques. Chaque requête trajectoire hybride a une partie spatiale et une partie temporelle. La partie spatiale est constituée d'une séquence de segments, chaque segment étant constitué d'une partie désignant le système de référence et une séquence de positions (géométriques ou symboliques) exprimées par rapport au système de référence correspondant. Les Figures 4.4 et 4.5 présentent des exemples de requêtes trajectoires hybrides pour l'environnement outdoor et respectivement indoor. Dans les deux exemples nous

```
{
  "query": {
    "spatial": [
      {
        "WGS84": [
          [43.6060, 1.44837],
          [43.6067, 1.44758],
          [43.6071, 1.44721]
        ]
      },
      {
        "RRTLSE": [
          {"Allées du Président Franklin Roosevelt":9}, {"14 Allées du Président Franklin Roosevelt":9}
        ]
      },
      {
        "RTTLSE": [
          {"Bus16": "Jean Jaures"}, {"Bus16": "Jean d'Arc"}
        ]
      }
    ],
    "temporal": {
      "start": "07-02-2014 14:00:00",
      "end": "07-02-2014 14:30:00"
    }
  }
}
```

FIGURE 4.4 – Exemple distanciation du template d’une requête trajectoire hybride extérieure

pouvons remarquer la structuration de la trajectoire dans une partie spatiale et une temporelle et la division de la partie spatiale en segments constitués des positions exprimés par différents systèmes de référence dont les identifiants doivent être connus par le système d’information qui implémente le modèle de données (e.g., WGS84 - système géodésique, RRTLSE - réseau routier de Toulouse, RTTLSE - réseau de transport de Toulouse).

Le besoin final identifié dans le cadre des systèmes de vidéosurveillance et la perspective finale de notre travail de recherche est de développer un framework qui permette le traitement de façon homogène des trajectoires qui contiennent des segments outdoor, indoor, géométriques et symboliques. Dans notre travail nous nous sommes concentrés sur le traitement des trajectoires outdoor comportant un mix de segments géométriques et symboliques. Une analyse similaire et une définition d’une trajectoire hybride indoor peuvent être trouvées dans l’annexe A. Par la suite, nous allons présenter le modèle de données utilisé pour répondre à la requête trajectoire hybride en environnement outdoor.

```
{
  "query": {
    "spatial": [
      {
        "Floor8": [
          [50.21627, 433.53088],
          [65.84485, 409.75342],
          [60.56707, 441.10219],
          [93.67555, 371.18614],
          [97.67467, 389.67915]
        ]
      },
      {
        "ICCARD": [0000110015010005]
      }
    ],
    "temporal": {
      "start": "10-02-14 12:55:34",
      "end": "10-02-2014 13:20:00"
    }
  }
}
```

FIGURE 4.5 – Template d’une requête trajectoire hybride intérieure

### 4.3 Modèle multi couches

L’idée de notre recherche est de définir un framework d’interrogation des données spatio-temporelles issues des différents capteurs et exprimées par rapport au réseau routier et au réseau de transport pour un filtrage des contenus vidéos en se basant sur une requête trajectoire hybride. Jusqu’à ce point dans ce chapitre nous avons présenté le type de requête supporté par notre framework. Par la suite nous allons décrire le modèle de données sur lequel nous allons nous appuyer pour répondre à la requête trajectoire hybride. Un grand nombre de travaux ont proposé des modèles de localisation géométriques et symboliques pour des environnements outdoor qui ont prouvé leur efficacité (voir section 2.3 du chapitre 2). En conséquence, l’idée n’est pas de présenter un modèle complètement nouveau, mais plutôt de définir un modèle qui intègre ou étend des modèles existants.

Les informations que nous proposons d’intégrer dans un modèle qui sera ensuite utilisé pour le développement des outils de gestion des collections vidéo (e.g., des outils d’annotation, de filtrage, de recherche) sont :

- Les positions des caméras fixes et le changement de ces positions dans le temps pour les caméras mobiles ainsi que l’attachement des caméras mobiles à des objets



- mobiles (e.g., caméras installées dans des bus) ;
- La géométrie de la scène observée par une certaine caméra. Cette scène est calculée à partir des caractéristiques optiques et d’installation de la caméra comme expliqué dans la section 3.3 du chapitre 3. La géométrie de la scène ”vue” par la caméra installée à une position donnée à un instant donné représente une information importante qui peut être utilisée dans beaucoup d’applications ;
- Le réseau routier et le réseau de transport en commun d’une ville qui sont des contextes très importants pour le domaine de la vidéosurveillance car les deux représentent des systèmes de références par rapport auxquels des localisations sont exprimées ;
- L’information temporelle qui combinée avec la localisation représentent les principales informations sur lesquelles s’appuient les domaines d’application comme la vidéosurveillance ou l’analyse du trafic routier.

Dans notre approche nous proposons un modèle à quatre couches [Codreanu et al., 2013] : (1) Réseau routier, (2) Réseau de transport, (3) Objets, (4) Caméras. Dans les sections suivantes, nous allons détailler chaque couche. Pour chaque ”couche” de notre modèle nous allons présenter une description, la définition formelle des données, les fonctions de mapping qui permettent de transformer des données d’une couche à une autre ou de représenter toutes les données par rapport à un système de référence commun.

### 4.3.1 Couche Réseau Routier

#### Description

La modélisation d’un *Réseau Routier* est relativement simple. Étant donné un réseau routier du monde réel, sa modélisation dans un plan 2D est donnée par une séquence ordonnée de points situés sur le réseau qui ont des positions par rapport au système géodésique ( $p = \langle \text{lat}, \text{long} \rangle$ ). Chaque deux points déterminent une ligne. Toutes ces lignes donnent une approximation de la forme réelle du réseau de rues dont la précision dépend du nombre de points pris en compte. En considérant un niveau de granularité supérieur, à cette représentation correspond une modélisation sous forme de graphe consacrée dans la littérature [Liu et al., 2012], [Jensen et al., 2003]. Les noeuds sont des points qui ont des positions 2D par rapport au système géodésique ( $\langle \text{lat}, \text{long} \rangle$ ). Chaque deux noeuds déterminent un segment et plusieurs segments composent une rue. Les points de croisement des rues sont des intersections. Cette modélisation permet de garder plusieurs niveaux de granularité du réseau routier



FIGURE 4.6 – Exemple de vecteur de points décrivant la forme d’une route

[Sandu Popa et al., 2011]. De cette façon, on peut considérer comme arrête du graphe chaque segment de rue, les portions de rue entre les grandes intersections ou les rues entières. La dernière option est la plus proche de la façon dont les adresses sont exprimées dans la vie réelle (e.g., 15 Rue Alsace Lorraine, Toulouse, France).

Nous allons définir les deux types de positions : une position géométrique qui est une position 2D par rapport au système géodésique (des coordonnées GPS  $\langle \text{lat}, \text{long} \rangle$ ) et une position symbolique relative au réseau routier qui est une position similaire à l’adresse postale. Il existe des fonctions et des API publiques qui permettent de passer d’un type de position à un autre (les dénominations utilisées par Google sont Geocode et Reverse Geocode). Une définition formelle de ces fonctions de mapping se trouve par la suite dans cette section (paragraphe Définition des données).

Les données qui composent cette couche peuvent ou non être stockées dans une base de données locale. Il existe des travaux qui transforment tout le réseau routier sous forme de graphe et le stockent dans une base de donnée locale (e.g., [Brinkhoff, 2002]) et des travaux qui utilisent directement des bases de données extérieures comme OpenstreetMaps ([Shen et al., 2011]).

L’avantage de l’utilisation d’une base de donnée externe est le fait de pouvoir extraire

la partie du graphe concernée par une requête spécifique sans devoir se soucier de problèmes tels que la maintenance ou l'optimisation de la base. Par exemple, l'API Google Maps permet de récupérer via son service Directions les points qui donnent l'approximation d'une certaine route comme illustré dans la Figure 4.6. Un extrait du vecteur des points est illustré dans la Figure 4.6 en format JSON tel que retourné par l'API Google Maps.

Dans notre implémentation du modèle nous avons choisi d'utiliser l'API Google Maps V3<sup>1</sup> afin d'interroger le réseau routier. L'API Google Maps offre des informations assez complètes et est facile à utiliser. Comme nous n'avons pas besoin d'autres types d'informations tels que les noms ou l'aire des bâtiments ou d'autres types d'objets (qui sont mis à disposition par d'autres outils comme Open Street Maps), notre choix s'est avéré suffisant. Néanmoins, d'autres solutions externes peuvent être envisageables et ne modifient ni le modèle envisagé ni l'expérimentation réalisée.

### Définition des données

*Définition 1* : Nous définissons le réseau routier à partir d'une fonction linéaire par morceaux (piecewise linear function)  $f : V \rightarrow R^2$ . Pour chaque point du vecteur retourné par la requête faite à Google Maps, un point d'interpolation est ajouté dans la définition de la fonction. L'interpolation linéaire consiste plus concrètement à "joindre" les points dans l'ordre. Nous allons noter l'ensemble de positions appartenant au réseau routier par  $R_R$ .

*Définition 2* : Nous définissons la fonction de mapping  $map0 : \text{positionsGPS} \rightarrow R_R$  qui donne la position sur réseau routier  $R_R$  (e.g., 14 Rue Montesquieu) à partir d'une position GPS. Nous définissons aussi la fonction inverse  $map0^{-1} : R_R \rightarrow \text{positionsGPS}$  (Reverse Geocoding de Google Maps).

## 4.3.2 Couche Réseau de Transport

### Description

Si la représentation du réseau routier est assez simple et canonique dans la littérature, pour le réseau de transport en commun les choses sont compliquées par les contraintes

---

1. <https://developers.google.com/maps/documentation/javascript/>

spatiales (les trajets prédéfinis) et temporelles (les horaires) imposées pour chaque trajet de chaque moyen de transport.

Afin de définir la modélisation du réseau de transport en commun nous nous sommes appuyés sur le format GTFS [Nick Kizoom, 2008]. C'est le format de description le plus utilisé par les agences assurant les services des transport et aussi par les fournisseurs d'applications logicielles de types GIS (Geographic information System) ou de calcul de trajets multimodaux. Dans ce qui suit nous allons présenter la formalisation du modèle de données utilisé pour le réseau de transport avec les enrichissements que nous avons apportés. Comme dans cette thèse nous nous sommes intéressées qu'au réseau de bus, par la suite nous allons nous référer que à ce-ci.

Le *Réseau de Transport* d'une ville est composé par un réseau de lignes de transport qui déservent des stations. Les noeuds du réseau de transport sont constitués par des stations de bus. Des séquences ordonnées de noeuds de transport constituent des sections (ou tronçons) qui composent les lignes de transport.

A chaque ligne de bus (appelée route dans le format GTFS), on associe plusieurs trajets en fonction de la fréquence de fonctionnement du bus. Un trajet correspond à un passage d'un bus (dans un seul sens) de la première à la dernière station. Pour chaque trajet (trip dans le format GTFS), le bus va arriver et va partir de chaque station à une heure donnée. Le trajet d'un bus est défini comme un tuple  $(Trajet_{id}, R_{id}, H)$ , où  $Trajet_{id}$  est l'identifiant du trajet (qui correspond à l'identifiant du bus précis par exemple le cinquième bus qui a circulé sur cette route),  $R_{id}$  représente l'identifiant de la ligne de bus ou de la route (e.g., ligne ou route numéro 2) et  $H$  contient les horaires des bus sur ce trajet.  $H$  est composé d'une séquence de tuples de la forme  $(S_{id}, S_{seq}, t_1, t_2)$  où  $S_{id}$  désigne l'identifiant de la station,  $S_{seq}$  désigne le numéro de séquence de la station par rapport à la ligne du bus (e.g., la deuxième station de la ligne du bus 79 est AFPA), et  $t_1$  et  $t_2$  représente les temps d'arrivé et de départ de la station  $S_{id}$ .

Comme illustré dans les cas d'utilisation présentés au début de ce chapitre, souvent l'interrogation du réseau de transport porte sur une partie d'un trajet d'un bus délimitée par deux (ou plusieurs) stations. En conséquence, nous allons définir la notion de section (ou tronçon) de trajet de bus qui est définie comme la portion de trajet déterminée par deux stations consécutives.

En considérant seulement les positions des stations des bus comme noeuds du réseau de bus (voir la figure 4.7), le mapping sur le réseau routier ne serait pas parfait. Par exemple, si une station est située sur une rue et la suivante sur une autre, le trajet du bus doit passer par l'intersection des rues). Par conséquence, le format GTFS propose

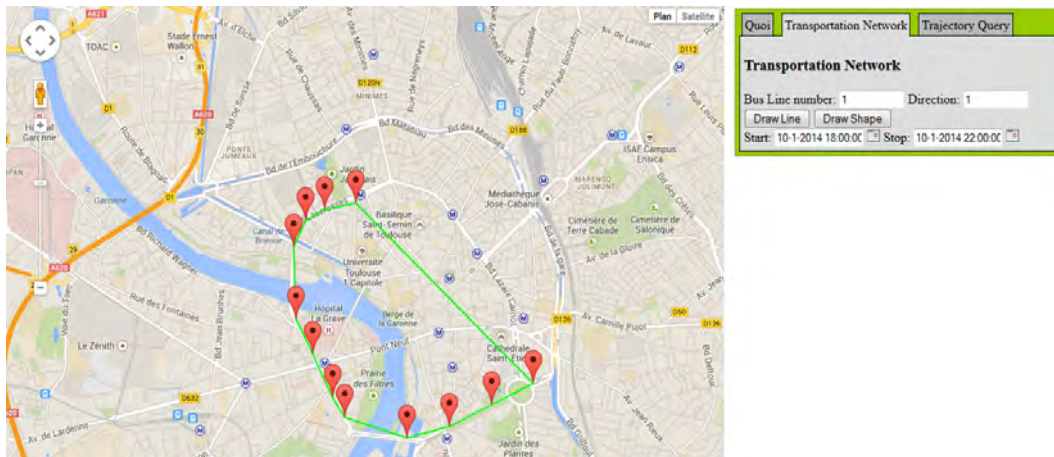


FIGURE 4.7 – Illustration de la ligne du bus numéro 1 sans appliquer la fonction de mapping de la ligne de transport sur le réseau routier

d'associer à chaque trajet une forme (shape dans la nomenclature du format). Une forme est définie comme une liste ordonnée de points qui sont suffisamment nombreux pour décrire la forme réelle d'une ligne de bus (voir la figure 4.8).

Les problèmes du format qui rendent difficile son interrogation que nous avons identifiés sont : (1) le format ne contient pas explicitement les sections de trajet, (2) un mapping des formes des sections des trajets n'est pas possible d'une façon directe. Cela rend l'interrogation et l'optimisation du format lourdes.

Afin de rajouter ces éléments et dans un but d'optimisation, nous proposons le schéma de stockage illustré dans la Figure 4.9 qui garde pour chaque tronçon (section) de ligne de bus les liens vers la liste des points qui donnent le mapping sur le réseau routier et vers les horaires de passage des bus par les stations qui délimitent le tronçon respectif. L'algorithme de construction de ce schéma est réalisé une seule fois. Ce schéma sera maintenu au cas où le réseau de bus subirait des modifications. Suite à l'exécution de cet algorithme, les tronçons de ligne sont créés. En plus, chaque tronçon se voit associé un MBR (Minimum Bounding Rectangle) (voir section 2.4.2 du chapitre 2) qui sera ensuite indexé à l'aide d'un R-tree pour une optimisation de l'interrogation. Aussi, par l'exécution d'un algorithme de recherche du voisin le plus proche (voir section 2.4.4) (algorithme qui est également exécuté une seule fois au moment de la construction du schéma), chaque station se voit associée le point forme le plus proche en termes de latitude et longitude. Cette information est utilisée pour faire un mapping des tronçons de lignes sur le réseau routier et par conséquence

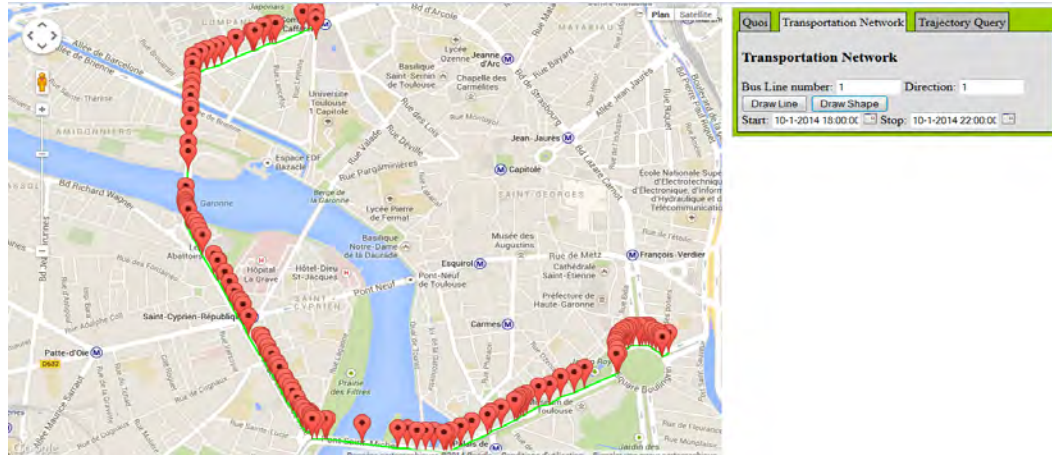


FIGURE 4.8 – Illustration de la ligne du bus numéro 1 après avoir appliqué la fonction de mapping de la ligne de transport sur le réseau routier

de créer une géométrie précise du tronçon (une polyline) qui sera utilisée dans les algorithmes de recherche que nous allons présenter dans une section suivante de ce chapitre. Par la suite nous allons présenter une synthèse des définitions des données liées au réseau de transport.

### Définition des données

*Définition 3* : On définit un réseau de transport  $R_T$  étant un tuple  $(R, Tr, St)$  ou  $R$  est l'ensemble de routes (de lignes de bus), une route étant une séquence de tuples  $(R_{id}, r_{nom})$  ou  $R_{id}$  représente l'identifiant de la route et  $r_{nom}$  son nom (e.g., ligne du bus numéro 2),  $Tr$  est l'ensemble de trajets correspondants aux routes (un trajet correspond à un passage du bus de la première station à la dernière d'un certaine ligne) de  $R$  et  $St$  représente l'ensemble de stations caractérisées par leur positions. La liaison entre un trajet d'un bus et les stations qui le composent est donnée par la définition 5.

*Définition 4* : Le trajet d'un bus (trip dans le format GTFS) est défini comme un tuple  $(Trajet_{id}, R_{id}, H)$ , ou  $Trajet_{id}$  est l'identifiant du trajet (qui correspond à l'identifiant du bus précis par exemple le cinquième bus qui a circulé sur cette route) et  $H$  contient les horaires des bus sur ce trajet.

*Définition 5* : L'horaire de passage  $H$  correspondant à un réseau de transport est composé d'une séquence de tuples  $(S_{id}, S_{seq}, Tr_{id}, t_1, t_2)$  ou  $S_{id}$  désigne l'identifiant

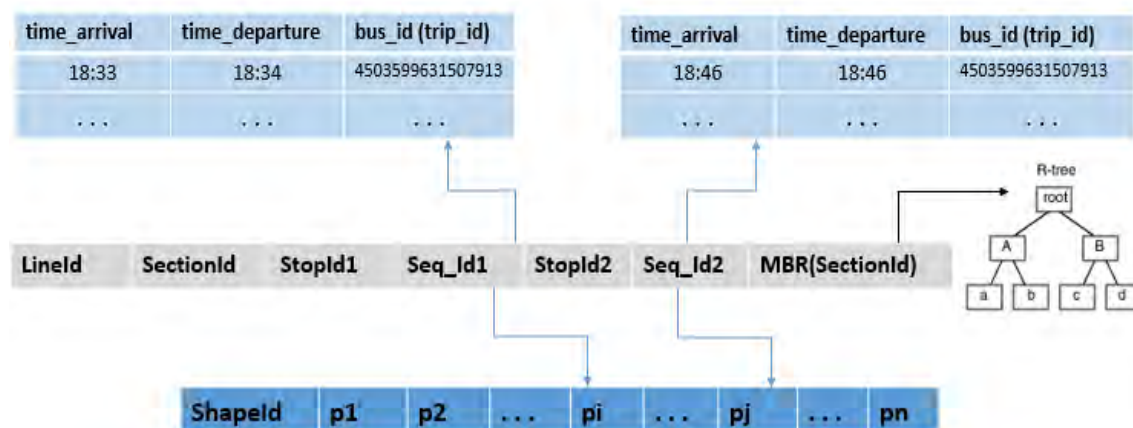


FIGURE 4.9 – Schéma de stockage des tronçons de ligne de transport

de la station,  $S_{seq}$  désigne le numéro de séquence de la station par rapport au trajet de la ligne du bus,  $Tr_{id}$  représente l'identifiant du trajet du bus et  $t_1$  et  $t_2$  représentent l'heure d'arrivée et de départ du bus de la station  $S_{id}$ .

*Définition 6* : Soit la fonction de mapping  $\text{map1} : \text{sections}(R_T) \rightarrow \text{positions}(R_R)$  qui associe à chaque section (tronçon) du réseau de transport en commun une séquence de points sur le réseau routier.

### 4.3.3 Couche Objets

#### Description

La couche *Objets* modélise les positions symboliques et géométriques des objets fixes et mobiles par rapport aux couches sous-jacentes. Les *Objets Fixes* sont situés sur des segments de rue (ont des positions qui appartiennent au réseau  $R_R$ ) (e.g., centres commerciaux, stations de bus). Dans le cas des *Objets Mobiles*, leur position change dans le temps. Chaque objet transmet périodiquement sa position en fonction de différentes stratégies (e.g., chaque  $n$  secondes, chaque fois que l'objet change de direction) que nous ne traitons pas dans notre travail de recherche. Nous supposons que les remontées sont assez fréquentes et que nous avons au moins une position par segment de rue. Dans notre modèle nous distinguons deux types d'objets mobiles : (1) objets qui se déplacent librement dans le réseau routier (e.g., voiture, personne) et (2) objets dont les trajectoires sont contraintes par une ligne (e.g., bus).

Dans notre travail nous nous sommes concentrés sur les objets mobiles qui suivent des trajets prédéfinies dans le cadre d'un réseau de transport en commun urbain (un réseau de bus plus précisément). Ce choix est basé sur trois arguments : (1) la plupart des approches qui définissent des algorithmes qui implémentent des requêtes spatio-temporelles traitent le cas des objets qui bougent librement dans un espace euclidien ou dans un réseau routier et beaucoup moins proposent des algorithmes pour le cas des objets qui suivent des trajets et des horaires prédéfinis, (2) l'intérêt pour des applications concernant les objets qui bougent dans le cadre des réseaux de transport en commun est de plus en plus grand, intérêt qui est matérialisé par la publication des formats standard comme le format GTFS, formats qui peuvent constituer la base pour la définition des extensions des solutions de recherche et d'indexation existantes, (3) le contexte de la vidéosurveillance est très lié aux moyens de transport en commun.

### **Définition des données**

Dans les définitions suivantes nous allons utiliser une notation spécifique pour désigner les types de données spatiaux et temporels (e.g., point, instant).

*Définition 7* : Soit  $FO = \{fo\}$  l'ensemble des objets fixes /  $fo =$  un objet fixe (e.g., un bâtiment, une station de bus),  $id(fo)$  donne son identifiant et  $position(fo)$  donne sa position /  $position(fo) \in \underline{point}$ .

*Définition 8* : Soit  $MO = \{mo\}$  l'ensemble des objets mobiles /  $mo =$  un objet mobile (e.g., bus, voiture, personne),  $id(mo)$  donne son identifiant,  $mposition_i(mo) = (position(mo)_i, t_i)$ ,  $position(mo)_i \in \underline{point}$  et  $time(mposition_i) = t_i \in \underline{instant}$  donne la position de l'objet (qui est un point) à l'instant  $t_i$ .  $TR(mo_i)$  donne la trajectoire de l'objet  $mo_i$  (voir la définition 9).

*Définition 9* : La trajectoire (discrète) d'un objet mobile est un ensemble ordonné de points avec un timestamp associé.  $TR = \{(mposition_i, t_i) / t_i < t_{i+1}\}$ .

### **4.3.4 Couche Caméras**

#### **Description**

Au-dessus de toutes ces données, nous modélisons les *Caméras*. Cette couche est composée des caméras fixes et mobiles. Les caméras fixes ont une position 2D exprimée



par rapport au système géodésique ( $\langle lat, long \rangle$ ) fixée au moment de l'installation. Les caméras mobiles sont associées à un objet mobile (e.g., bus) et leur trajectoire est la même que celle de l'objet mobile. La nouvelle génération de caméras possède des capteurs GPS incorporés et même des boussoles. Les technologies développées autour de ces caméras rendent possible l'extraction automatique des caractéristiques de prise de vue, par exemple : l'orientation, l'inclinaison, etc. (voir section 3.3 du chapitre 3) Ces caractéristiques peuvent changer dans le temps car beaucoup de caméras installées ont des fonctions de balayage (caméras DOM) qui permettent le changement du champ de vue périodiquement ou en fonction des événements (un mouvement brusque qui se produit par exemple) qui sont détectés et qui "attirent l'attention de la caméra". Un grand nombre de caméras embarquées est également déployé dans les bus ou dans les voitures de police.

En se basant sur ces éléments, il est possible de modéliser le champ de vue et de tracer ses modifications dans le temps. Le champ de vue est calculé à partir de cinq paramètres principaux [Arslan Ay et al., 2010a] : la position, l'angle de vue, l'orientation, la distance visible et la taille du capteur. Les équations d'après lesquelles ce calcul est réalisé sont présentées dans la section 3.3.1 du chapitre antérieur. Nous avons classifié ces caractéristiques en deux catégories : caractéristiques de localisation, caractéristiques du champ de vue (caractéristiques optiques : angle de vue, orientation, distance visible, taille du capteur ; caractéristiques d'installation : hauteur, inclinaison). Une présentation de tous les paramètres qui influencent le champ de vue d'une caméra et les méthodes de calcul de la géométrie de la scène filmée par la caméra à un moment donné est réalisée dans la section 3.3 du chapitre 3. Dans notre travail, à cause de la difficulté d'avoir des données de test concernant l'installation des caméras, nous avons considéré que les caractéristiques de localisation et les caractéristiques optiques dans le calcul des champs de vue. La prise en compte des données d'installation peut être faite de façon simple selon les formules présentées dans la section 3.3 du chapitre antérieur.

### **Définition des données**

*Définition 10* : Soit  $fov = (focalDistance, direction, visibleDistance, angleView, sensorSize, t_i)$ , un vecteur de caractéristiques optiques d'une caméra  $c_j$  à l'instant  $t_i$ . A partir de ces caractéristiques, le champ de vue (horizontal) de la caméra peut être calculé [Arslan Ay et al., 2010a].

*Définition 11* : Soit  $FC = \{fc\}$  l'ensemble de caméras fixes/  $fc$  est une caméra fixe,

$\text{id}(fc)=c_i$  donne son identifiant,  $\text{position}(c_i)$  donne sa position/  $\text{position}(c_i) \in \underline{\text{point}}$  et  $\text{fov}(c_i)$  donne l'ensemble  $\text{fov}_i$  des changements de son champ de vue.

*Définition 12* : Soit  $\text{MC} = \{\text{mc}\}$  l'ensemble des caméras mobiles/  $\text{mc}$  est une caméra mobile attachée à un objet mobile,  $\text{id}(\text{mc})=c_i$  donne son identifiant et  $\text{mo}(c_i) \in \text{MO}$  donne l'objet mobile auquel la caméra est associée. La trajectoire de la caméra mobile  $c_i$  sera celle de l'objet mobile avec l'identifiant  $\text{mo}(c_i)$  (voir définitions 8 et 9).

Avant de passer à la présentation de la spécification des opérateurs que nous proposons dans notre travail, nous définissons les fonctions et prédicats suivants :

- $\text{geometry}(\underline{\text{point}} p, \underline{\text{fov}} f) : \underline{\text{region}}$  calcule la géométrie du champ de vue d'une caméra à partir de sa position et de ses caractéristiques optiques à un instant de temps.
- $\text{intersects}(\underline{\text{polyline}} \text{seg}, \underline{\text{region}} g) : \text{boolean}$  permet de vérifier si la géométrie  $g$  intersecte un segment de rue  $\text{seg}$ .
- $\text{intersects}(\underline{\text{point}} p, \underline{\text{polyline}} \text{seg}) : \text{boolean}$  permet de vérifier si un point  $p$  intersecte un segment de rue  $\text{seg}$ .
- $\text{intersects}(\underline{\text{point}} p, \text{set}(\underline{\text{polyline}}) \text{tr}) : \text{boolean}$  permet de vérifier si un point  $p$  intersecte un ensemble de segments et qui retourne vrai si  $\exists \text{seg}_i \text{ in tr} / \text{intersects}(p, \text{seg}_i)$ .

Pour la suite, nous disposons de deux fonctions,  $\text{prec}$  et  $\text{succ}$ , définies sur des ensembles ordonnés permettant d'atteindre respectivement le prédécesseur et le successeur d'un élément de l'ensemble.

En se basant sur les définitions de données présentées dans la section antérieure ainsi que sur les fonctions et les prédicats que nous venons de décrire, nous allons présenter des opérateurs pour la sélection des caméras fixes et mobiles pertinentes pour une requête trajectoire hybride donnée.

## 4.4 Spécification des opérateurs

Comme défini dans la section 4.2.2, le point d'entrée de notre système est une requête trajectoire hybride, constituée des positions géométriques et symboliques exprimées par rapport à des différents systèmes de référence et d'un intervalle de temps. Une fois que cette requête est interprétée par un module interpréteur de requête par rapport à notre modèle de données, elle est transformée en :

- une trajectoire spatiale constituée d'une séquence de segments  $(u_1, u_2, \dots, u_k)$ ;

– un intervalle de temps  $[t_1, t_2]$ .

Ces données peuvent être extraites à partir des données d'enquête. Le but est de proposer à l'opérateur une liste de séquences vidéo susceptibles de contenir la trajectoire recherchée. Pour cela, il nous faut rechercher les caméras fixes dont le champ de vue (qui peut être variable) intersecte la trajectoire dans l'intervalle de temps et les caméras mobiles susceptibles d'avoir filmé la scène recherchée.

Compte tenu des objectifs définis, le but est de trouver tous les champs de vues des caméras fixes qui ont intersecté le trajet dans l'intervalle donné  $([t_1, t_2])$  et les caméras mobiles susceptibles de contenir des informations pertinentes pour la requête car la trajectoire de l'objet mobile auquel elles sont associées a intersecté le trajet entre  $t_1$  et  $t_2$ . Nous allons présenter par la suite séparément les deux cas de sélection des caméras fixes et respectivement des caméras mobiles.

L'opérateur `hasSeen` se définit ainsi [Codreanu et al., 2014b], [Codreanu et al., 2014a]. Étant donnée la trajectoire spatiale composée de segments  $tr=(u_1, \dots, u_n)$  et l'intervalle de temps  $[t_1, t_2]$ , `hasSeen(tr, t_1, t_2)` retourne l'ensemble des caméras  $c_i$  ( $1 \leq i \leq m$ ) associées à un segment  $u_k$  ( $1 \leq k \leq n$ ) et un extrait vidéo entre deux instants  $t_{start}^i$  et  $t_{end}^i$  avec  $t_{start}^i \in [t_1, t_2]$  et  $t_{end}^i \in [t_1, t_2]$ . Chaque élément de cet ensemble indique que l'extrait vidéo entre  $t_{start}^i$  et  $t_{end}^i$  de la camera  $c_i$  a filmé le segment  $u_k$ . C'est donc l'ensemble des segments vidéos des caméras susceptibles d'avoir filmé la scène recherchée.

$$hasSeen : u_1, u_2, \dots, u_n, [t_1, t_2] \Rightarrow \begin{cases} c_1 : t_{start}^1 - > t_{end}^1, u_k (1 \leq k \leq n) \\ c_2 : t_{start}^2 - > t_{end}^2, u_k (1 \leq k \leq n) \\ \dots \\ c_m : t_{start}^m - > t_{end}^m, u_k (1 \leq k \leq n) \end{cases}$$

Par la suite nous allons décrire les spécifications de l'opérateur `hasSeen` séparément pour les caméras fixes et mobiles.

#### 4.4.1 Caméras fixes

Le résultat pour les cameras fixes est l'ensemble de triplets :  $R = \{r = (c_i, u_k, [t_a, t_b])\}$ ,  $c_i \in FC$ ,  $u_k \in tr$ ,  $t_1 \leq t_a < t_b \leq t_2$ . L'opérateur que nous avons défini vérifie quelles sont les caméras fixes dont le champ de vue a intersecté un des segments du trajet de la requête et entre quels moments de temps (les instants  $t_a$  et  $t_b$ ).

$r \in R \equiv$  Il existe  $fov_j \in fov(c_i)$  / ( $fov(c_i)$  est l'ensemble des instants de changement du champ de vue de la caméra  $c_i$ ) tel que :

$\text{time}(fov_j) \in [t_1, t_2]$ $\wedge \text{intersects}(u_k,$ $\quad \text{geometry}(\text{position}(c_i), fov_j))$ $\wedge t_a = \text{time}(fov_j)$ $\wedge t_b = \min(\text{time}(\text{succ}(fov_j)), t_2)$	<p><i>Dans le cas où le point de changement est à l'intérieur de l'intervalle <math>[t_1, t_2]</math>, si la géométrie du champ de vue correspondant intersecte le segment du trajet de la requête <math>u_k</math>, alors l'intervalle de temps commence au moment du changement du champ de vue et se finit au prochain changement du champ de vue ou à la fin de l'intervalle de la requête</i></p>
$\vee$ $\text{time}(fov_j) \in [t_1, t_2]$ $\wedge \text{intersects}(u_k,$ $\quad \text{geometry}(\text{position}(c_i), fov_j))$ $\wedge t_a = t_1$ $\wedge t_b = \min(\text{time}(\text{succ}(fov_j)), t_2)$	<p><i>Dans le cas où le point de changement est avant <math>t_1</math>, et le suivant est après <math>t_1</math>, si la géométrie du champ de vue correspondant intersecte le segment du trajet <math>u_k</math> de la requête, alors l'intervalle de temps commence au moment <math>t_1</math> et se finit au prochain changement du champ de vue ou à la fin de l'intervalle de la requête.</i></p>

#### 4.4.2 Caméras mobiles

L'opérateur que nous avons défini retrouve quelles sont les caméras associées à des objets mobiles dont une position connue intersecte un des segments du trajet de la requête et entre quels moments de temps il peut l'avoir intersecté (instants  $t_a$  et  $t_b$ ). Dans ce cas, la mobilité des caméras ne permettent pas de calculer l'intersection précise entre le champ de vue de la caméra et les segments de la requête (par exemple une caméra placée sur le côté extérieur droit d'un bus). Le résultat obtenu indique uniquement que la caméra mobile se trouvait sur un segment de la requête dans l'intervalle de temps de la requête.

Le résultat pour les caméras mobiles est l'ensemble de triplets :  $R = \{r = (c_i, u_k, [t_a, t_b])\}$ ,  $c_i \in MC$ ,  $u_k \in tr$ ,  $t_1 \leq t_a < t_b \leq t_2$ .

$r \in R \equiv$  Il existe  $mposition_i \in TR(\text{mo}(c_i))$  (il existe une position dans la trajectoire de l'objet mobile auquel la caméra mobile  $c_i$  est attachée) tel que :

$\text{time}(m\text{position}_i) \in [t_1, t_2]$ $\wedge \text{intersects}(\text{position}(m\text{position}_i), u_k)$ $\wedge \text{not intersects}$ $(\text{position}(\text{prec}(m\text{position}_i)), \text{tr})$ $\vee (\text{intersects}(\text{position}(\text{prec}(m\text{position}_i)), \text{tr})$ $\wedge \text{time}(\text{prec}(m\text{position}_i)) < t_1$ $\wedge t_a = \max(\text{time}(\text{prec}(m\text{position}_i)), t_1)$ $\wedge t_b = \min(\text{time}(\text{succ}(m\text{position}_i)), t_2)$ $\vee$ $\text{time}(m\text{position}_i) \in [t_1, t_2]$ $\wedge \text{intersects}(\text{position}(m\text{position}_i), u_k)$ $\wedge \text{intersects}$ $(\text{position}(\text{prec}(m\text{position}_i)), \text{tr})$ $\wedge t_a = \text{time}(m\text{position}_i)$ $\wedge t_b = \min(\text{time}(\text{succ}(m\text{position}_i)), t_2)$	<p><i>Dans le cas où la remontée de la position de l'objet est sur le trajet de la requête et est à l'intérieur de l'intervalle <math>[t_1, t_2]</math> et la position d'avant n'est pas sur le trajet de la requête, ou la position d'avant de l'objet est sur la trajet mais avant <math>t_1</math>, alors l'intervalle de temps commence au maximum entre le moment de la dernière remontée et <math>t_1</math> et se finit à la prochaine remontée ou à <math>t_2</math></i></p> <p><i>Dans le cas où la remontée de la position de l'objet est sur le trajet de la requête et est à l'intérieur de l'intervalle <math>[t_1, t_2]</math> et la position d'avant est aussi sur le trajet de la requête, alors l'intervalle de temps commence au moment de la remontée et se finit à la prochaine remontée ou à <math>t_2</math></i></p>
---	--

Par la suite, nous allons présenter les algorithmes qui implémentent les deux opérateurs définis.

## 4.5 Algorithmes de recherche

Les algorithmes qui implémentent les spécifications précédentes se déroulent en deux étapes, *l'étape de sélection des candidats* et *l'étape d'affinage des résultats*. Dans les deux cas, sélection des caméras fixes ou mobiles, l'étape de sélection des candidats est basée seulement sur un filtrage purement spatial, l'aspect temporel étant pris en compte dans l'étape d'affinage des résultats.

La trajectoire initiale donnée comme requête contient des segments composés des positions géométriques et symboliques exprimées par rapport à des systèmes de référence différents et un intervalle de temps. Après l'interprétation d'une telle requête, elle est exprimée comme une séquence de segments spatiaux (poli-lignes) et un intervalle de temps. Cette séquence de segments avec l'information temporelle constitue l'entrée des deux algorithmes.

#### 4.5.1 Algorithme de sélection des caméras fixes

Afin de sélectionner les caméras fixes (mais dont le champs de vue peut changer), nous proposons l'algorithme 1. Par la suite, nous allons expliquer les deux étapes de cet algorithme :

- *L'étape de filtrage* applique un algorithme correspondant à une requête de type region query similaire à celui présenté dans [Tao et al., 2002] afin de sélectionner pour chaque segment de la requête les caméras situées à une distance inférieure ou égale à la distance de visibilité maximale de toutes les caméras existantes dans la base de données. Cela évite d'évaluer l'intersection spatiale (opération coûteuse) pour des champs de vue des caméras qui sont situées à une distance qui rend impossible la visibilité de la trajectoire requête (le filtrage est exécuté par la fonction  $extraireCamDist(u_k, \max(visibleDistance(fov_j)))$ ), où  $u_k$  représentent les segments de la requête trajectoire hybride et  $visibleDistance(fov_j)$  donne la distance de visibilité maximale d'un champ de vue d'une caméra.
- Seulement pour les caméras qui ont été sélectionnées suite au filtrage, *L'étape d'affinage* calcule la géométrie des instances des champs de vue ( $construire\_polygone(fov_j(c_i))$ ) concernées par l'intervalle de temps de la requête. Sachant qu'une base de données de caméras (qui est censée contenir les caractéristiques des caméras appartenant à plusieurs systèmes qui gèrent des réseaux de vidéosurveillance) peut contenir un nombre impressionnant de caméras, cette approche présente un grand avantage par rapport aux approches qui calculent et stockent la géométrie dans la base ou qui ne proposent pas de filtrage. Cette étape est illustrée par un exemple de requête présenté dans la section 4.6.1.

**Algorithm 1:** Algorithme de sélection des cameras fixes qui ont intersecté la trajectoire de la requête

---

**Entrées:** Une suite de segments de rue :  $u_k$  et un intervalle de temps :  $[t_1, t_2]$ .

**Sorties:** La liste des caméras fixes qui ont vu la trajectoire donnée avec les segments de rues et les intervalles de temps correspondants

```

1  pour chaque  $u_k$  de la requête faire
2  |    $maxVisibleDistance \leftarrow max(visibleDistance(fov_j))$ 
3  |    $listeCam \leftarrow extraireCamDist(u_k, maxVisibleDistance)$ 
4  fin
5  pour chaque  $c_i$  de  $listeCam$  faire
6  |   pour chaque  $(fov_j(c_i))$  faire
7  |   |   si  $time(fov_j(c_i)) \geq t_1$  et  $time(fov_j(c_i)) \leq t_2$  alors
8  |   |   |    $geometry_{ij} \leftarrow construire\_polygone(fov_j(c_i));$ 
9  |   |   |   pour chaque  $u_k$  de la requête faire
10 |   |   |   |   si  $geometry_{ij}$  intersecte  $u_k$  alors
11 |   |   |   |   |    $ajouterResultat(c_i, u_k, [time(fov_j), min(succ(time(fov_j)), t_2)]);$ 
12 |   |   |   |   fin
13 |   |   |   fin
14 |   |   fin
15 |   |   si  $time(fov_j(c_i)) < t_1$  et  $t_1 \leq time(succ(fov_j(c_i)))$  alors
16 |   |   |    $geometry_{ij} \leftarrow construire\_polygone(fov_j(c_i));$ 
17 |   |   |   pour chaque  $u_k$  de la requête faire
18 |   |   |   |   si  $geometry_{ij}$  intersecte  $u_k$  alors
19 |   |   |   |   |    $ajouterResultat(c_i, u_k, [t_1, min(time(succ(fov_j)), t_2)]);$ 
20 |   |   |   |   fin
21 |   |   |   fin
22 |   |   fin
23 |   fin
24 fin

```

---

## 4.5.2 Algorithme de sélection des caméras mobiles

Comme nous nous intéressons aux systèmes de vidéosurveillance urbains et à la sécurité des moyens de transport en commun, nous avons considéré le cas des caméras mobiles qui sont attachées aux bus, donc aux objets mobiles dont le trajet est connu. En conséquence nous avons "customisé" l'algorithme de recherche des caméras mobiles par l'introduction d'une étape de filtrage qui s'appuie sur les informations concernant les trajets et les horaires de passage des bus afin de sélectionner les objets

mobiles qui ont intersecté la trajectoire de la requête.

Plus précisément, à partir du standard de description des données de transport GTFS, nous proposons le schéma de stockage des données décrivant les stations et les horaires des bus illustré dans 4.9. A partir de ce schéma de stockage qui pour chaque section ou tronçon de ligne de bus (défini par deux stations consécutives), garde des liens vers la liste de points qui "dessinent" la forme du tronçon et vers la liste des horaires de passage des bus par les deux stations qui déterminent le tronçon respectif.

Nous allons détailler les deux étapes de l'algorithme comme dans le cas des caméras fixes :

- La première étape de l'algorithme, *l'étape de filtrage* (lignes 3-10 de l'algorithme 2), sélectionne les sections (ou tronçons) des trajets des bus qui intersectent les segments de la requête. Les sections des trajets des bus sont indexées en utilisant un index spatial R-tree. Cette première étape de l'algorithme exécute aussi un filtrage temporel (la fonction  $filterTime(section_j, t_1, t_2)$  implémentée par l'algorithme 3) pour affiner la liste d'objets candidats calculée lors du filtrage spatial. Lors de ce filtrage temporel, parmi les bus sélectionnés lors du filtrage spatial, nous allons sélectionner les bus qui sont censés passer par les stations qui déterminent les tronçons qui intersectent le trajet de la requête entre  $t_1$  et  $t_2$ . L'intervalle de la requête  $[t_1, t_2]$  est ajusté afin de prendre en compte un facteur d'incertitude qui est estimé en fonction du jour de la semaine, du moment du jour (e.g., en général les bus ont plus de retard quand le trafic est congestionné) ou des événements spéciaux (e.g., une grève). Jusqu'à ce point, l'algorithme se base sur des données stockées dans le schéma 4.9 construite à partir des données de description du réseau de bus. Ce schéma est construit une seule fois à priori.
- L'étape d'affinage des résultats candidats générés par la première étape de cet algorithme utilise des données de localisation réelles concernant les trajets des bus. La fonction  $affinage(mo_i, u_k, [t_1, t_2])$  implémente l'opérateur défini dans la section 4.4.2 de ce chapitre. Le résultat final de cette étape est donnée par la liste des caméras associées aux bus dont le trajet a intersecté la trajectoire de la requête entre  $[t_1, t_2]$ , avec le segment et l'intervalle de temps correspondant.

Un exemple concret de sélection réalisée à l'aide de cet algorithme est donné dans la section suivante.



**Algorithm 2:** Algorithme de sélection des caméras mobiles qui ont intersecté la trajectoire de la requête

---

**Entrées:** Une suite de segments de rue :  $u_k$  et un intervalle de temps :  $[t_1, t_2]$ .

**Sorties:** La liste des caméras mobiles susceptibles d'avoir filmé la trajectoire donnée

```
1 listSectionCandidates ← null
2 listFinalObj ← null
3 listSections ← filterMBR( $u_k$ );
4 pour chaque sectioni ∈ listSections faire
5   | pour chaque  $u_k$  faire
6   |   | si intersects(polyline(sectioni),  $u_k$ ) alors
7   |   |   | listSectionCandidates ← ajouter(listCandidates, sectioni);
8   |   |   fin
9   |   fin
10 fin
11 pour chaque sectionj ∈ listCandidates faire
12 | listMOCandidates ← filterTime(sectionj,  $t_1$ ,  $t_2$ )
13 fin
14   /* Passage à l'étape d'affinage de résultats */
15 pour chaque  $u_k$  faire
16 |   pour chaque  $mo_i$  ∈ listMOCandidates faire
17 |   | listFinalObj ← ajouterResultat(listFinalObj, affinage( $mo_i$ ,  $u_k$ ,  $[t_1, t_2]$ ));
18 |   fin
19 fin
20 pour chaque  $mo_i.id$  de listFinalObj faire
21 | listeCameras ← selectionnerCameras( $mo_i.id$ );
22 fin
```

---

---

**Algorithm 3:** Algorithmme de la fonction  $filterTime(section_i, t_1, t_2)$

---

**Entrées:** La liste de sections de trajets de bus candidates  $listCandidates$  et un intervalle de temps :  $[t_1, t_2]$ .

**Sorties:** La liste des bus qui ont traversé les sections de trajets dans l'intervalle  $[ti_1, ti_2]$

```

1  $listMOCandidates \leftarrow null$ 
2  $uncertainty \leftarrow val$ 
3  $ti_1 \leftarrow t_1 - uncertainty;$ 
4  $ti_2 \leftarrow t_2 + uncertainty;$ 
5 pour chaque  $section_i \in listCandidates$  faire
6   pour chaque  $bus_j$  qui passe par la section  $i$  faire
7     si
8        $((arrivalTime(bus_j, section_i.stop_1) > ti_1) \wedge (arrivalTime(bus_j, section_i.stop_2) >$ 
9          $ti_2 \vee arrivalTime(bus_j, section_i.stop_2) < ti_2)) \vee$ 
10         $((arrivalTime(bus_j, section_i.stop_1) < ti_1) \wedge (arrivalTime(bus_j, section_i.stop_2) <$ 
11          $ti_2) \vee (arrivalTime(bus_j, section_i.stop_2) > ti_2))$  alors
12       |  $listMOCandidates \leftarrow ajouter(listMOCandidates, bus_j)$ 
13     fin
14   fin
15 fin

```

---

## 4.6 Exemples de requêtes

### 4.6.1 Caméras fixes

Supposons les trois caméras illustrées dans les schémas des figures 4.10, 4.11, 4.12, 4.13 illustrant les localisations des caméras fixes  $C_1$ ,  $C_2$  et  $C_3$ . Le réseau routier est représenté par les deux rues (S1-S5 et S6-S10). Supposons la trajectoire de la requête  $TR = S1, S2, S3, S4, S5$  et l'intervalle de la requête  $I = [t_1, t_2]$ . Les moments de changement des champs de vues des caméras sont illustrés dans la figure 4.14. On distingue 4 intervalles de temps ( $[time(fov_j(C_3)), time(fov_k(C_2))]$ ,  $[time(fov_k(C_2)), time(fov_{j+1}(C_3))]$ ,  $[time(fov_{j+1}(C_3)), time(fov_{k+1}(C_2))]$  et  $[time(fov_{k+1}(C_2)), \dots]$ ).

La Figure 4.14 présente les prises de vue des caméras  $C_2$  et  $C_3$  en fonction de temps. Les différents moments où les champs de vue des caméras  $C_2$  et  $C_3$  changent (chacune des quatre figures 4.10, 4.11, 4.12, 4.13 correspond à un des quatre intervalles) sont marqués en couleurs correspondant aux géométries des champs de vues des caméras

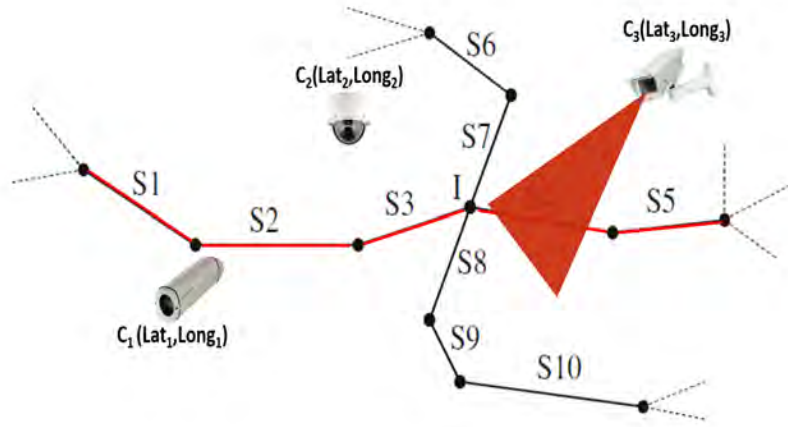


FIGURE 4.10 – Champs de vue des caméras dans le premier intervalle de temps

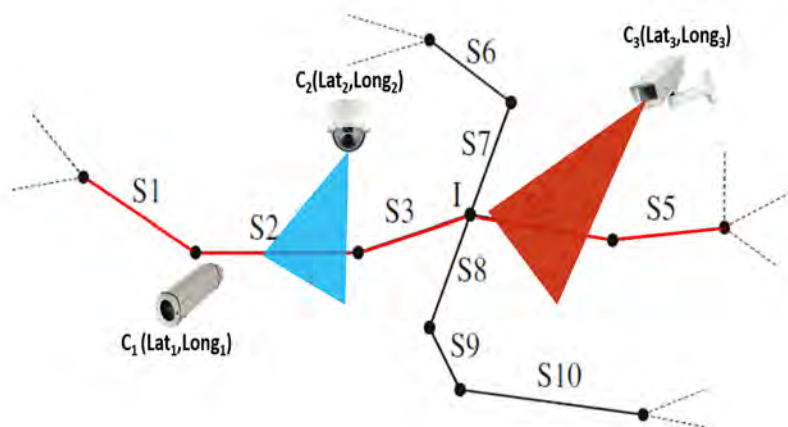


FIGURE 4.11 – Champs de vue des caméras dans le deuxième intervalle de temps

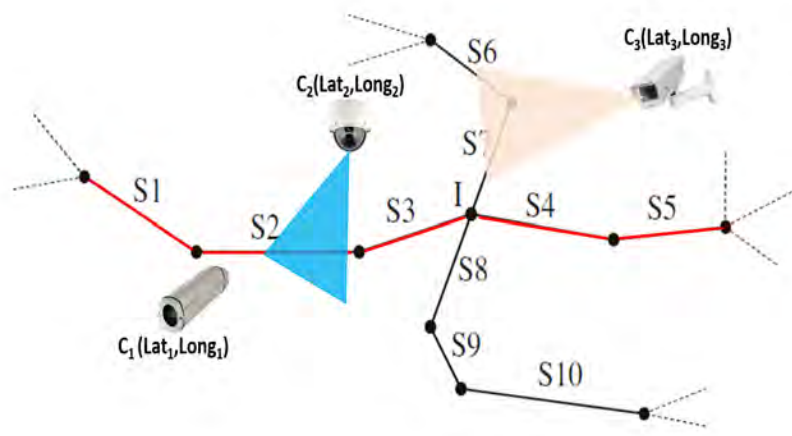


FIGURE 4.12 – Champs de vue des caméras dans le troisième intervalle de temps

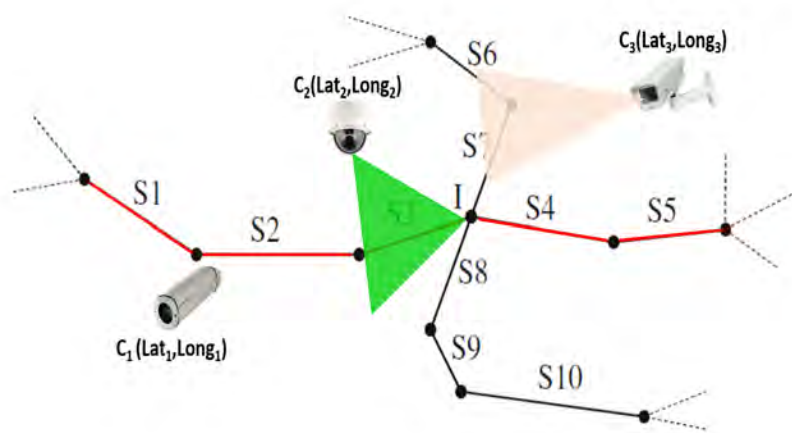


FIGURE 4.13 – Champs de vue des caméras dans le quatrième intervalle de temps

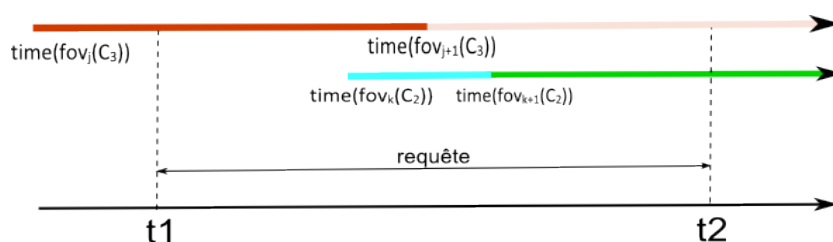


FIGURE 4.14 – Les moments de changement des champs de vue des caméras fixes

illustrées dans les figures 4.10, 4.11, 4.12, 4.13.

Supposons l'intervalle de la requête  $[t_1, t_2]$  avec  $time(fov_j(C_3)) < t_1 < time(fov_k(C_2))$  et  $time(fov_{j+1}(C_3)) < t_2$ .

La sélection des caméras fixes est réalisée par l'algorithme 1. Nous allons expliquer cet algorithme par l'intermédiaire de l'exemple illustré dans les quatre figures. Les premières lignes de l'algorithme (1-4) représentent une étape de filtrage. De toutes les caméras de la base de donnée nous allons sélectionner celles qui se situent à une distance maximale inférieure ou égale à la distance de visibilité maximale des caméras de la base. Dans notre cas les seules caméras qui ont possiblement filmé les segments de la requête sont  $C_1$ ,  $C_2$  et  $C_3$ .

Pour chacune des caméras sélectionnées lors de la première étape, nous allons chercher les moments où celles-ci ont modifié leur champ de vue (lignes 5,6 de l'algorithme). Les lignes 7-20 traitent les deux cas possibles : le changement est entre  $t_1$  et  $t_2$  (e.g.,  $time(fov_k(C_2))$ ) ou le changement est avant  $t_1$  (e.g.,  $time(fov_j(C_3))$ ). Les géométries des champs de vues sont construites et l'intersection avec les segments de la requête sont évaluées.

Le résultat de la requête pour notre exemple est :

$\{(C_2, S_2, [time(fov_k(C_2)), time(fov_{k+1}(C_2))]), (C_2, S_3, [time(fov_{k+1}(C_2)), t_2]), (C_3, S_4, [t_1, time(fov_{j+1}(C_3))])\}$ .

## 4.6.2 Caméras mobiles

Considérons le même schéma de la figure 4.15. Le but est maintenant de sélectionner les caméras mobiles susceptibles d'avoir filmé la trajectoire de la requête  $TR=S1, S2, S3, S4, S5$ . Cette sélection est faite selon l'algorithme 2.

Après l'étape de filtrage spatial et temporel des trajectoires des objets mobiles (lignes

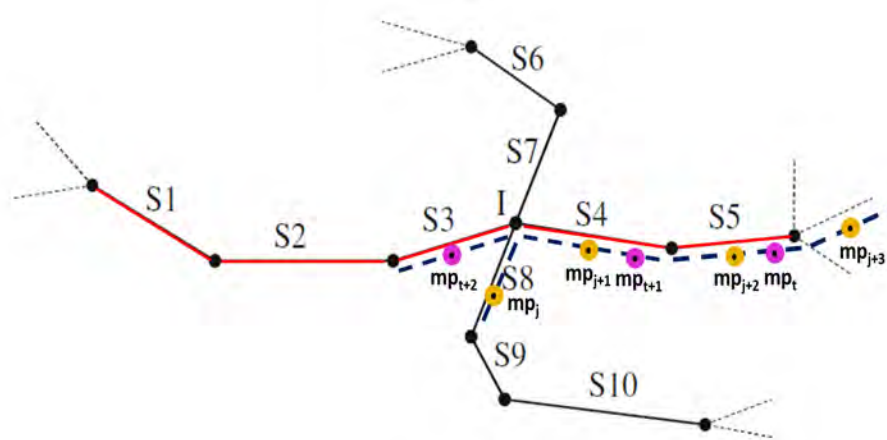


FIGURE 4.15 – Illustration d'une requête de sélection des objets mobiles

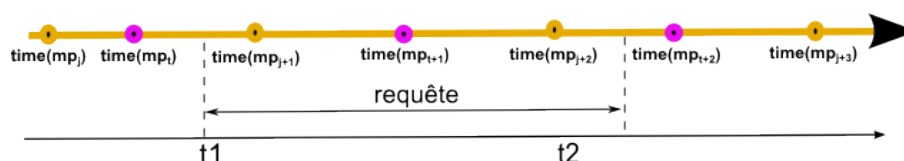


FIGURE 4.16 – Les changements des positions des objets mobiles dans le temps

3-13 de l'algorithme 2), deux objets candidats sont sélectionnés (deux bus ont traversé la trajectoire de la requête dans l'intervalle de temps donné). Supposons les deux objets mobiles ayant les trajectoires marquée en pointillé sur la figure (S8, S4, S5, ...). Nous supposons que l'objet envoie au moins une remontée  $mp_j$  (mobile position) contenant sa position et un timestamp par segment de rue. En considérant chaque segment de la rue et chaque objet mobile (lignes 14-18 de l'algorithme), la fonction  $affinage(mo_i, u_k, [t_1, t_2])$  va tester les cas expliqués dans 4.4.2 : la position de l'objet est sur la trajectoire de la requête entre  $t_1$  et  $t_2$  ( $mp_t$ ,  $mp_t$ ,  $mp_{j+1}$ ,  $mp_{j+2}$  comme illustré dans la Figure 4.16) et la position d'avant intersecte aussi ( $mp_{j+1}$  et  $mp_{j+2}$ ) ou la position d'avant n'intersecte pas la trajectoire ( $mp_j$  et  $mp_{j+1}$ ) ou elle intersecte mais avant  $t_1$  ( $mp_t$  et  $mp_{t+1}$ ).

Le résultat est  $\{(obj_i, S_4, [t_1, \text{time}(mp_{j+2})]), (obj_i, S_5, [\text{time}(mp_{j+2}), t_2]), (obj_{i+1}, S_4, [t_1, t_2])\}$

## 4.7 Conclusion

Dans ce chapitre nous avons présenté notre proposition :

- d’un modèle de données qui intègre des informations spatio-temporelles décrivant le réseau routier et celui de transport d’une ville, les objets mobiles (e.g., bus) qui circulent dans ces réseaux et les caméras fixes et mobiles qui composent le réseau de vidéosurveillance urbain ;
- d’un modèle de requête trajectoire hybride composée d’un ensemble de segments spatiaux contenant des positions géométriques et symboliques exprimées par rapport aux différents systèmes de références ;
- d’un opérateur qui, en se basant sur le modèle de données proposé et sur la requête trajectoire hybride définie, extrait de façon automatique les caméras fixes et mobiles susceptibles d’avoir filmé la trajectoire donnée comme requête

Les opérateurs de sélection proposés se basent sur un modèle de données spatiales et temporelles des caméras de vidéosurveillance dans le cadre d’un réseau de transport en commun urbain qui permet la reconstitution des trajectoires des objets mobiles des caméras mobiles (associées à des objets mobiles) et des positions des caméras fixes ; chaque caméra se voit associé un champ de vue dont la géométrie est calculée au moment de la requête pour les caméras fixes susceptibles d’avoir capturé des vidéos pertinentes pour la trajectoire cible. L’approche proposée, renforcée par des solutions d’indexation et de stockage optimisées et complétée par des solutions d’analyse du contenu vidéo, peut constituer la base d’outils de Forensic [Sedes et al., 2012] si recherchés dans le cadre de la vidéosurveillance.





# Chapitre 5

## Application

### 5.1 Introduction

Le but de ce chapitre est de proposer des éléments de validation du modèle et des algorithmes présentées dans le chapitre antérieur. Comme déjà expliqué et montré dans le chapitre de l'état de l'art dans les sections 2.3.1 et 2.3.2, la validation n'est pas une tâche facile quand il faut croiser des données concernant le réseau routier, le réseau de transport et des données liées aux caméras fixes et mobiles. En particulier, les jeux de données sont difficile à constituer. Le but de ce chapitre est de présenter notre proposition d'architecture de framework d'interrogation des données spatio-temporelles liées aux caméras, de montrer l'utilité et la faisabilité du framework par l'intermédiaire d'exemples de cas d'utilisation illustrant le fonctionnement du framework. Les expérimentations réalisées portent sur trois points : i) la définition et l'implémentation d'un prototype du framework proposé, ii) un cas d'utilisation de recherche de vidéos montrant la faisabilité, iii) une étude qualitative d'estimation du temps de calcul sur un jeu de données constitué à partir de l'intégration d'un réseau routier externe (API Google Maps), d'un réseau de transport et données de localisation des bus de la ville de Toulouse fournies dans le cadre du projet Toulouse OpenData et des métadonnées liées aux caméras capturées à partir des capteurs embarqués dans des smartphones Android.

Ce chapitre commence (Section 5.2) par la présentation de l'architecture du framework que nous avons proposé et qui implémente le modèle de données, le modèle de requête trajectoire hybride et les algorithmes de recherche présentés dans le Chapitre

précédent. Ensuite, dans les sections 5.2.1, 5.2.2, 5.2.3 et 5.2.4 les principaux modules de cette architecture sont détaillés. Afin de montrer la faisabilité du framework implémenté, un exemple de réalisation d'un cas d'utilisation est présenté dans 5.3. Des expérimentations qui illustrent une étude qualitative des algorithmes de sélection (présentés dans la section 4.5.2 du Chapitre 4) et leur comportement en fonction de la taille du dataset utilisé. Nous finissons par une conclusion.

## 5.2 Architecture du framework proposé

Le modèle de données et les algorithmes proposés dans le chapitre précédent ont été implémentés et testés dans le cadre d'un framework dont l'architecture est illustrée dans la Figure 5.1. Les principales modules d'un tel framework sont :

- Terminal Interface (Interface utilisateur) : regroupe les briques qui réalisent l'interaction avec l'utilisateur et la visualisation des résultats et des données via une API Google Maps ;
- Query Interpreter (Interpréteur de requête) : réalise la transformation de la requête trajectoire hybride (conformément à la définition donnée dans la section 4.2.2 du Chapitre 3) soumise par l'utilisateur en requête spatio-temporelle (une séquence de segments spatiaux projetés sur le réseau routier de la ville et un intervalle de temps) ;
- Search Engine (Moteur de recherche) : implémente les algorithmes de recherche définis dans le chapitre précédent en deux étapes : filtrage et affinage des résultats ;
- Storage (Stockage) : contient la base de données spatio-temporelle et la collection de données et des modules qui permet l'interaction avec celles ci ;
- SQL Query Generator (Générateur de requête SQL) : réalise la communication entre le moteur de recherche et le module de stockage ;
- Data Collecting (Collecte de données) : est en charge de de la collecte des données vidéos et spatio-temporelles issues des capteurs GPS, boussoles et accéléromètres et de la synchronisation de toutes ces données ;

L'implémentation du prototype a été réalisée en utilisant JDK 1.7 avec les drivers JDBC pour la connexion avec la base de données Oracle et la technologie JSP pour la connexion avec l'interface codée en utilisant HTML, JAVASCRIPT et AJAX.

Par la suite nous allons présentons en détail les modules les plus importants de notre architecture.

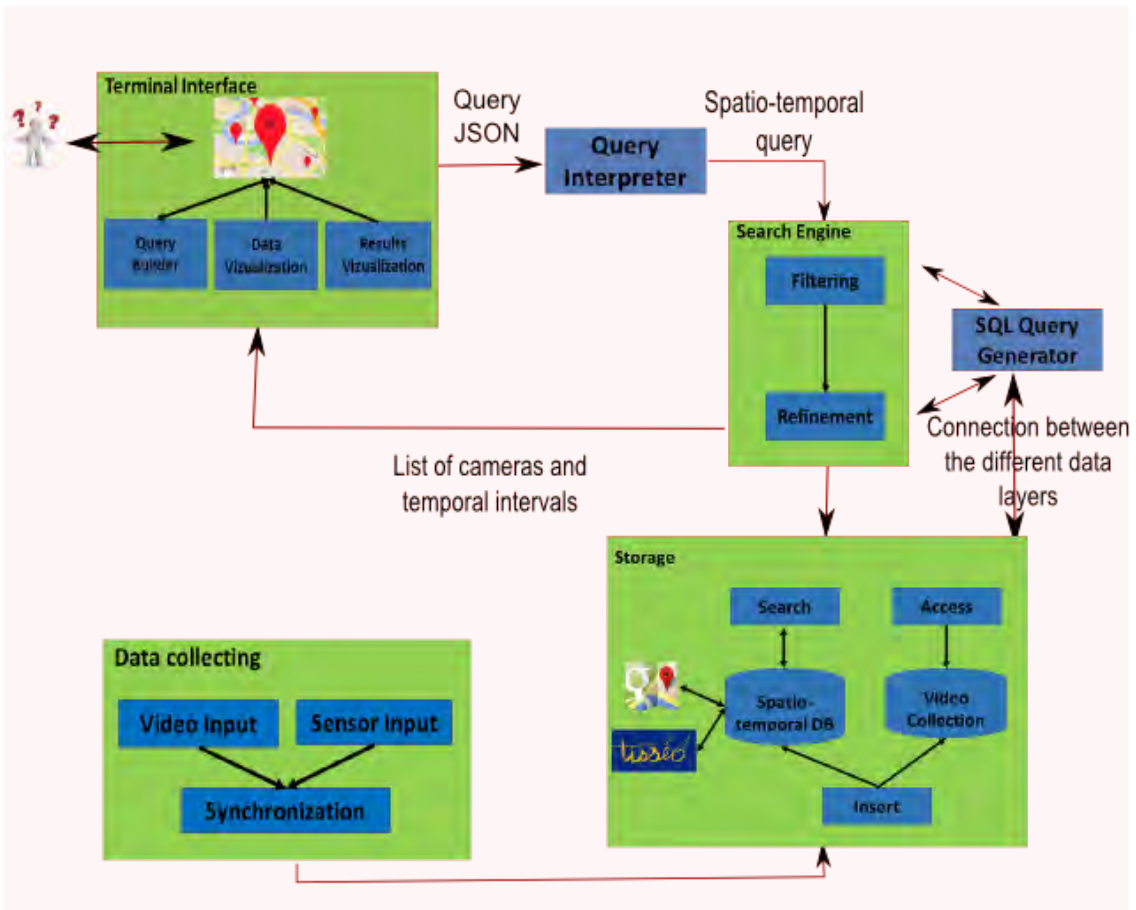


FIGURE 5.1 – Architecture du framework proposée

### 5.2.1 Le module de collecte de données

Le module au niveau duquel les données sont collectées est composé des dispositifs de capture des contenus vidéo et des informations spatio-temporelles issues des capteurs associés aux dispositifs (e.g., capteur GPS, boussole, accéléromètre).

Le problème principal rencontré dans l'acquisition de données issues des différents capteurs est la synchronisation des différents flux d'information. Prenons un exemple : supposons une caméra qui enregistre du contenu visuel avec un framerate de 30 frames par secondes, un capteur GPS qui enregistre des données de localisation avec une fréquence d'une position par seconde et une boussole qui enregistre la direction à chaque modification de celle-ci. Dans ce cas, des opérations de synchronisation pour palier le problème des différentes fréquences d'échantillonnages doivent être mis en place en se basant sur des techniques comme l'interpolation ou le calcul de moyenne [Wåhslén et al., 2010]. Un autre problème qui se pose et qui concerne la collecte de données est le format de stockage de métadonnées, format qui peut être incorporé dans la vidéo ou géré séparément [Sèdes, 1998]. Dans une perspective de développement d'un framework générique qui puisse faciliter l'interopérabilité des systèmes de vidéosurveillance (e.g., le traitement des données issues du système de la RATP par les agents de la SNCF) la solution optimale est d'arriver à une généralisation de la mise en oeuvre de la norme ISO22311 avec présence de métadonnées normalisées (voir chapitre 3). Des projets au niveau national sont menés pour la mise en place de cette généralisation parmi lesquels le projet METHODEO [Goudou et al., 2013] auquel nous avons participé comme présenté dans le chapitre 3 de ce manuscrit.

Dans le cadre de notre expérimentation, nous collectons les données provenant du GPS, de la boussole, de l'accéléromètre et de la caméra et nous les synchronisons en utilisant une estampille temporelle. Nous avons utilisé pour la collection des données une application développée par nous mêmes et deux applications commerciales (GPSLogger<sup>1</sup>, Accelerometer Monitor<sup>2</sup>, Accelerometer Pro<sup>3</sup>). Ces applications enregistrent les modifications dans le temps des données issues des capteurs embarqués dans les smartphones. Plus précisément elles enregistrent des séquences de paires (valeur, timestamp) ou la valeur est soit une position GPS (Lat, Long), soit un angle d'orientation par rapport au pôle Nord, soit un angle d'inclinaison calculé à partir des données enregistrés par l'accéléromètre (X,Y,Z,tilt) (cf. à la Fi-

---

1. <https://play.google.com/store/apps/details?id=com.mendhak.gpslogger&hl=fr>  
2. [https://play.google.com/store/apps/details?id=com.lul.accelerometer&hl=fr\\_FR](https://play.google.com/store/apps/details?id=com.lul.accelerometer&hl=fr_FR)  
3. [https://play.google.com/store/apps/details?id=com.a10.acmeter.donate&hl=fr\\_FR](https://play.google.com/store/apps/details?id=com.a10.acmeter.donate&hl=fr_FR)

Date	Lat	Long	Compass	Accuracy	Accelerometer(X,Y,Z,tilt)
2014-07-19 17:04:42	43.6047116	1.4453106	1.6938289	8.000000	0.6276256,9.963556,2.6674087,75.0407548945918
2014-07-19 17:04:43	43.6047116	1.4453106	-	8.000000	-
2014-07-19 17:04:44	43.6047116	1.4453106	-	12,00000	-
2014-07-19 17:04:45	43.6046832	1.4453513	358.60864	8.000000	0.588399,9.92433,2.3143694,76.89535310673617
2014-07-19 17:04:48	43.6046832	1.4453513	-	8.000000	-
2014-07-19 17:04:49	43.6046832	1.4453513	-	8.000000	-
2014-07-19 17:04:50	43.6046651	1.4453764	-	8.000000	-
2014-07-19 17:04:51	43.6046651	1.4453764	-	8.000000	-
2014-07-19 17:04:52	43.6046651	1.4453764	-	8.000000	-
.....					
2014-07-19 17:05:15	43.6046625	1.4453068	0.077175155	8.000000	0.4314926,10.081236,1.7651969,80.07725117353431

FIGURE 5.2 – Extrait des données capturées par les capteurs

gure 5.2). Le but de notre expérimentation n'étant pas liée à la problématique de la synchronisation des données provenant des différents capteurs, nous avons fixé a priori une fréquence d'échantillonnage d'une seconde pour le GPS et comme tous ces les capteurs sont associés au même dispositif nous avons utilisé un horloge unique (synchronisée au temps universel) pour tous les capteurs. Les données provenant de la boussole et l'accéléromètre sont enregistrées seulement quand des modifications interviennent. Un extrait des données capturées au niveau du module de capture est illustré dans la Figure 5.2. Ces données sont ensuite parsées pour une insertion dans les tables correspondantes de notre modèle de données présenté dans le chapitre 4. Cette opération est réalisée par le module *Insert* de notre architecture.

En ce qui concerne les données décrivant le réseau de transport nous utilisons les données décrivant le réseau de transport public de Toulouse mises à disposition dans le cadre du projet Toulouse Open Data<sup>4</sup>. Ces données décrivent les caractéristiques spatiales (stations et formes) et temporelles (horaires de passage) des 103 lignes de bus qui desservent la ville de Toulouse. Le format de description de ces données (GTFS) et les améliorations que nous avons apportées à celui ci sont décrites dans le chapitre précédent. Un extrait des données de transport sauvegardées dans notre format de stockage est illustré dans la Figure 5.3. Ces données décrivent la structure du réseau de bus de Toulouse, plus précisément pour chaque ligne, les positions des stations et les horaires de passage des bus par les stations respectives. En plus, chaque ligne est associée à une liste de points (positions GPS) qui permette de décrire la forme exacte (projetée sur le réseau routier) de la ligne respective (la description détaillée de la modélisation du réseau de transport et donnée dans la section 4.3.2 du

4. <http://data.toulouse-metropole.fr/>

SHAPE_ID	STOP_ID1	STOP_ID2	STOP_LAT1	STOP_L...	STOP_LAT2	STOP_L...	SH...	SHA...	SG_MBR	...
4503603927006462	3377699720881654	3377699720888732	43,4419	1,59884	43,4364	1,5989	530	548	[MDSYS.SDO_GEOMETRY]	24
4503603927006462	3377699720888331	3377699720888332	43,5074	1,56874	43,5048	1,56701	288	297	[MDSYS.SDO_GEOMETRY]	13
4503603927006462	3377699720888358	3377699720888355	43,4585	1,60385	43,4557	1,60962	394	404	[MDSYS.SDO_GEOMETRY]	17
4503603927006462	3377699722882623	3377699720884561	43,5396	1,51261	43,5352	1,52015	106	133	[MDSYS.SDO_GEOMETRY]	5
4503603927006462	3377699720884620	3377699720881868	43,5183	1,54596	43,5179	1,54204	208	218	[MDSYS.SDO_GEOMETRY]	7
4503603927006462	3377699722609764	3377699720881654	43,4467	1,59778	43,4419	1,59884	515	530	[MDSYS.SDO_GEOMETRY]	23
4503603927006462	3377699720888735	3377699720888331	43,5108	1,57007	43,5074	1,56874	276	288	[MDSYS.SDO_GEOMETRY]	12
4503603927006462	3377699723426206	3377699720888733	43,5131	1,5568	43,5108	1,56487	251	263	[MDSYS.SDO_GEOMETRY]	10
4503603927006462	3377699720885706	3377699722609764	43,4486	1,60637	43,4467	1,59778	482	515	[MDSYS.SDO_GEOMETRY]	22
4503603927006473	3377699722609766	3377699720888736	43,5073	1,56908	43,511	1,57013	272	284	[MDSYS.SDO_GEOMETRY]	13
4503603927006473	3377699720888736	3377699720888734	43,511	1,57013	43,511	1,56481	284	297	[MDSYS.SDO_GEOMETRY]	14
4503603927006473	3377699723426206	3377699723426204	43,5131	1,5568	43,511	1,55522	309	315	[MDSYS.SDO_GEOMETRY]	16
4503603927006473	3377699720888965	3377699720881875	43,4534	1,61314	43,4535	1,62103	78	106	[MDSYS.SDO_GEOMETRY]	5
4503603927006473	3377699720881868	3377699720884619	43,5179	1,54204	43,5181	1,54636	347	355	[MDSYS.SDO_GEOMETRY]	18
4503603927006473	3377699722193944	3377699722609747	43,5489	1,5062	43,5478	1,51366	531	574	[MDSYS.SDO_GEOMETRY]	24
4503603927006473	3377699720884619	3377699720884560	43,5181	1,54636	43,5353	1,52002	355	445	[MDSYS.SDO_GEOMETRY]	19
4503603927006473	3377699720881874	3377699720881907	43,4819	1,57225	43,489	1,56907	209	224	[MDSYS.SDO_GEOMETRY]	10
4503603927026678	3377699720888322	3377699720881651	43,4792	1,52528	43,4866	1,53144	133	165	[MDSYS.SDO_GEOMETRY]	11
4503603927026678	3377699720885792	3377699720885795	43,4632	1,56861	43,4591	1,57443	211	234	[MDSYS.SDO_GEOMETRY]	14

FIGURE 5.3 – Extrait de la base de données du réseau de transport

Chapitre 4).

Pour les données décrivant le réseau routier, nous utilisons une ressource extérieure, l'API Google Maps V3<sup>5</sup>. L'API Google Maps offre des informations complètes, l'usage est très répandu et c'est facile à utiliser. Comme nous n'avons pas besoin d'autre types d'informations comme les noms des bâtiments ou d'autre types d'objets (qui sont mis à disposition par d'autres outils comme Open Street Maps), notre choix s'est avéré suffisant. Néanmoins, d'autre solutions externes peuvent être envisagées. Un facteur important dans notre choix a été le fait que les données concernant le réseau de transport en commun de Toulouse mises à disposition dans le cadre du projet Toulouse Open Data sont elles aussi dans le format GTFS qui est un des formats proposé par Google pour la description des réseaux de transport. Nous nous sommes servis de l'API Google Maps pour le développement de l'interface utilisateur interactive qui est décrite par la suite.

5. <https://developers.google.com/maps/documentation/javascript/>



FIGURE 5.4 – Visualisation des géométries des champs de vue des caméras

## 5.2.2 Le module interface utilisateur

Le module *Terminal Interface* de l'architecture proposée dans la Figure 5.1 permet de saisir ou de définir les points composant les segments de la requête à partir de la carte interactive, de visualiser des données comme les trajets des bus (voir chapitre 4) ou les géométries des champs de vue d'une caméra (voir Figure 5.4) et de visualiser les résultats. Ces derniers sont présentés comme une liste d'extraits vidéos provenant des caméras sélectionnées par l'algorithme de recherche. L'avantage de cette interface est le fait de pouvoir visionner les extraits vidéos et voir sur la carte les prises de vues des caméras sources.

La requête peut être directement renseignée dans un fichier structuré au format JSON ou peut être exprimée à partir de l'interface graphique de notre framework illustrée dans la Figure 5.5. Pour chaque segment de la requête on définit un système de référence et des positions symboliques (par énumération) ou géométriques (par énumération ou interaction avec la partie carte de notre interface) (voir Figure 5.6). Par exemple, dans la Figure 5.5, on définit un segment exprimé par rapport au réseau routier (Identifiant du système de référence "RRTLSE") et composé de

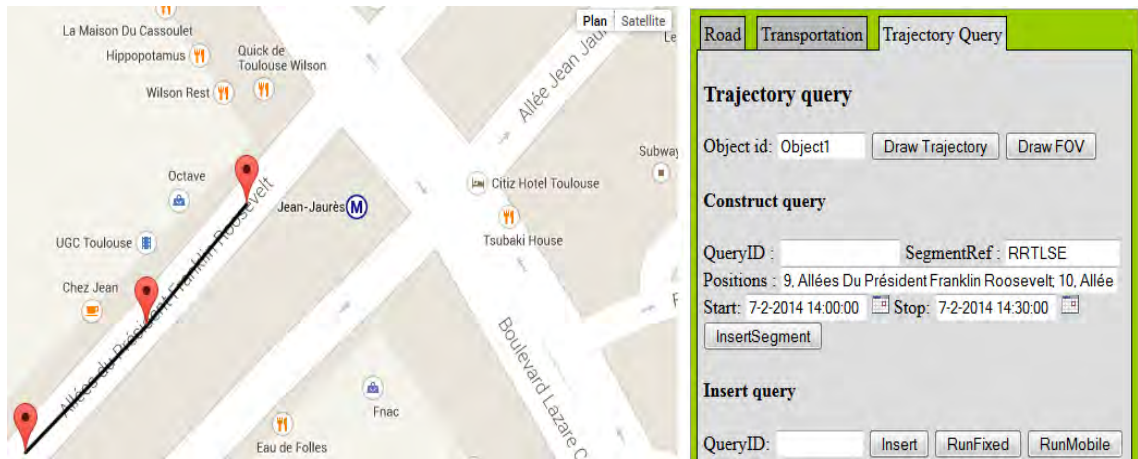


FIGURE 5.5 – Interface de définition des segments de la requête

trois positions : 9, Allées Du Président Franklin Roosevelt, 10, Allées Du Président Franklin Roosevelt et 12, Allées Du Président Franklin Roosevelt. La structuration de ce segment de requête en format JSON est illustrée par la Figure 5.6. La requête sera ensuite "interprétée" par le module d'interprétation de la requête et le résultat (la séquence de segments spatiaux et l'intervalle de temps) sera stocké dans la base de données. L'avantage d'utiliser une table stockée en mémoire pour les requêtes est que les segments peuvent ensuite être indexés par des index spatiaux [Stojanovic et al., 2005].

Par la suite nous allons décrire la façon dont la requête hybride est transformée en requête spatio-temporelle par l'interpréteur.

### 5.2.3 Le module d'interprétation de la requête

Ce module prend en entrée une requête JSON répondant au template qui est illustré par l'intermédiaire d'un exemple dans la Figure 5.6. Pour définir le schéma ou le template de notre requête, plus précisément les règles de dénomination d'une position, d'un enchaînement de positions (segment) et d'un enchaînement de segments (trajectoire) nous allons nous appuyer sur le standard JSON<sup>6</sup> (JavaScript Object Notation) qui définit un standard d'échange qui est utilisé à large échelle dans les dernières années par les technologies web, y compris par les plateformes de gestion

6. <http://www.json.org/jsonfr.html>



```

{
  "query": {
    "spatial": [
      {
        "WGS84": [
          [43.6060, 1.44837],
          [43.6067, 1.44759],
          [43.6071, 1.44721]
        ]
      },
      {
        "RTTSE": [
          {"Allées du Président Franklin Roosevelt":9}, {"Allées du Président Franklin Roosevelt":10}, {"Allées du Président Franklin Roosevelt":12}
        ]
      },
      {
        "RTTSE": [
          {"Bus16":"Jean Jaures"}, {"Bus16":"Jean d'Arc"}
        ]
      }
    ],
    "temporal": {
      "start": "07-02-2014 14:00:00",
      "end": "07-02-2014 14:30:00"
    }
  }
}

```

Annotations dans l'image :

- Référence système géodésique (pointe vers "WGS84")
- Positions géométriques (pointe vers les coordonnées dans "WGS84")
- Référence réseau routier de Toulouse (pointe vers "RTTSE" avec routes)
- Positions symboliques (pointe vers les noms de routes)
- Référence réseau de transport de Toulouse (pointe vers "RTTSE" avec bus)
- Positions symboliques (pointe vers les noms de bus)
- Intervalle de temps (pointe vers "start" et "end")

FIGURE 5.6 – Exemple de fichier de requête JSON

des données géographiques comme Google Maps.

La schématisation de cette brique de notre framework est illustrée dans la Figure 5.7. Ce module se base sur le modèle de données (notamment les couches réseau routier et réseau de transport) pour interpréter des segments de trajectoire contenant des positions géométriques et symboliques exprimées par rapport à des systèmes de références différents afin de traduire cette requête dans une séquence de segments spatiaux (des géométries exprimées par rapport au système géodésique).

Par la suite, nous allons présenter le module qui réalise la recherche proprement dite et celui qui est en charge du stockage.

## 5.2.4 Le module de stockage et le module de recherche

Nous allons commencer cette section par la présentation du module de stockage (le module *Storage* de la Figure 5.1) qui comprend : (1) la collection des vidéos issues des caméras et qui contient les fichiers vidéos nommés selon la règle (décrite par la norme ISO22311 présentée dans la section 3.5) : <track\_nom>\_video\_<type\_of\_encoding>\_<timestamp>.<extension>, où

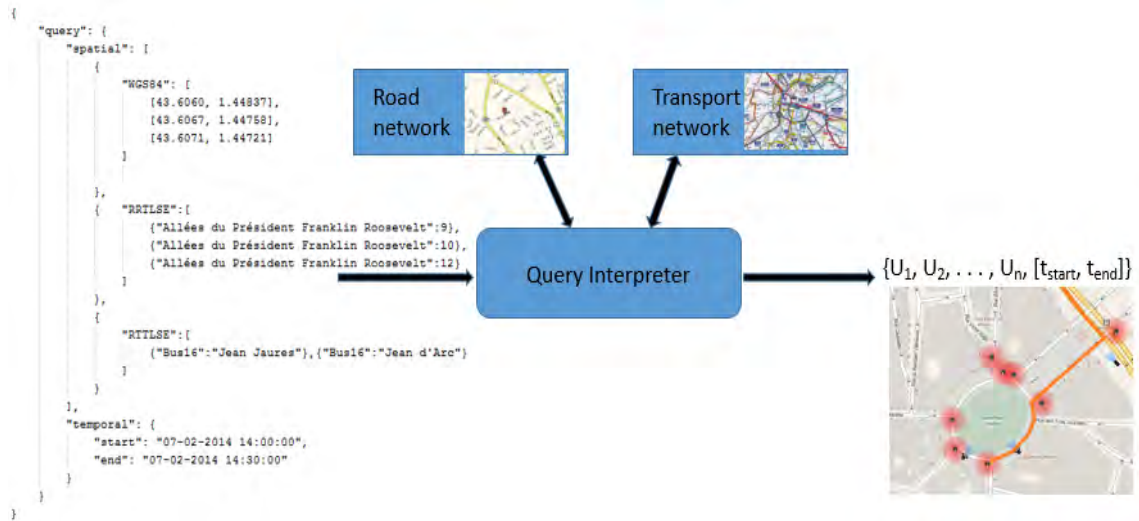


FIGURE 5.7 – Schéma du module d'interprétation de la requête

$\langle \text{track\_nom} \rangle$  est le nom de la piste vidéo,  $\langle \text{type\_of\_encoding} \rangle$  désigne le type d'encodage de la vidéo,  $\langle \text{timestamp} \rangle$  représente l'estampille temporelle du début de la vidéo et  $\langle \text{extension} \rangle$  est l'extension du fichier. Un fichier "index" est utilisé pour faire le mapping des sources (caméras) et des pistes (fichiers vidéos) issues de chaque caméra; (2) la base de donnée spatio-temporelle qui contient les informations décrivant les différentes couches présentées dans le chapitre 4 et qui a été implémentée en utilisant Oracle Spatial<sup>7</sup>.

Le module *Search Engine* implémente les algorithmes de recherche présentés dans le chapitre 4 avec les deux étapes *Filtering* et *Refinement*. La communication entre ce module et le module de stockage est réalisée via le module *SQL Query Generator*. Le module *Access* fournit des méthodes pour accéder le contenu vidéo stocké dans la collection vidéo. Ce module reçoit la liste des résultats, ou chaque tuple du résultat a la forme (CameraId,StartTime,EndTime) et implémente le service `extract(CameraId,StartTime,EndTime)` qui extrait le segment de la vidéo filmée par la caméra ayant l'identifiant CameraId entre StartTime et EntTime. Ce module n'est pas implémenté pour l'instant dans notre système. La vidéo entière est retournée, l'utilisateur devra chercher l'intervalle indiqué. Néanmoins, ce point est facile à développer et à intégrer dans notre application.

Dans la suite de ce chapitre nous allons illustrer la faisabilité et l'apport de notre

7. <http://www.oracle.com/fr/products/database/options/spatial/index.html>

proposition par la présentation d'un cas d'utilisation réalisé à l'aide de notre framework.

### 5.3 Exemple de cas d'utilisation réalisé

Afin de valider notre framework, nous allons présenter un des scénarios que nous avons mis en place afin d'illustrer les fonctionnalités et la faisabilité de notre proposition. Voici la description de notre scénario. Nous avons simulé un système de vidéosurveillance qui est composé de cinq caméras fixes et deux caméras mobiles disposées Route de Narbonne à Toulouse. Dans nos expérimentations nous nous sommes servis de plusieurs dispositifs mobiles comme des smartphones : un Samsung Galaxy S4, un Samsung Galaxy S5, un Motorola RAZR i XT890 et une tablette Asus FonePad 7. En utilisant ces dispositifs nous avons collecté 10 vidéos (avec la caméra dans une position fixe ou en mouvement) d'une durée totale approximative d'une heure et demie. L'enregistrement de ces vidéos à été fait soit avec les caméras en position fixe (dans une disposition montrée par la Figure 5.10) soit en mobilité (enregistrements effectués avec deux caméras situées à l'intérieur des bus pour simuler les caméras de vidéosurveillance embarquées).

Les caméras fixes ont la disposition géographique illustrée dans la figure 5.9. Des captures d'écran des scènes filmées par les cinq caméras sont illustrés dans la Figure 5.10. Les caméras C1 et C2 changent de champ de vue chaque heure et elles sont en train de surveiller deux aires différentes : la caméra C2 surveille le passage piétons (image C2a de la Figure 5.10) et l'entrée dans une résidence (image C2b de la Figure 5.10) et la caméra C1 surveille la Route de Narbonne au niveau de l'entrée dans la grande station de bus Paul Sabatier (image C1a de la Figure 5.10) et une petite entrée dans le campus qui donne sur la rue (image C1b de la Figure 5.10). La Figure 5.8 montre le changement des champs de vue des caméras C1 et C2, la première partie de la figure représente les champs de vue avant et la deuxième ceux d'après le changement de l'orientation des caméras. Le changement des caractéristiques (position, orientation, accéléromètre) des caméras (fixes et mobiles) est enregistré par les applications mentionnées dans la section 5.2.1.

Le scénario est le suivant : une victime simulée prend un repas dans un restaurant (Subway) sur la Route de Narbonne à Toulouse le dimanche, 29 septembre, vers 18 :45. Elle sort du restaurant et marche vers la station de métro Paul Sabatier. La, elle se rend compte que son portefeuille a disparu. La requête soumise à notre système dans ce cas la est illustrée dans la Figure 5.9 et est représenté par le segment



FIGURE 5.8 – Champs de vue des caméras avant 19h (a) et après 19h (b)

noir et l'intervalle de temps [28-9-2014 18 :50 :00, 28-9-2014 19 :15 :00] (par défaut dans l'interface de l'application toutes les vidéos de la base sont affichées).

Suite à la première étape de filtrage, la caméra C5 est éliminée car sa distance par rapport au segment de la requête dépasse la distance de visibilité maximale des caméras de la base de données. Dans notre expérimentation, nous avons fixé la distance de visibilité des caméras à 200m car nous avons filmé en ville dans une zone avec beaucoup de constructions et dans un tel contexte, une caméra ne peut "voir" très loin. Pour les autres caméras, les intersections des géométries des champs de vues et de la trajectoire pour l'intervalle de temps donné, selon l'algorithme présenté dans la section 4.5 du chapitre précédent sont évaluées. Bien que la caméra C4 a une distance de visibilité maximale permettant de filmer la trajectoire, son champ de vue n'intersecte jamais la trajectoire requête donc elle n'apparaîtra pas dans le résultat. Le champ de vue de la caméra C3 intersecte sur toute la période de la requête la trajectoire donc le tuple (C3, [18 :50,19 :15]) fera partie du résultat. Le champ de vue de la caméra C2 intersecte la trajectoire à partir de 19h donc le tuple (C2, [19 :00,19 :15]) sera renvoyé. Par contre, la caméra C1 "voit" la trajectoire avant 19h, du coup le tuple (C1, [18 :50,19 :00]) fera partie de la liste des résultats. La liste des résultat est ordonnée en fonction de la distance des caméras de la liste par rapport au point de départ (dans notre exemple le point de départ est le restaurant Subway donc la caméra C3 est la plus proche).

Le premier élément de la liste des résultats est (C3, [18 :50,19 :15]). En regardant la vidéo correspondante, on observe que à partir de 18 :57 la victime marche et un homme habillé en bleu marche derrière elle (quelques cadres de cette vidéo sont montrés dans la Figure 5.11).

En passant au deuxième résultat retourné (C2, [19 :00,19 :15]), à partir de la minute 19 :01 les deux personnes apparaissent dans les images. La Figure 5.12 montre les cadres les plus importants issues des enregistrements de cette caméra, le moment de l'apparition des personnages dans la scène filmée, le moment où le vol se produit (on peut voir dans les images le moment où le suspect habillé en bleu commet l'infraction) et le moment où la victime et l'agresseur partent vers la droite donc vers les station de bus et de métro Paul Sabatier. Étant donné que le dernier résultat retourné (C1, [18 :50,19 :00]) concerne un intervalle de temps qui se passe avant les faits que nous venons de regarder sur les autres deux caméras, cette caméra ne contient pas d'images pertinentes.

Par la suite, une deuxième requête sera soumise qui visera le segment de rue entre le passage piétons et la station de métro Paul Sabatier et l'intervalle horaire [19 :02,

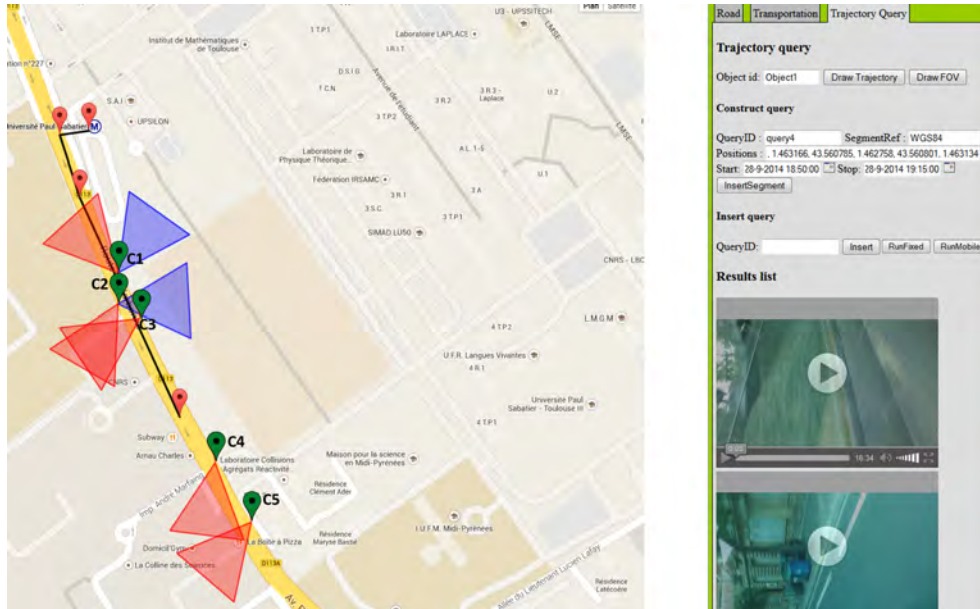


FIGURE 5.9 – Disposition de notre réseau de caméras

19 :32] afin de rechercher des caméras mobiles qui ont possiblement filmé l'agresseur dont les caractéristiques sont maintenant connues et qui pourront donc fournir des images claires qui permettront l'identification de la personne en cause. L'algorithme de recherche des caméras mobiles est ensuite exécuté. Cet algorithme est présenté dans la section 4.5.2 du Chapitre 4 et il implémente l'opérateur décrit dans la section 4.4.2. Cet opérateur sélectionne les objets mobiles dont la trajectoire a intersecté la trajectoire de la requête dans l'intervalle de temps donné et donne comme résultat à l'utilisateur les identifiants des caméras associées aux objets sélectionnés, car elles sont susceptibles d'avoir filmé la trajectoire requête dans l'intervalle de temps visé. Nous avons utilisé comme collection d'objets mobiles les données liées aux trajets des bus de Toulouse. En conséquence le résultat de notre requête va être la liste des identifiant des bus qui sont passés par le segment de rue de la requête dans l'intervalle [19 :02, 19 :32].

Les identifiants des bus donnés comme résultat sont : 4503603927976693 (qui correspond au bus numéro 2 qui passe à 19 :30 par la station Paul Sabatier), 4503603927986760 (qui correspond au bus numéro 34 qui passe à 19 :30 par la station Paul Sabatier) et 4503603927985747 (qui correspond au bus numéro 78 qui passe à 19 :30 par la station Paul Sabatier). La raison pour laquelle nous avons une liste assez "simple" de résultats est que le jour choisi pour l'expérimentation (28/09/2014) est



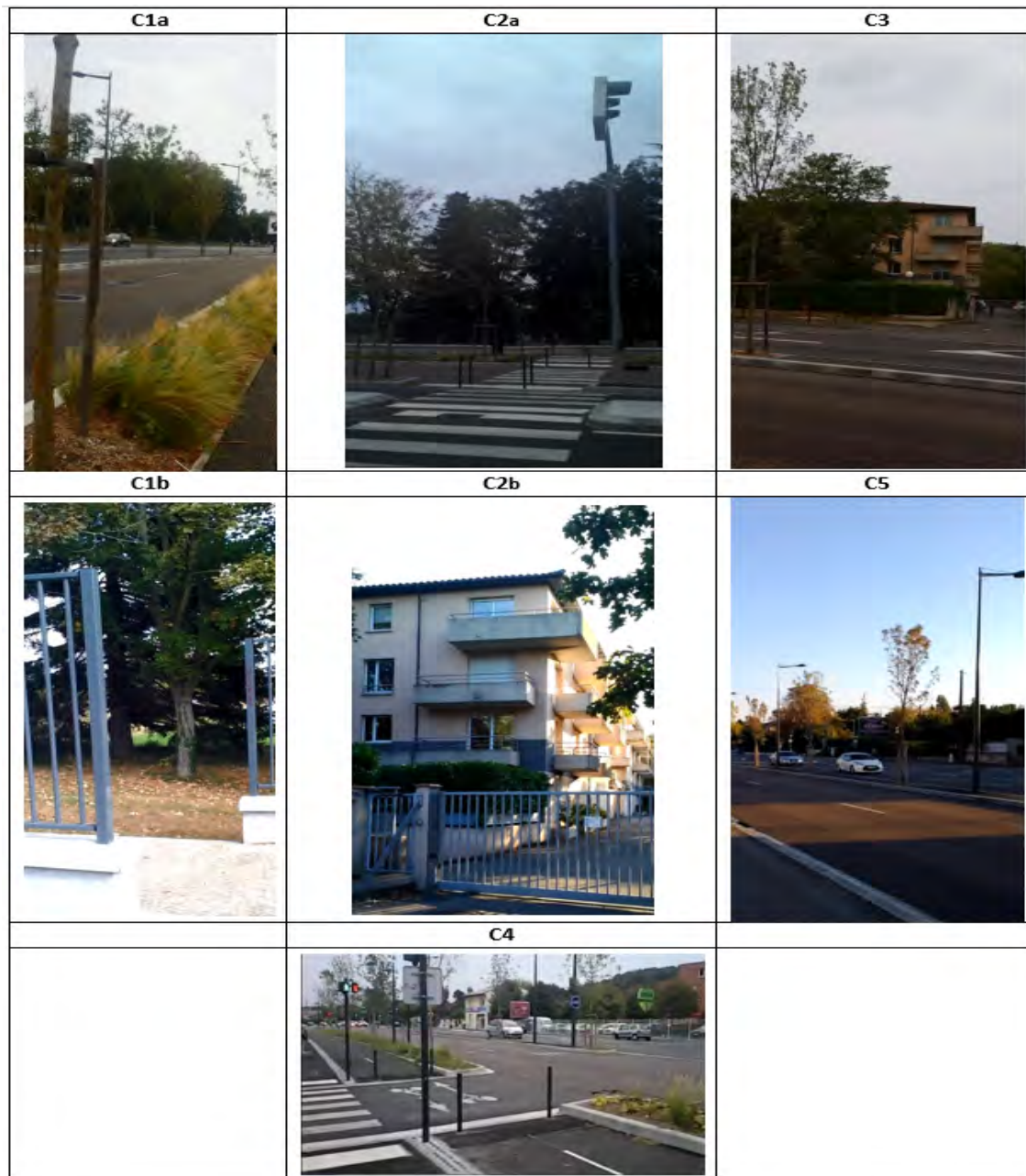


FIGURE 5.10 – Scènes filmées par les caméras de notre réseau



FIGURE 5.11 – Des cadres vidéo filmés par la caméra C3

un dimanche et les bus de Toulouse ont des horaires beaucoup moins denses (3 sur 7 bus qui circulent par la station Paul Sabatier ne passent pas le dimanche).

Nous avons deux caméras installées dans le bus numéro 2 : une devant (voir la Figure 5.14) et une sur le côté (voir la Figure 5.13). En regardant les enregistrements vidéos issus de ces deux caméras dans l'intervalle de la requête (l'intervalle retourné est tout l'intervalle de la requête car Paul Sabatier est station de départ pour la ligne 2 du coup il n'y a pas de position précédant celle-ci dans les trajectoires du bus) on observe que les deux ont filmé le suspect et que seulement les images filmées par la caméra de devant peuvent permettre l'identification de la personne.

## 5.4 Expérimentations concernant le réseau de transport en commun

La collection que nous avons utilisée pour tester l'algorithme de sélection des caméras (avec les segments de la requête et l'intervalle de temps concernés) installées dans les bus qui ont traversé les segments de la requête est constitué par les données qui décrivent le réseau de transport en commun de Toulouse (en format GTFS)





FIGURE 5.12 – Des cadres vidéo filmés par la caméra C2



FIGURE 5.13 – Des cadres vidéo filmés par la caméra installée au latéral du bus numéro 2



FIGURE 5.14 – Des cadres vidéo filmés par la caméra installée dans la partie devant du bus numéro 2

qui ont été publiées dans le cadre du projet Toulouse Open Data. Quelques chiffres concernant ces données sont présentés dans la Figure 5.15. Le réseau de transport de Toulouse comprend 103 lignes de bus. Au cours d'une année ces bus se déplacent selon 336 trajets qui contiennent en moyenne approximativement 25.6 stations par trajet. La forme de chacun de ces trajets est décrite par approximativement 350 points. La variation du nombre de sections de trajets en fonction de la ligne de bus est illustrée dans la Figure 5.16.

Le but de cette expérimentation a été de tester le comportement de notre algorithme de sélection des caméras mobiles qui ont potentiellement pu filmer la trajectoire de la requête en fonction de la taille de la requête et des trajets ciblés ainsi que du nombre d'objets mobiles de la collection et de leur nombre de positions. Afin de réaliser des tests plus importants, nous avons généré des points de trajectoire à partir des données existant dans le format GTFS. Comme expliqué dans le chapitre précédent, le format GTFS fournit une description du réseau de transport en commun en se basant sur les éléments suivants : chaque ligne est composée par une séquence de stations, une liste de positions et des listes des horaires de passage des différents bus à chaque station. Comme les stations sont situées à des distances assez importantes, chaque tronçon (défini par deux stations consécutives) se voit associé une liste de points qui vont décrire sa forme (le mapping sur le réseau routier comme présenté dans la section 4.3.2 du chapitre précédent) que nous allons nommer points forme. Nous considérons que les bus se déplacent avec une vitesse constante par tronçon égale à la somme des distances entre chaque deux points forme consécutifs et l'intervalle de temps défini par les temps d'arrivée des bus dans les deux stations qui définissent le tronçon. En utilisant cette vitesse calculée, nous avons généré des points trajectoire en associant les positions des points forme à des timestamps calculés. Cela a amené à la génération d'environ 3 millions de positions.

Ensuite nous réalisons des tests de performance afin d'obtenir des résultats utiles qui montrent que notre méthodologie est applicable dans une configuration de système de gestion des trajectoires des objets mobiles qui se déplacent dans un réseau de transport en commun proche du "monde réel". Tous les tests ont été réalisés sur un processeur Intel Core i7-3687U CPU 2.6 GHz avec 8 Go RAM et un système d'exploitation Windows 7. Le SGBD utilisé est Oracle 11g.

Nous avons choisi pour notre test deux trajectoires requêtes illustrées dans la Figure 5.17 d'une distance de 2,5 km. La Figure 5.17a représente une trajectoire au centre ville de Toulouse et la Figure 5.17b représente une trajectoire dans un quartier situé dans l'extrémité sud de Toulouse où le réseau de bus est moins dense. Nous avons divisé chacune de ces trajectoires en 10 segments (la division a été faite de façon

Nombre de routes	103
Nombre de trajets	336
Nombre moyen de sections par trajet	24.8
Nombre moyen de points shape par trajet	349.6
Nombre moyen de stations par trajet	25.57
Nombre total d'objets mobile	16700
Nombre total de points de trajectoire	~ 3 millions

FIGURE 5.15 – Chiffres décrivant le dataset Tisseo

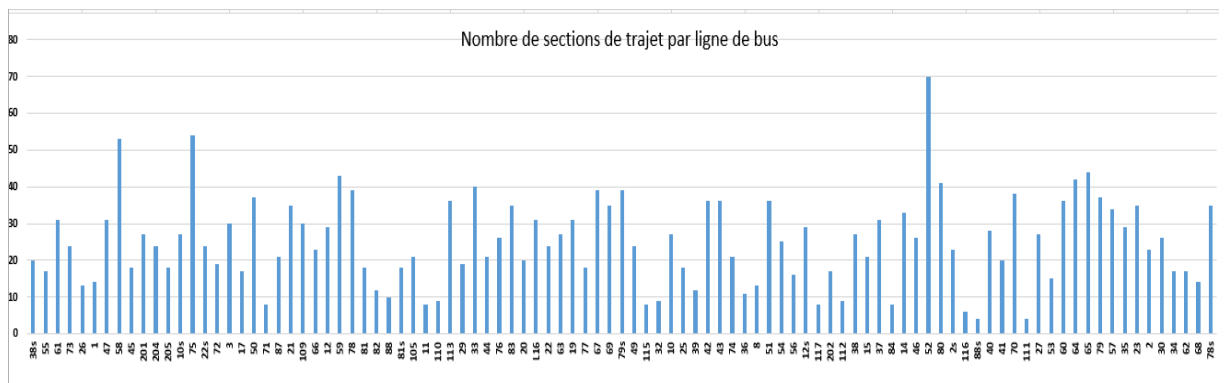


FIGURE 5.16 – Distribution du nombre de sections de trajet par ligne de bus



FIGURE 5.17 – Les deux trajectoires que nous allons utiliser pour le test de notre algorithme

similaire entre les deux trajectoires). A partir de ces 10 segments nous avons généré 10 requêtes en commençant avec une requête d'un segment et en rajoutant à chaque fois un segment de plus.

Le but a été de tester le temps de réponse de notre algorithme qui intersecte la trajectoire de la requête avec les points des trajectoires des objets mobiles de la base (présenté dans la section 4.5.2 du Chapitre 4) en fonction du nombre de segments de la requête et du nombre d'objets mobiles (des bus) concernés. Les résultats obtenus sont illustrés dans la Figure 5.17. Nous observons que notre algorithme donne des temps de réponse assez bons vu le nombre de points trajectoire élevé de la base. Le graphique 5.19 montre une grande différence entre les temps de réponse obtenus pour les deux requête. Cette différence est due à la variabilité de la densité du réseau de transport entre différentes régions de la ville. Le réseau est beaucoup plus dense au centre ville (voir le nombre d'objets retourné par les requêtes) ce qui a comme conséquence une augmentation du temps de réponse.

## 5.5 Conclusion

Dans ce chapitre nous avons décrit une architecture de framework pour l'interrogation des métadonnées spatio-temporelles associées aux contenus vidéos issus des caméras fixes et mobiles que nous avons proposé. L'architecture est construite autour des modules suivants : stockage des différentes couches du modèle de données, l'exécution des algorithmes mais aussi l'interface avec les utilisateurs. L'illustration

Requête	Nombre de segments	Temps de réponse (en secondes)	Nombre d'objets mobiles (bus) retournés
Requete1 – Quartier	1	5	5
	2	6	5
	3	8	5
	4	13	12
	5	17	12
	6	20	12
	7	23	12
	8	29	19
	9	32	19
	10	36	19
Requete2 - Centre Ville	1	9	12
	2	11	12
	3	12	12
	4	31	23
	5	45	42
	6	48	42
	7	50	42
	8	51	42
	9	73	64
	10	75	64

FIGURE 5.18 – Resultats du test

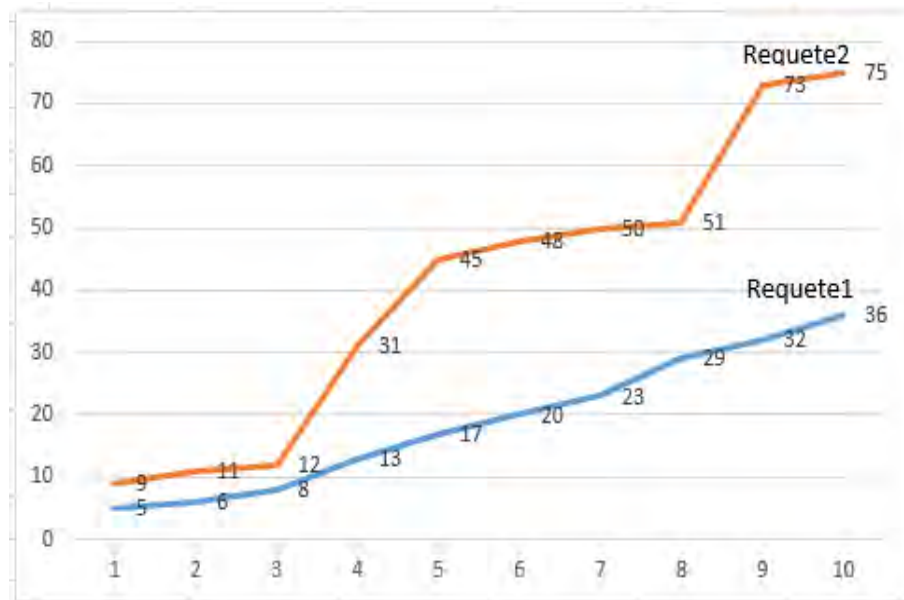


FIGURE 5.19 – Graphique de comparaison entre Requête 1 et Requête 2

du cas d'utilisation que nous avons implémenté montre la faisabilité et les bénéfices de la solution proposée dans le contexte de la vidéosurveillance. Notre approche est complémentaire des approches d'analyse de contenu vidéo (qui sont hors du cadre de ce travail) et ensemble peuvent constituer une solution puissante de filtrage des vidéos tant recherchée par les opérationnels de la vidéosurveillance. Nos expérimentations montrent que les algorithmes proposés donnent des résultats prometteurs sur une base de donnée contenant plus de 3 millions d'objets mobiles et sur une machine de taille modeste, des améliorations pouvant être amenées par des techniques de calcul parallèle ou des techniques plus poussées d'optimisation de la base de données.





# Chapitre 6

## Conclusions générales et perspectives

### 6.1 Synthèse des propositions

Nos travaux présentés dans ce mémoire se situent dans le contexte général du filtrage des contenus vidéos basé sur des métadonnées associées aux contenus. Plus concrètement, le domaine d'application de ces travaux relève de l'aide aux opérateurs humains de vidéosurveillance dans l'analyse manuelle d'extraits vidéos particuliers (e.g., recherche individu, objet perdu) afin de faciliter leur tâche, qui est souvent lourde et sensible aux erreurs, par la mise en place d'outils qui visent la réduction de l'espace et implicitement du temps de recherche.

Après une étude des systèmes de vidéosurveillance dans le cadre de plusieurs projets de recherche menés en collaboration avec la Police Nationale, RATP, SNCF, Thalès Sécurité, nous avons fait les observations suivantes qui ont servi d'hypothèses ou d'analyse du besoin pour les contributions de ce travail :

1. le grand nombre de systèmes de vidéosurveillance gérés par des entités différentes ayant leur propres formats de données et de métadonnées a comme conséquence un manque d'interopérabilité entre les sorties des différents systèmes (e.g., il est difficile pour la police d'utiliser directement les enregistrements issus du système d'un centre commercial) ;
2. le point de départ de toute recherche dans une collection de vidéo surveillance est l'information spatiotemporelle ;

3. toute trajectoire, d'objet ou d'individu, par exemple le trajet d'une victime ou d'un agresseur avant et après les faits, doit pouvoir être reconstituée et, en se basant sur celle-ci, on essaye d'extraire les images permettant d'identifier le(s) objet(s), par exemple le(s) agresseurs, suspects, pick-pockets, etc. ;
4. cette trajectoire peut être exprimée par rapport à différents systèmes de référence (indoor ou outdoor) : système géodésique, réseau routier, réseau de transport, plan d'un bâtiment ;
5. la qualité des enregistrements étant faible et les conditions d'acquisition étant très diverses, la performance des outils d'analyse du contenu vidéo est souvent insuffisante ;
6. l'inexistence des métadonnées comme celles disponibles pour les vidéos du web ou des médias sociaux (e.g., texte autour des vidéos, titre, commentaires, tags) doit être compensée ;
7. le manque d'une image "requête" dans la plupart des situations (à part des portraits robot) est crucial au moins pour une première "itération" de la recherche.

Suite à ces observations, nos contributions du travail présenté portent sur les points suivants :

- comme une conséquence directe des observations 1, 5, 6, 7, la première contribution de ce travail est constituée par une étude des métadonnées utiles dans le contexte de la vidéosurveillance. Cette étude a abouti à la proposition d'un dictionnaire de métadonnées et d'une structuration de celui-ci dans un format basé sur la norme ISO 22311.
- les observations 2, 3, 4 ont mené à une focalisation de la recherche sur les métadonnées spatio-temporelles du dictionnaire défini par la première contribution. En conséquence, la recherche des extraits vidéos dans une collection issue des systèmes de vidéosurveillance est formulée comme un problème de modélisation des données spatio-temporelles. Plus précisément, les contributions proposées sont les suivantes (nous nous sommes focalisés sur l'environnement outdoor) :
  - la définition du concept de requête trajectoire hybride composée des positions exprimées par rapport à différents systèmes de référence (e.g., système géodésique, réseau routier, réseau de transport) ;
  - un modèle de données multicouches qui intègre des données décrivant d'une part "l'environnement" : réseau routier, réseau de transport, et d'autre part le mouvement des objets et le changement des champs de vue des caméras ;

- des opérateurs qui permettent la sélection automatique d'un ensemble de caméras susceptibles d'avoir filmé une scène recherchée et l'identification des séquences (intervalles de temps) vidéos correspondant à chaque caméra.

Nos contributions ont été validées dans le cadre d'un prototype mettant en oeuvre les contributions présentées. Il est basé sur l'API Google Maps pour construire des requêtes trajectoires hybrides et utilise des data sets fournis dans le cadre de l'open data par différentes collectivités (Transport Collectif de Toulouse).

## 6.2 Perspectives de recherche

Les contributions proposées et les expérimentations effectuées ont donné lieu à de nombreuses perspectives à court, moyen et long terme. Ces perspectives portent sur les points suivants :

1. Pour l'instant, la notion de trajectoire hybride ne concerne que l'hétérogénéité des positions exprimées par rapport à différents systèmes de référence en environnement outdoor. Le but final est de pouvoir définir de façon homogène une trajectoire qui commence en environnement outdoor (e.g., sur la Route de Narbonne à Toulouse) et qui se finit en milieu indoor (e.g., sur les quais du métro Paul Sabatier, direction Borderouge). Un premier pas a été fait dans cette direction par l'intermédiaire d'un travail réalisé en collaboration avec le NII de Tokyo sur la définition d'une trajectoire hybride en milieu indoor (détaillé en Annexe A).
2. Une des hypothèses de notre travail est que la variété et le grand volume des contenus vidéos rendent impossible l'analyse exhaustive de ceux-ci. En conséquence, il faut s'appuyer sur des métadonnées pertinentes dans le contexte de la vidéosurveillance pour réduire l'espace et implicitement le temps de recherche. Dans notre travail, nous nous appuyons sur des métadonnées spatio-temporelles afin de filtrer les vidéos qui ne concernent pas la trajectoire spatio-temporelle d'intérêt pour l'enquête. D'autres mesures de filtrage "négatif" peuvent être développées en se basant sur les métadonnées ou sur les caractéristiques de l'image. Par exemple, dans le cas d'utilisation présenté dans la section 5.3 du Chapitre 5, nous pouvons observer que les images illustrées dans la Figure 5.13 sont trop floues et ne sont pas utiles. En connaissant la vitesse de la caméra et en établissant par des expérimentations un seuil à partir duquel les images ne sont pas utilisables, alors ces images pourraient être

filtrées automatiquement sans les analyser. D'autres types de mesures de filtrage "négatif" des vidéos peuvent être définies en combinant les métadonnées et les paramètres de qualité d'image par exemple.

3. Le cadre d'expérimentation d'un framework comme celui que nous proposons est très complexe à cause du manque des collections de données et des protocoles généralisés. Dans ce sens, les principaux acteurs du domaine de la vidéosurveillance (opérationnels, autorités nationales, commerciaux, chercheurs) ont démarré une démarche de collaboration qui a comme objectif final la mise en place de plusieurs démonstrateurs dans des conditions réelles spécifiques au contexte de la vidéosurveillance afin de pouvoir tester des propositions d'algorithmes et de frameworks. Notre équipe fait partie des partenaires de cette démarche et plusieurs propositions de projets sous-jacents ont été déposées dans ce sens.
4. Le cadre applicatif de notre travail a été donné par les projets de recherche que nous avons menés dans le domaine de la vidéosurveillance. Toutefois, l'approche de modélisation multicouche que nous avons utilisée est assez générique et suppose une indépendance des différentes couches qui fait qu'elle pourrait être utilisée dans d'autres contextes d'application. Dans un environnement outdoor, notre framework pourrait être utilisé dans des contextes d'applications comme : des systèmes de transport intelligents<sup>1</sup>, des systèmes de suivi des trajectoires de animaux [Wannous, 2014]. Dans des environnement indoor, notre framework de modélisation des trajectoires hybrides des objets mobiles pourrait être utilisé dans le contexte de restaurants intelligents qui visent à suivre les comportements nutritionnels des personnes afin de pouvoir leur faire des recommandations. Les trajectoires peuvent être utilisées dans le sens où si une personnes passe trop de fois pendant un seul repas au rayon des sauces ou des desserts, cela pourrait être le signe d'une alimentation peu équilibrée.
5. Pour l'instant notre travail concerne les trajectoires hybrides des objets et des caméras. Une perspective très intéressante est d'utiliser les statistiques sur les informations passées afin de pouvoir faire de prédictions sur les trajectoires futures des objets. Par exemple, la notion de température définie dans [Manzat, 2013] afin de définir le niveau de popularité d'un contenu multimédia en fonction de nombre de fois qu'il est "utilisé" (e.g., cliqué, ouvert en temps que réponse pertinente à une requête) peut être étendue aux segments de rues sur lesquels des incidents se sont passés. Dans ce cas, la température traduirait une mesure d'insécurité d'un certain segment de rue. Cette mesure pourrait

---

1. <http://imsc.usc.edu/intelligent-transportation.html>

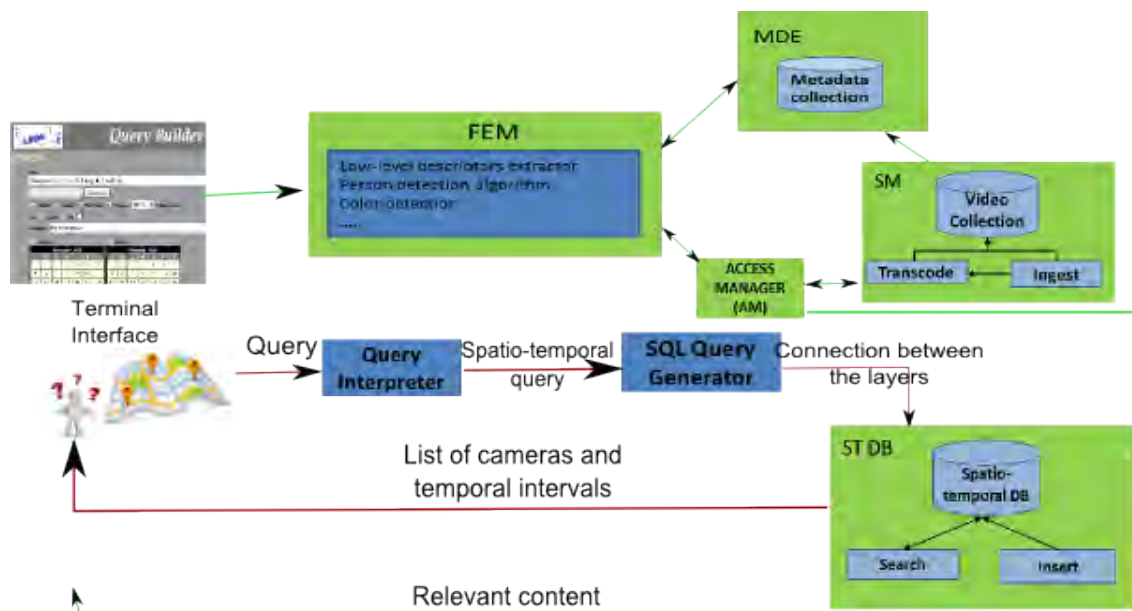


FIGURE 6.1 – Exemple d’architecture d’outil qui intègre le filtrage spatio-temporel et l’analyse vidéo

évoluer dans le temps et se ”propager” sur les segments proches.

6. Dans ce manuscrit, nous proposons de nous baser sur les métadonnées pour un filtrage du contenu et une exécution ”intelligente” des algorithmes d’indexation sur des sous-parties de la collection vidéo. Notre approche est complémentaire des approches d’analyse du contenu vidéo et ensemble peuvent constituer une solution puissante de recherche et d’indexation des contenus vidéos qui pourra être la base d’un outil Forensic tant recherché. La Figure 6.1 montre le schéma d’un tel outil (travail effectué en collaboration avec une équipe de NJIT de Newark) qui intègre les modules de filtrage spatio-temporel et les modules d’exécution des algorithmes d’analyse du contenu (FEM (Features Extractor Manager)). La nomenclature des modules d’analyse du contenu est celle utilisée sur le projet LINDO et est expliquée en détail dans [Brut et al., 2011a].



# Annexe A

## Indoor Location Model

### A.1 Introduction

Sensor networks are being widely deployed for a wide range of applications : location, navigation, measurement, detection, surveillance etc. In these applications users issue different types of spatio-temporal queries (e.g., position queries, region queries, trajectory queries) over a sensor database in order to extract the sensors or the objects situated on a certain path or in a certain region, the distance between certain objects or sensors, the intersections between objects' paths, the aggregation of certain measurements for a certain region or path (e.g., the energy consumption in a room or on a given path).

The sensors used in this wide range of applications are diverse : location sensors that generate tracking information (e.g., RFID, Wifi), sensors that detect orientation or inclination (e.g., compass, accelerometer), sensors that generate different types of measurements (e.g., temperature, illumination, energy consumption). The first type of sensor information is stored and managed within spatial and location models. These models can have other sensor information associated to them (e.g., we can deploy sensors that measure energy consumption for every energy consumer object in a building so that we could compute the energy consumption in a room or on the path of a person<sup>1</sup>) resulting in databases that contain spatial, temporal and sensor information [Bonnet et al., 2001], [Lee and Chung, 2008].

In an outdoor environment, these applications mainly use space and location models

---

1. <http://www.gizmag.com/meterplug-app-save-electricity/25638/>

based on GPS sensors, that are widely embedded in cars' on-board systems, smartphones, video cameras, etc. The indoor environments are becoming bigger and more complex and people are spending large parts of their lives inside these spaces (e.g., working buildings, subway infrastructure, airports).

The development of many types of indoor sensors (RFID [Want, 2006], Bluetooth [Feldmann et al., 2003], Wifi [Scuturici and Ejigu, 2006], video cameras [Aravecchia et al., 2010], IR sensors [Want et al., 1992], etc.) led to the creation of large volumes of tracking data that enable a range of services similar to those enabled by GPS-based positioning in outdoor settings. The main problem in the indoor context is the heterogeneity of the position information given by the different sensors. This heterogeneity concerns different cases : (1) the positions are either geometric or symbolic (while the geometric coordinates given by a Wifi based location system are 2D coordinates, symbolic coordinates are available via cell-IDs in cellular networks, as well as via other positioning technologies, such as radiofrequency tags (RFIDs) or infrared (IR) beacons), (2) the positions are expressed with regards to different coordinate systems, (3) there is a need to combine data coming from different sensors by taking into account the uncertainty. However, the multitude and variety of positioning systems and its discussion is beyond the scope of this work. A survey of indoor positioning systems is presented in [Sana, 2013]. Also, we are not focusing on the third point concerning different sensor data fusion. More information can be found in [Hightower et al., 2002] who present a multi-layer software engineering model for developing a common location framework that computes the fusion of different location types coming from different sensors.

Our work mainly addresses the first type of heterogeneity and consists of developing of a generic solution for objects trajectories modeling and querying framework. In the following section we give formal definitions related to the notion of object trajectory and we argue that path (or trajectory) oriented retrieval queries are not enough exploited in the existing query frameworks. Afterwards we present our conceptual modeling and operators that provide applications with direct support for heterogeneous trajectories.

## A.2 Trajectory oriented query

The idea behind the concept of object trajectory originates in the need to capture the movement of an object in an area for a period of time. The *movement track* of an object (or raw movement, or raw trajectory) is defined basically as a sequence of



spatio-temporal positions (i.e., positions that refer to the modification of an object’s spatial characteristics (coordinates and/or geometry) in time) [Parent et al., 2013]. Depending on the sensors that detected them, the positions can be either geometric (sequence of (x,y,t) triplets for 2D positions) [Parent et al., 2013] or symbolic (sequence of (rfidtag, t)) [Jensen et al., 2009]. Also depending on the object’s type and of the capability of the sensors associated to the object or embedded in the environment, additional data can be associated to the object’s movement (e.g., for a mobile camera, it is interesting to capture information like orientation and field of view) [Arslan Ay et al., 2008].

Based on this raw data, [Parent et al., 2013] define a *trajectory* as being composed of the the object’s movement track parts that are of interest for a given application in a given time interval delimited by  $t_{start}$  and  $t_{end}$ . These sequences of positions can be segmented based on several criteria (e.g., if a trajectory of a person is composed of many positions, a part of them have been crossed by foot and another part in a bus, we can say that his trajectory is composed of two segments).

Let us formalize this definition. A *trajectory* is defined as a tuple composed of a trajectory identifier, an object identifier, a set of trajectory segments (or units) and a global time interval that marks the beginning and the end of the trajectory.

$$Tr = (tr_{id}, obj_{id}, \{u_k\}, [t_{start}, t_{end}]) \quad (A.1)$$

$$U_k = (u_{id}, \{position_i\} / 1 \leq i, [t_{kstart}, t_{kend}]) \quad (A.2)$$

Many efforts were done regarding the definition of a *trajectory segment* (also called trajectory unit or episode). [Mountain and Raper, 2001] define a trajectory episode as a maximal subsequence of a trajectory such that all its spatio-temporal positions comply with a given predicate. The predicate may concern spatio-temporal aspects (the direction is the same) or other contextual aspects (e.g., the transportation mean is the same). The formalized definition is illustrated in A.2. A trajectory segment  $k$  is a tuple composed of a segment identifier, one or a set of homogeneous positions and an optional time interval for the segment.

There are objects that move free in the environment [Parent et al., 2006] like tornadoes, wild fires or white storks [Spaccapietra et al., 2008] and there are objects whose movement is constrained by road networks [Gutting et al., 2006], transportation networks [Booth et al., 2009], [Zheng et al., 2010] or the structure of the built

ding [Jensen et al., 2009]. Therefore, trajectory segments may be defined with regards to the underlying space topology (with regards to the streets segments, to transportation infrastructure and to a building architecture [Cao and Wolfson, 2005], [Jensen et al., 2009]. In this case a trajectory can be defined as the evolution of an object's position in space during a time interval [Spaccapietra et al., 2008]. This definition is formalized by the equation A.3.

$$Tr : [t_{begin}, t_{end}] \rightarrow space \quad (A.3)$$

The utility of spatial queries concerning past object's movement and predictive queries has been proven by a large amount of both research and commercial work. There are lots of work concerning spatial queries like : **position queries** [Becker and Durr, 2005] : return the locations of mobile and static objects (geometric or symbolic location) and are represented according to either a geometric or a symbolic model of space, **range queries** ([Gu et al., 2009], [Arslan Ay et al., 2008], [Scuturici and Ejigu, 2006], [Yuan and Schneider, 2010]) : "what are the pharmacies situated in this area" or "what are the paintings situated within 50 meters of my position", "what are the printers closer to my position or situated in the same room", **k- nearest neighbor queries** : "what are the closest available k nearest friends" [Zhang et al., 2003], **navigation queries** [Afyouni et al., 2014] : discovering optimal paths (with regards to criteria like distance or time) to some points of interest like parks or buildings or rooms.

Most of the works dealing with object trajectories aim to use past object trajectories to detect movement patterns and trajectory clustering [Patino et al., 2011] and hence to predict future object positions. This is done using geometric trajectories [Hendawi and Mokbel, 2012], [Calderara et al., 2009] and semantic trajectories [Parent et al., 2013]. Less work consider path oriented retrieval query and from our knowledge there is no existing query framework that can process heterogeneous trajectory (containing both symbolic and geometric segments). In the following we will present two use cases that will show the utility of such query.

### A.2.1 Video Protection Query

A victim of an aggression inside a metro station files a complaint. He/She is asked to describe his/her movement and the main characteristics of the aggressor. Based on his/her description and on witnesses testimonials the policemen manage to



FIGURE A.1 – The plan of the first floor of a subway station

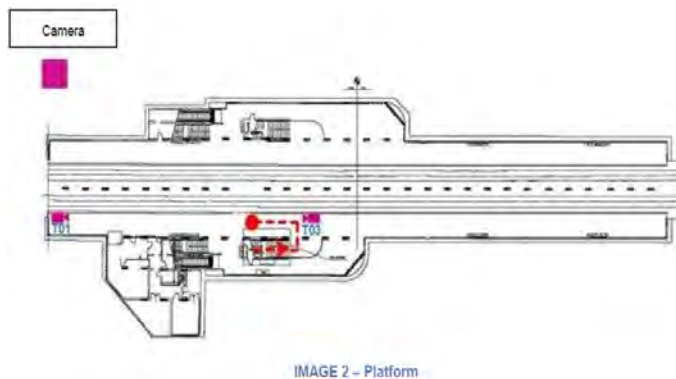


FIGURE A.2 – The plan of the platform of a subway station

reconstitute a possible relevant trajectory. The Figures B.1 and A.2 illustrate the person's trajectory inside the subway station that starts at the entrance in the station (B.1) and ends on the platform (A.2). In order to identify the aggressor, the videosurveillance system recordings will be analyzed. Based on the reconstituted trajectory the videosurveillance operators select the cameras whose fields of view have intersected the given trajectory in the given time interval.

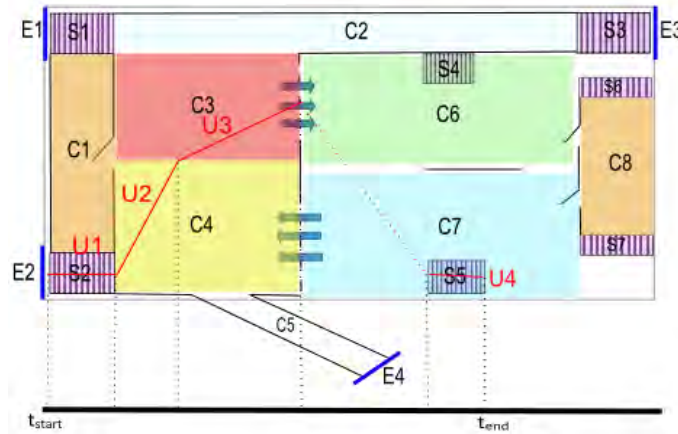


FIGURE A.3 – The plan of the first floor of a subway station

### A.2.2 Surveillance of a working area

Let's suppose an "intelligent" working building. Every employee has a smartphone that is tracked by positioning systems based on different localization sensors. Other types of sensors are monitoring the energy consumption and the temperature everywhere in the building. The figure A.3 illustrates the trajectory of a person inside the building. The trajectory is composed out of four segments ( $u_1$ ,  $u_2$ ,  $u_3$  and  $u_4$ ). There might be a lack of position information for a certain period of time (a person closed his/her smartphone and no information exist about it's position). The dotted line represents a gap in the definition of the trajectory (called "hole") that can be "filled" using interpolation functions [Vazirgiannis and Wolfson, 2001]. Based on this trajectory, we want to compute different operators like aggregation of the energy consumption, identifying the sensors situated on the path or at a certain distance of the path.

Lets consider the next case. There is an incident in a certain area of the building. A person is moving inside a prohibited are of the building. His/her movement track is generated based on the detection of his/her smartphone by the surrounding Wifi cells. This generates coordinates  $(x,y)$  with regards to the plan of the building. At some point the signal is lost and the only information that we have is that he entered a secured room (his badge was validated at the entrance of the room). In this case the person's trajectory would look like the one in the figure A.4. The first two segments of the trajectory are  $u_1$  and  $u_2$ , there is a period where the object was lost and the last part of the trajectory is  $u_3$  which is the room  $C_1$ . In that case we want to find

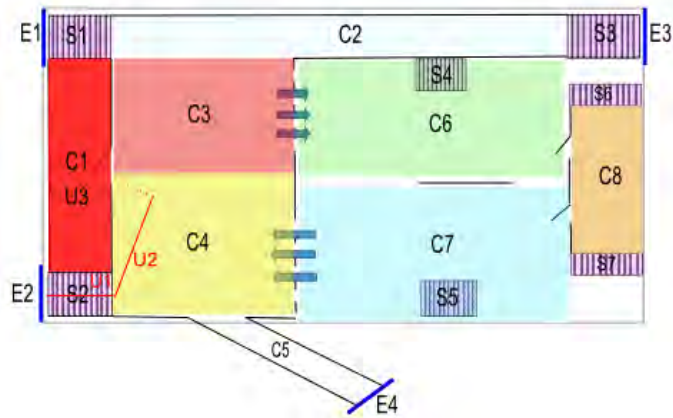


FIGURE A.4 – Example of trajectory

the video cameras on the given path in order to visualize the recordings.

In the following we are presenting the requirements that we identified for designing of a system that could support the queries identified in the two use cases.

### A.2.3 Requirements for a heterogeneous trajectory based query framework

The main problems when designing a object trajectory querying framework are : (1) the integration of multiple (geometric and symbolic) segments within a trajectory, (2) the integration of multiple coordinate systems (local coordinate system or global coordinate systems, indoor or outdoor), (3) the integration of different trajectories coming from different sensors (e.g., the same object has been tracked by different sensor, how to combine the two paths).

Based on this model, the requirements for the query framework that we developed followed the following strategy :

- Specification of the location model : even if the purpose of our work isn't to define yet another indoor location model since lots of efforts have been done in this direction, we will present a state of the art of the different types of existing approaches and we will describe and motivate our choices, by emphasizing the particularities of our model
- Specification of the query language (definition of a query template)

- Query transformation to SQL (definition of a query interpreter module)
- Data and query management tools (for a single coordinate system, multiple modelings)
- Aggregation operators within or after the query
- Geocoding and Reverse geocoding within a coordinate system

The identified requirements for the underlying location model are :

- *Geometric and symbolic location* As reflected by the given use cases, it is important for a location model to support both geometric and symbolic positions because : (1) sensors outputs are heterogeneous, (2) users employ more semantic location descriptions.
- *Connectivity relation* The representations of trajectories requires to use the connectivity information between locations. For that purpose it is crucial to use the *topological properties* of the environment (the plan of the building) because the object's movement is constrained by that topology. That information needs to be modeled explicitly in a location model.
- In order to integrate symbolic and geometrical coordinates, we need to define *mappings* between the spatial model (the representation of the environment) and the geometric coordinates. For that purpose, the geometry information related to every section (room in our case) of the spatial model needs to be integrated in the data model.
- *Distance* : Since many queries imply detecting the objects that are situated in the visibility range of a sensor, it is important that the information stored in the location model enables distance computation.
- *Coordinate system transformation* : When we are dealing with positioning technologies that express locations with regards to different coordinate systems, it is important to describe transformations between these reference systems.

In the following we are going to present a state of the art of the existing location models and their suitability to support trajectory queries by comparing them with regards to the identified requirements.

### **A.3 State of the art : Trajectories in indoor location models**

Spatial models imply the definition of a topology model that traduces how the space is represented and afterwards, the definition of object location and trajectories with

regards to the space model. This representation is called location model. We are talking about two types of location models : geometric location models and symbolic location models [Leonhardt, 1998], [Afyouni et al., 2012]. The indoor space is partitioned either using geometrical methods that generate a space division in uniform or non-uniform cells [Ledoux and Gold, 2008], [Mekni, 2010] or using symbolic elements, most of the time related to the topological elements of the building (rooms, hallways, doors). In our case, we are going to consider only the approaches that use building structure elements for space partitioning since the trajectories are mainly influenced by this structure. Based on this space partitioning, geometric and symbolic location models are defined.

Within *geometric models*, local and/or global reference systems are associated to space modules and objects locations are expressed with regards to these coordinates systems. The main advantage of these kind of models is the location high accuracy (ignoring the sensor measurements errors) and the straight forward computation of distances. The most important drawbacks are related to the non representation of topological relations (e.g., connections between different rooms) and the fact that this representation is non intuitive for users.

These drawbacks are addressed by the *symbolic location modeling* that is based on much more semantically descriptions about moving objects based on structural entities and/or points of interest (e.g., room/floor identifier, building name). The problems with this type of models are : the accuracy depends on the level of abstraction of the space model, the model is less suitable for distance computation, the choice of named locations dependent to the application (lack of interoperability), the modeling effort is sometimes considerable (the symbols for locations often need to be constructed and managed manually). Some works have also combined advantages from both geometrical and symbolic models and proposed *hybrid location models* [Buschka, 2005], [Leonhardt, 1998].

The existing surveys concerning location models are focused on the different approaches capacity of answering position, range and nearest neighbor queries [Becker and Durr, 2005], [Afyouni et al., 2012]. In the following we are going to present a state of the art of the geometric, symbolic and hybrid existing models by comparing them with regards to criteria relevant for our work : how to these models support different types of queries (trajectory or related to trajectory queries, topological (inside, overlaps)) ?

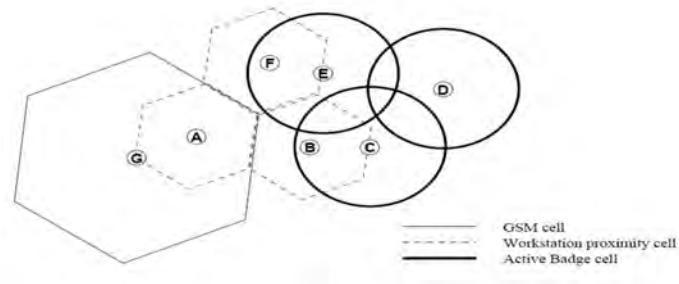


FIGURE A.5 – Cell-based locations

### A.3.1 Geometric location models

Within geometric location models, the space (or partitions of the space) is seen as an Euclidean space so entities locations as a 2D  $(X,Y)$  or 3D  $(X,Y,Z)$  set of coordinates with regards to global or local reference systems.

### A.3.2 Symbolic location models

Many research works addressed the problems of lack of topological relations and of semantics of the geometrical location representations. As can be seen from our examples, in many cases the trajectories given by users are semantic and are expressed with regards to the building topology.

The following location models represent locations with regards to different spatial partitions that can be modeled either by using different sensors and their range of action (Cell based and Zone based) or using building structure elements that can be modeled either as sets, either using graphs either by using hierarchies.

- *Cell based models* : different cells are defined with regards to different sensor's range of action (e.g., Wifi, RFID, ICCARD) and object's locations (symbolic) are expressed using cell ids (see Figure A.5 [Leonhardt, 1998]).

An example of cell-based location is presented by [Wang et al., 2012]. Their purpose is to artificially generate indoor objects trajectory (sequence of (rfidCell, timeStart, timeEnd)) by using a simulated building plan with a rfid sensors network deployed. Some supplementary geometrical information like distance between different sensors positions, sensors ranges and objects speed are used in order to compute probabilities for one object to move from one location to another. The topology properties of the building are not explicitly used.



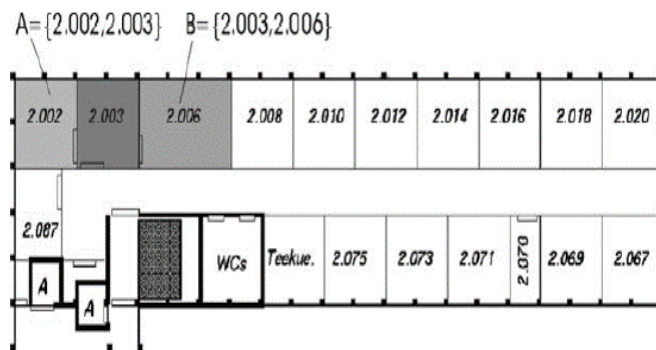


FIGURE A.6 – Set based model of a building floor

[Lee and Chung, 2008],

- *Set based* approaches model the indoor space and the indoor locations by creating sets and subsets of place identifiers based on the architectural properties of the environment (see Figure A.6 [Becker and Durr, 2005]). For example, the second floor of a building  $L_{Floor2}$  is represented as the set of all room numbers on that floor. This information can be organized hierarchically in order to represent the fact that a room is part of a floor and a floor is part of building that has two wings [Durr and Rothermel, 2003]. The resulting structure is a lattice (see Figure A.7). Set-based operations are used to evaluate overlapping or inclusion relations between two locations. No direct connectivity between locations can be represented and the distance than can be computed is only qualitative.

In order to cope with the lack of connectivity information, [Li and Lee, 2008] propose also a lattice-based approach but the symbolic identifiers are associated to the two types of concepts describing indoor space, location (a bounded space such a room) and exit (points that enable entering and leaving locations). A concept lattice  $(C(L,E,I), \leq)$  is build where  $L$  is the set of locations,  $E$  is the set of exits,  $I$  is the binary relation between locations and exits and  $\leq$  is the containment relation (see Figure A.8).

- *Graph based* approaches model symbolic locations using a graph structure [Hu and Lee, 2004] , [Jensen et al., 2009]. The vertices of the graph are represented by places in the indoor environment (e.g., rooms, staircases). The edges of the graph are the connections between the different places.
- *Object based models*

[Gonzalez et al., 2006] [Satoh, 2005]

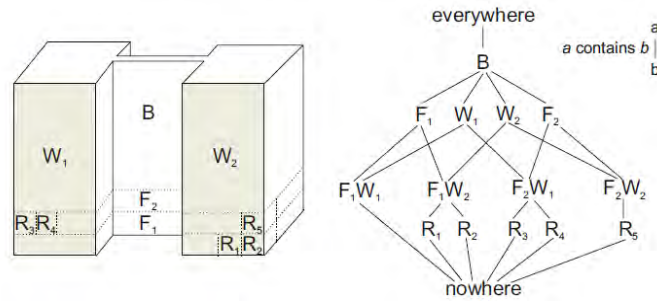


FIGURE A.7 – Lattice structure

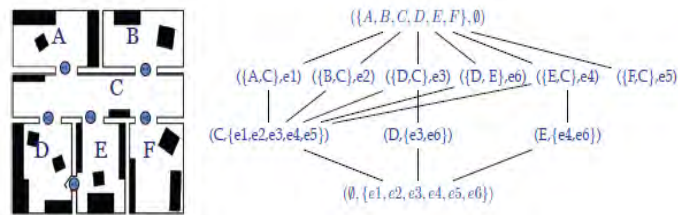


FIGURE A.8 – Lattice structure

### A.3.3 Hybrid models

Hybrid models address the limits of geometric or symbolic models by offering solutions that combine the two localization models [Leonhardt, 1998], [Afyouni et al., 2014], [Jiang and Steenkiste, 2002].

## A.4 Query template definition

According to the steps of our strategy, we started by defining the template of the query that our system will support based on the two motivation examples that we presented. Let us first give the definition of what we named hybrid trajectory based query. As explained before, an object’s trajectory is a sequence of trajectory segments and a time interval. Each segment is composed of some positions and an optional time interval for the segment. The positions are homogeneous within a segment (e.g., are expressed with regards to the same coordinate system and are either geometric or symbolic) but different segments can contain heterogeneous positions like the two examples show. So, a hybrid trajectory is one than contains also symbolic and

```
{
  "query": {
    "spatial": [
      {
        "WGS84": [
          [43.56077,1.46298],
          [43.56088,1.46292],
          [43.56366,1.46156]
        ]
      },
      {
        "Street Montesquieu": [14,19]
      },
      {
        "Bus2": ["Université Paul Sabatier","Faculté de Pharmacie"]
      }
    ],
    "temporal": {
      "start": "07-02-2014 14:00:00",
      "end": "07-02-2014 14:30:00"
    }
  }
}
```

FIGURE A.9 – Hybrid trajectory based query example

geometric trajectory segments. This hybrid trajectory will constitute the entry point of our query framework.

In our previous work, we have defined a similar hybrid trajectory query for an outdoor urban environment. In a such environment objects positions may be expressed mainly either using GPS coordinates, either with regards to the road network (e.g., 14 Street Montesquieu, Toulouse, France), either with regards to the transportation network (e.g., bus line number 2, station 3). We are going to illustrate our query schema (or template) by leaning on the JSON standard<sup>2</sup> that represents one of the most employed solutions for geographical (and not only geographical) data exchange format. According to the trajectory definition that we gave, a trajectory query has a spatial and a temporal part. The spatial part is a sequence of segments, and the temporal part has a *start* and an *end* attribute. Every trajectory segment has a reference system part and an array of positions with regards to that reference system.

Since the ultimate goal of our research is to define a seamless indoor/outdoor, symbolic/geometric and local/global reference system trajectory model, we followed the same the format in order to define hybrid indoor trajectory queries. The Figure A.10 illustrates a trajectory that contains two segments, one composed of geometrical positions expressed with regards to the floor plan and one composed of one symbolic position which represents the ID of the sensor (ICCARD reader) which is expressed

---

2. <http://json.org/>

```

{
  "query": {
    "spatial": [
      {
        "Floor8": [
          [50.21627, 433.63088],
          [65.84485, 409.75342],
          [60.56707, 441.10219],
          [93, 67555, 371, 18614],
          [97, 67467, 389, 67915]
        ]
      },
      {
        "ICCARD": [0000110015010005]
      }
    ],
    "temporal": {
      "start": "10-02-14 12:55:34",
      "end": "10-02-2014 13:20:00"
    }
  }
}

```

FIGURE A.10 – Hybrid indoor trajectory based query example

with regards to the ICCARD readers network deployed in the building.

The illustration of the query is presented in the Figure A.11. The green dots connected by the red line represents the first segment. The blue dot represents the position of the ICCARD reader that corresponds to the second segment.

## A.5 Geometrical-Symbolic information mapping

In order to be able to manage heterogeneous trajectory segments, the geometrical information related to the symbolic structural parts of the building plan needs to be captured. The mapping function *RoomsToGeometrical* is defined as :

$$RoomsToGeometrical : Rooms \rightarrow Polygon \quad (A.4)$$

For now we have done the mapping only for the rooms whose name appears in the Figure A.12 but doing it for the rest of the rooms is straightforward since it can be done using the graphical interface by clicking on the boundary points of the room and then by pressing draw (see Figure A.12). The polygon is saved in an Oracle Spatial database together with a RoomID.



FIGURE A.11 – Indoor trajectory visualization



FIGURE A.12 – Snapshot of the indoor plan



FIGURE A.13 – Snapshot of the indoor plan

The doors and connections between the different rooms of the plan have also an associated geometry (a line) that can be defined using the graphical interface similar to the room geometries.

$$\text{DoorsToGeometrical} : \text{Door} \rightarrow \text{Line} \quad (\text{A.5})$$

## A.6 Representing topology

As we can see from the Figure A.11, the spatial trajectory resulted by connecting the points detected by the positioning system gives incorrect results (lines crossing walls). One solution would be to express trajectory segments using a high density of points (for example for every second). Another solution is to use the topological properties of the building floor in order to map the objects movement on this topology. In order to represent the topological properties of a building we are going to use the graph based approach [Jensen et al., 2009]. Each piece of the building (room, hallway, staircase) represent vertices of the graph and the connections between them (doors, passages) represent edges. We took a small part of our floor plan in order to illustrate the topology graph. The graph in the figure A.15 represents the topology graph corresponding to a small part of the floor plan illustrated by the figure A.14.

Every vertex and edge in the topology graph has associated a geometry attribute. Based on this graph we are going to define a mapping function that will take as



FIGURE A.14 – Indoor floor building plan

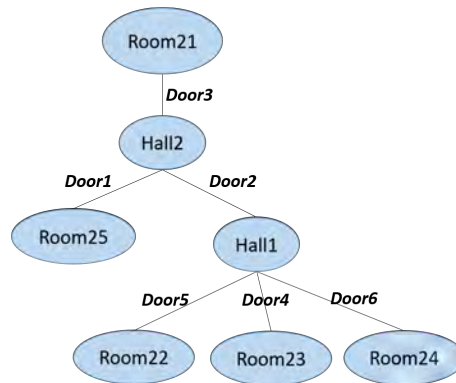


FIGURE A.15 – Example of topology graph

input a geometrical trajectory segment and will map it to the building topology by enriching the sequence of points forming the trajectory. Let us explain via an example. Given the first segment of the trajectory illustrated in A.10, the figure A.16 illustrates the trajectory without considering the topology graph. After applying the mapping function whose algorithm is illustrated by algorithm 4, the resulted trajectory is illustrated in the figure A.17.

---

**Algorithm 4:** The algorithm for mapping trajectory to the floor graph

---

**Input:** A sequence of points :  $points_i$  , and a topology graph with the associated geometries  $G < Rooms, Doors >, RoomsToGeometrical, DoorsToGeometrical$

**Output:** A sequence of points :  $points_j$

```
1 finalList ← initialList // initialize the result list with the initial list of trajectory points
2 for each pair (pointi, pointi+1) do
3   roomStart ← (geomToSymbolic(pointi); roomStop ← (geomToSymbolic(pointi+1);
4   if roomStart ≠ roomStop then
5     intermediatePath ← computeShortestPath(roomStart, roomStop, G);
6     //computing the shortest path between the two rooms in the topology graph, the list
       intermediatePath will contain also startRoom and endRoom
7   end
8   for each  $r_k$  in intermediatePath do
9     connection ← getConnection( $r_k$ ,  $r_{k+1}$ )
10    //extract the connections(doors or other types of connections) between the rooms on
       the path between each pair of points in the trajectory
       insert(finalList, middlePoint(connection)) //insert at the good position (between
       pointi and pointi+1) the point representing the centroid of the geometry associated
       to connection between the rooms
11  end
12 end
```

---

## A.7 Indoor coordinate system

Expressing localization information with regards to different local reference system is one of the main causes for heterogeneity and lack of interoperability among location models. In outdoor environment, the existence of the GPS reference system makes things much more easier. In general, spatial models have a multi layer structure with associated mapping functions that can be used to project any local coordinates with regards to the GPS reference system. A challenging local coordinate system example





FIGURE A.16 – Example of trajectory before the mapping



FIGURE A.17 – Example of trajectory after the mapping

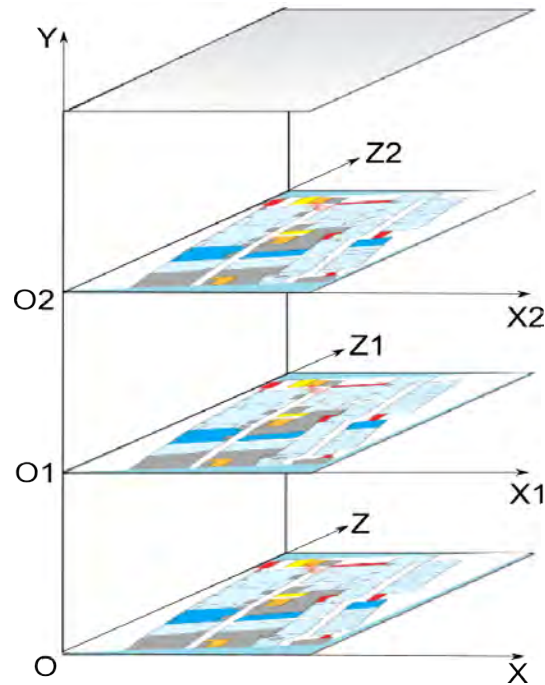


FIGURE A.18 – Coordinates system of a building

in outdoor environment can be one associated to moving objects (such as trains or buses), where local reference systems can help to address objects, such as travelers with respect to their compartment in the train and not their absolute position to the ground.

In indoor environments, there isn't a global reference system similar to GPS. Therefore, local coordinate systems might be associated to the whole building (for 3D positions), to one floor or to each room. Given the plan of the building we are going to consider one coordinate system per floor like in the Figure A.18. Positions are expressed using 2D coordinates with regards to plan of the floor. A third coordinate is associated to each position to indicate the floor number.

Therefore, in order to satisfy the generality and query requirements identified in the introduction, a location service needs to accommodate both symbolic and geometric location information. In particular, it must be able to process query segment in either representation, which implies performing reference systems transformations and geometric to symbolic mappings that the user isn't aware of.

# Annexe B

## GTFS (General Transit Feed Specification) format

The General Transit Feed Specification (GTFS)<sup>1</sup> defines a common format for public transportation schedules and associated geographic information. The format was developed in 2005, when Trimet in Portland, Oregon began working with Google on incorporating transit agency data in their trip planners.[1] They came up with Google Transit Feed Specification, which was easily maintainable and could be imported into Google Maps. Google offered their trip planning services for free to any agency that formatted and maintained their transit data in that format, later to become called General Transit Feed Specification. Now, GTFS has become the most popularly-used data format in the world, with increasing numbers of agencies choosing to share their transit data with the public.

GTFS "feeds" allow public transit agencies to publish their transit data and developers to use that data to write applications. The feeds are represented in a series of text files that are compressed into a ZIP file, and include information such as fixed-route schedules, routes, and bus stop data. Many transit agencies have created and published GTFS data with the primary purpose being integration with Google Maps. However, GTFS data can be used by a variety of third-party software applications for many purposes, such as trip planning, ridesharing, and mobile applications.

---

1. <https://developers.google.com/transit/gtfs/reference?hl=fr-FR>

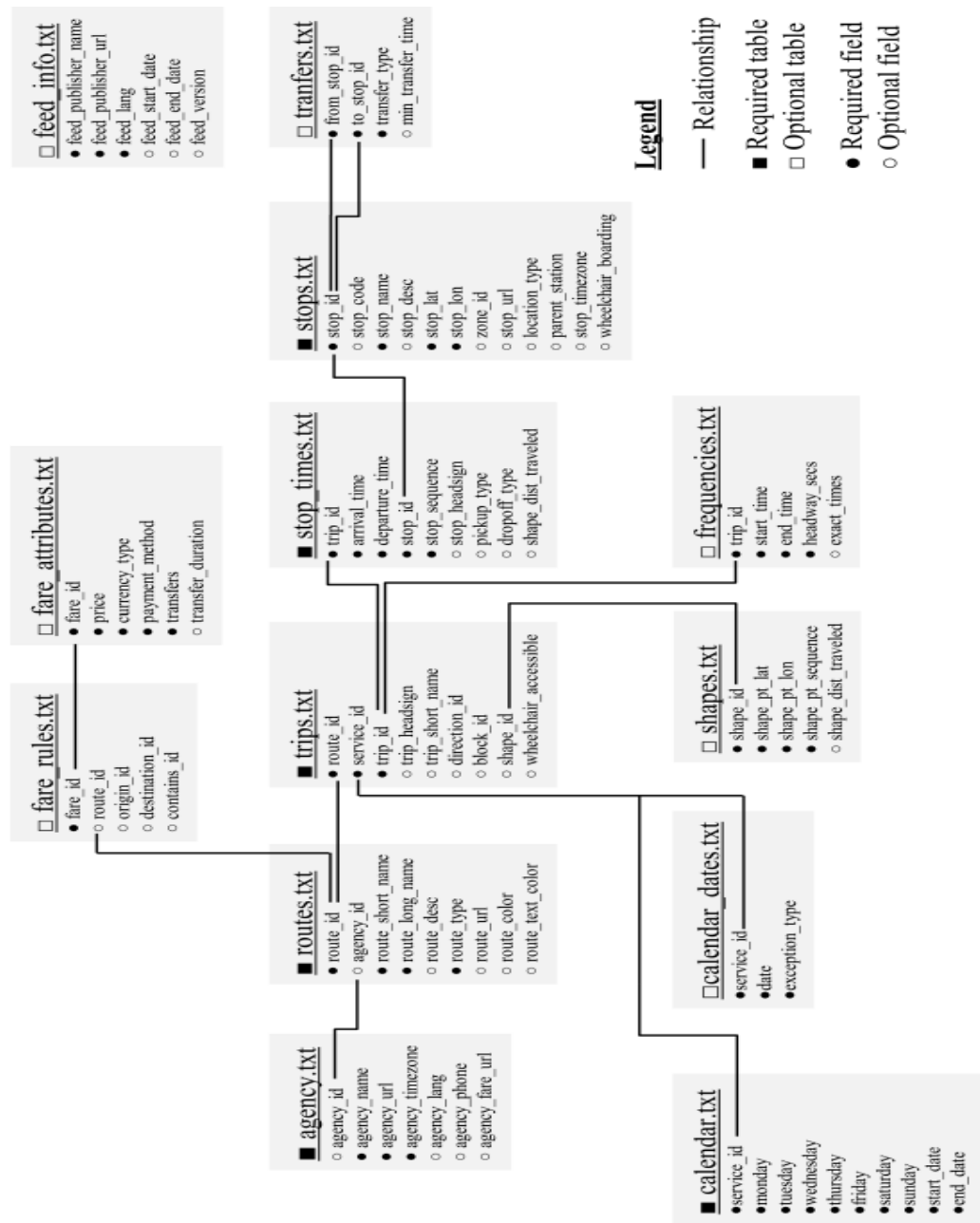


FIGURE B.1 – Le schema UML du format GTFD

# Bibliographie

- [Afyouni et al., 2012] Afyouni, I., Ray, C., and Claramunt, C. (2012). Spatial models for context-aware indoor navigation systems : A survey. *J. Spatial Information Science*, 4(1) :85–123.
- [Afyouni et al., 2014] Afyouni, I., Ray, C., Ilarri, S., and Claramunt, C. (2014). A postgresql extension for continuous path and range queries in indoor mobile environments. volume 15, pages 128 – 150. Special Issue on Information Management in Mobile Applications Special Issue on Data Mining in Pervasive Environments.
- [AL-Khalidi et al., 2013] AL-Khalidi, H., Taniar, D., and Safar, M. (2013). Approximate algorithms for static and continuous range queries in mobile navigation. *Computing*, 95(10-11) :949–976.
- [Amriki and Atrey, 2014] Amriki, K. and Atrey, P. (2014). Bus surveillance : how many and where cameras should be placed. *Multimedia Tools and Applications*, 71(3) :1051–1085.
- [AN2V, 2014] AN2V (2014). Livre blanc technico-juridique sur les évolutions de l’arrêté de 2007 concernant la vidéoprotection. Technical report, AN2V-Association nationale de la vidéoprotection.
- [Aravecchia et al., 2010] Aravecchia, M., Calderara, S., Chiossi, S., and Cucchiara, R. (2010). A videosurveillance data browsing software architecture for forensics : From trajectories similarities to video fragments. In *Proceedings of the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence*, MiFor’10, pages 37–42, New York, NY, USA. ACM.
- [Arslan Ay et al., 2010a] Arslan Ay, S., Kim, S. H., and Zimmermann, R. (2010a). Generating synthetic meta-data for georeferenced video management. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS ’10, pages 280–289, New York, NY, USA. ACM.
- [Arslan Ay et al., 2010b] Arslan Ay, S., Zimmermann, R., and Kim, S. (2010b). Relevance ranking in georeferenced video search. *Multimedia Systems*, 16(2) :105–125.

- [Arslan Ay et al., 2008] Arslan Ay, S., Zimmermann, R., and Kim, S. H. (2008). Viewable scene modeling for geospatial video search. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 309–318, New York, NY, USA. ACM.
- [Beal, 2003] Beal, J. J. (2003). Contextual geolocation : A specialized application for improving indoor location awareness in wireless local area networks. In *MICS2003 : The 36th Annual Midwest Instruction and Computing Symposium*.
- [Becker and Durr, 2005] Becker, C. and Durr, F. (2005). On location models for ubiquitous computing. *Personal Ubiquitous Comput.*, 9(1) :20–31.
- [Beelen, 2004] Beelen, M. (2004). Personal intelligent travel assistant.
- [Benetis et al., 2006] Benetis, R., Jensen, S., Karciuskas, G., and Saltenis, S. (2006). Nearest and reverse nearest neighbor queries for moving objects. *The VLDB Journal*, 15(3) :229–249.
- [Bohm and Rakow, 1994] Bohm, K. and Rakow, T. C. (1994). Metadata for multimedia documents. *SIGMOD Record*, 23 :21–26.
- [Bonnet et al., 2001] Bonnet, P., Gehrke, J., and Seshadri, P. (2001). Towards sensor database systems. In Tan, K.-L., Franklin, M., and Lui, J.-S., editors, *Mobile Data Management*, volume 1987 of *Lecture Notes in Computer Science*, pages 3–14. Springer Berlin Heidelberg.
- [Booth et al., 2009] Booth, J., Sistla, P., Wolfson, O., and Cruz, I. F. (2009). A data model for trip planning in multimodal transportation systems. In *Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology*, EDBT '09, pages 994–1005. ACM.
- [Borole et al., 2013] Borole, N., Rout, D., Goel, N., Vedagiri, P., and Mathew, T. V. (2013). Multimodal public transit trip planner with real-time transit data. *Procedia - Social and Behavioral Sciences*, 104(0) :775 – 784. 2nd Conference of Transportation Research Group of India (2nd CTRG).
- [Brakatsoulas et al., 2005] Brakatsoulas, S., Pfoser, D., Salas, R., and Wenk, C. (2005). On map-matching vehicle tracking data. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB'05, pages 853–864. VLDB Endowment.
- [Brinkhoff, 2002] Brinkhoff, T. (2002). A framework for generating network-based moving objects. *Geoinformatica*, 6(2) :153–180.
- [Brut et al., 2011a] Brut, M., Codreanu, D., Dumitrescu, S., Manzat, A.-M., and Sedes, F. (2011a). **A Distributed Architecture for Flexible Multimedia**

- Management and Retrieval. In *Proceedings of the 22nd International Conference on Database and Expert Systems Applications - Volume Part II*, DEXA'11, pages 249–263. Springer-Verlag.
- [Brut et al., 2011b] Brut, M., Codreanu, D., Manzat, A.-M., and Sèdes, F. (2011b). **Distributed multimedia indexing and optimal resources utilization : an implementation based on metadata, context and usage.** *The Journal of Multimedia Processing and Technologies, Intelligent Multimedia Processing*, 2(4) :197–225.
- [Brut et al., 2011c] Brut, M., Codreanu, D., Manzat, A.-M., and Sedes, F. (2011c). **Adapting Indexation to the Content, Context and Queries Characteristics in Distributed Multimedia Systems.** In *Proceedings of the 2011 Seventh International Conference on Signal Image Technology & Internet-Based Systems*, SITIS'11, pages 118–125, Washington, DC, USA. IEEE Computer Society.
- [Buschka, 2005] Buschka, P. (2005). *An Investigation of Hybrid Maps for Mobile Robots*. PhD thesis, Dept. of Technology, Orebro University, Sweden.
- [Calderara et al., 2009] Calderara, S., Prati, A., and Cucchiara, R. (2009). Video surveillance and multimedia forensics : An application to trajectory analysis. In *Proceedings of the First ACM Workshop on Multimedia in Forensics*, MiFor'09, pages 13–18, New York, NY, USA. ACM.
- [Cao and Wolfson, 2005] Cao, H. and Wolfson, O. (2005). Nonmaterialized motion information in transport networks. In *Proceedings of the 10th International Conference on Database Theory*, ICDT'05, pages 173–188, Berlin, Heidelberg. Springer-Verlag.
- [Cheung and Fu, 1998] Cheung, K. L. and Fu, A. W.-C. (1998). Enhanced nearest neighbour search on the r-tree. *SIGMOD Rec.*, 27(3) :16–21.
- [Chleq et al., 1999] Chleq, N., Bremond, F., and Monique, T. (1999). Image understanding for prevention of vandalism in metro stations. In Regazzoni, C., Fabri, G., and Vernazza, G., editors, *Advanced Video-Based Surveillance Systems*, volume 488 of *The Springer International Series in Engineering and Computer Science*, pages 106–116. Springer US.
- [Chu and Smeulders, 2010] Chu, D. and Smeulders, A. W. M. (2010). Thirteen hard cases in visual tracking. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 103–110.
- [Codreanu et al., 2011] Codreanu, D., Laborie, S., and Sedes, F. (2011). **Modelling Multimedia Indexing Algorithms Description for Implicit and Explicit**

- Indexation. In *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*, pages 232–238.
- [Codreanu et al., 2012] Codreanu, D., Manzat, A.-M., and Sedes, F. (2012). **Optimal resources utilization for indexing within a distributed multimedia retrieval system : an implementation for a video surveillance use case.** In *DMS*, pages 107–112. Knowledge Systems Institute.
- [Codreanu et al., 2013] Codreanu, D., Manzat, A.-M., and Sedes, F. (2013). **Mobile Objects and Sensors within a Video Surveillance System : Spatio-temporal Model and Queries.** In *IMMoA*, volume 1075 of *CEUR Workshop Proceedings*, pages 52–59. CEUR-WS.org.
- [Codreanu et al., 2014a] Codreanu, D., Péninou, A., and Sèdes, F. (2014a). **Modélisation de trajectoires cible/caméra : requêtes spatio-temporelles dans le cadre de la vidéosurveillance (poster).** In Reynaud, Chantal and Martin, Arnaud and Quiniou, René and, editor, *Journées Francophones Extraction et Gestion de Connaissances (EGC)*, volume RNTI-E-26, pages 573–576, <http://www.antsearch.univ-tours.fr/rnti/>. Revue des Nouvelles Technologies de l’Information (RNTI).
- [Codreanu et al., 2014b] Codreanu, D., Péninou, A., and Sèdes, F. (2014b). **Recherche d’extraits vidéos par reconstitution des trajectoires de caméras mobiles à partir d’un modèle spatio-temporel-Application à la vidéosurveillance.** In *INFormatique des Organisations et Systemes d’Information et de Decision (INFORSID), Lyon, France*, pages 11–26. Association INFORSID.
- [Compton et al., 2009] Compton, M., Henson, C., Lefort, L., Neuhaus, H., and Sheth, A. (2009). A survey of the semantic specification of sensors. In *In 2nd International Semantic Sensor Networks Workshop*.
- [Cucchiara, 2005] Cucchiara, R. (2005). Multimedia surveillance systems. In *Proceedings of the Third ACM International Workshop on Video Surveillance & Sensor Networks, VSSN’05*, pages 3–10, New York, NY, USA. ACM.
- [Cupillard et al., 2002] Cupillard, F., Bremond, F., and Thonnat, M. (2002). Tracking groups of people for video surveillance. In Remagnino, P., Jones, G., Paragios, N., and Regazzoni, C., editors, *Video-Based Surveillance Systems*, pages 89–100. Springer US.
- [De Sutter et al., 2006] De Sutter, R., Notebaert, S., and Van de Walle, R. (2006). Evaluation of metadata standards in the context of digital audio-visual libraries. In *Proceedings of the 10th European Conference on Research and Advanced Technology for Digital Libraries, ECDL’06*, pages 220–231. Springer-Verlag.



- [Debnath and Borcea, 2013] Debnath, H. and Borcea, C. (2013). Tagpix : Automatic real-time landscape photo tagging for smartphones. In *6th International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*. Mobilware 2013.
- [Denning, 2000] Denning, P. J. (2000). Computer science : The discipline.
- [Dijkstra, 1959] Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1) :269–271.
- [Dufour, 2012] Dufour, J.-Y. (2012). *Intelligent Video Surveillance Systems*. Wiley.
- [Durr and Rothermel, 2003] Durr, F. and Rothermel, K. (2003). On a location model for fine-grained geocast. In Dey, A., Schmidt, A., and McCarthy, J., editors, *UbiComp 2003 : Ubiquitous Computing*, volume 2864 of *Lecture Notes in Computer Science*, pages 18–35. Springer Berlin Heidelberg.
- [Erwig and Schneider, 2002a] Erwig, M. and Schneider, M. (2002a). Spatio-temporal predicates. *IEEE Trans. on Knowl. and Data Eng.*, 14(4) :881–901.
- [Erwig and Schneider, 2002b] Erwig, M. and Schneider, M. (2002b). Stql – a spatio-temporal query language.
- [Feldmann et al., 2003] Feldmann, S., Kyamakya, K., Zapater, A., and Lue, Z. (2003). An indoor bluetooth-based positioning system : concept, implementation and experimental evaluation. In *International Conference on Wireless Networks, ICWN 2003*, pages 109–113.
- [Feng et al., 2007] Feng, J., Wu, L., Zhu, Y., Mukai, N., and Watanabe, T. (2007). Continuous k-nearest neighbor search under mobile environment. In Dong, G., Lin, X., Wang, W., Yang, Y., and Yu, J., editors, *Advances in Data and Web Management*, volume 4505 of *Lecture Notes in Computer Science*, pages 566–573. Springer Berlin Heidelberg.
- [Forlizzi et al., 2000] Forlizzi, L., Guting, R. H., Nardelli, E., and Schneider, M. (2000). A data model and data structures for moving objects databases. In *SIGMOD conference*, pages 319–330.
- [Frentzos, 2003] Frentzos, E. (2003). Indexing objects moving on fixed networks. In Hadzilacos, T., Manolopoulos, Y., Roddick, J., and Theodoridis, Y., editors, *Advances in Spatial and Temporal Databases*, volume 2750 of *Lecture Notes in Computer Science*, pages 289–305. Springer Berlin Heidelberg.
- [Girgensohn et al., 2006] Girgensohn, A., Shipman, F., Dunnigan, A., Turner, T., and Wilcox, L. (2006). Support for effective use of multiple video streams in security. In *Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, VSSN’06*, pages 19–26, New York, NY, USA. ACM.

- [Gonzalez et al., 2006] Gonzalez, H., Han, J., Li, X., and Klabjan, D. (2006). Warehousing and analyzing massive rfid data sets. In *Data Engineering, 2006. ICDE '06. Proceedings of the 22nd International Conference on*, pages 83–83.
- [Goudou et al., 2013] Goudou, J.-F., Naud, L., Giulieri, L., Rabarisoa, J., Pietquin, O., Codreanu, D., and Petrovska, D. (2013). **L'évaluation d'algorithmes d'analyse vidéo** Quelques pistes. In *Workshop Interdisciplinaire sur la Sécurité Globale, WISG'13*.
- [Graham et al., 1965] Graham, C. H., Bartlett, N. R., Brown, J. L., Hsia, Y., Meuller, C. C., and Riggs, L. A. (1965). *Vision and Visual Perception*. John Wiley & Sons, Inc.
- [Gu et al., 2009] Gu, Y., Yu, G., Guo, N., and Chen, Y. (2009). Probabilistic moving range query over rfid spatio-temporal data streams. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM'09*, pages 1413–1416, New York, NY, USA. ACM.
- [Güting et al., 2000] Güting, R. H., Böhlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M. (2000). A foundation for representing and querying moving objects. *ACM Trans. Database Syst.*, 25(1) :1–42.
- [Guttting et al., 2006] Guttting, H., de Almeida, T., and Ding, Z. (2006). Modeling and querying moving objects in networks. *The VLDB Journal*, 15(2) :165–190.
- [Guttting and Schneider, 2005] Guttting, H. and Schneider, M. (2005). *Moving Objects Databases*. Morgan Kaufmann Publishers.
- [Hansen et al., 2009] Hansen, R., Wind, R., Jensen, C. S., and Thomsen, B. (2009). Seamless indoor/outdoor positioning handover for location-based services in streamspin. In *Mobile Data Management : Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, pages 267–272.
- [Hecht, 2001] Hecht, E., editor (2001). Addison-Wesley Publishing Company, 4 edition.
- [Hendawi and Mokbel, 2012] Hendawi, A. M. and Mokbel, M. F. (2012). Predictive spatio-temporal queries : A comprehensive survey and future directions. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, MobiGIS'12*, pages 97–104, New York, NY, USA. ACM.
- [Hightower et al., 2002] Hightower, J., Brumitt, B., and Borriello, G. (2002). The location stack : A layered model for location in ubiquitous computing. In *Proceedings*

- of the *Fourth IEEE Workshop on Mobile Computing Systems and Applications*, WMCSA'02, pages 22–30, Washington, DC, USA. IEEE Computer Society.
- [Houda et al., 2010] Houda, M., Khemaja, M., Oliveira, K., and Abed, M. (2010). A public transportation ontology to support user travel planning. In *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*, pages 127–136.
- [Hu and Lee, 2004] Hu, H. and Lee, D.-L. (2004). Semantic location modeling for location navigation in mobile environment. In *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, pages 52–61.
- [Hwang et al., 2003] Hwang, T.-H., Choi, K.-H., Joo, I.-H., and Lee, J.-H. (2003). Mpeg-7 metadata for video-based gis applications. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings. 2003 IEEE International*, volume 6, pages 3641–3643.
- [Ilarri et al., 2010] Ilarri, S., Mena, E., and Illarramendi, A. (2010). Location-dependent query processing : Where we are and where we are heading. *ACM Comput. Surv.*, 42(3) :12 :1–12 :73.
- [Jensen et al., 2003] Jensen, C. S., Kolářvr, J., Pedersen, T. B., and Timko, I. (2003). Nearest neighbor queries in road networks. In *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems*, GIS '03, pages 1–8, New York, NY, USA. ACM.
- [Jensen et al., 2009] Jensen, C. S., Lu, H., and Yang, B. (2009). Graph model based indoor tracking. In *Mobile Data Management : Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 122–131.
- [Jiang and Steenkiste, 2002] Jiang, C. and Steenkiste, P. (2002). A hybrid location model with a computable location identifier for ubiquitous computing. In *Proceedings of the 4th International Conference on Ubiquitous Computing*, UbiComp'02, pages 246–263, London, UK, UK. Springer-Verlag.
- [Keval, 2009] Keval, H. U. (2009). *Effective design, configuration, and use of digital CCTV*. PhD thesis, University College London.
- [Kim et al., 2010] Kim, S. H., Arslan Ay, S., and Zimmermann, R. (2010). Design and implementation of geo-tagged video search framework. *J. Vis. Comun. Image Represent.*, 21(8) :773–786.
- [Kolodziej and Hjelm, 2006] Kolodziej, K. and Hjelm, J. (2006). *Local positioning systems : LBS applications and services*. CRC Press.
- [Kraus, 2012] Kraus, M. (2012). *Active Multi-Camera Navigation in Video Surveillance Systems*. PhD thesis, Delft University of Technology.

- [Lassila, 1998] Lassila, O. (1998). Web metadata : a matter of semantics. *Internet Computing, IEEE*, 2(4) :30–37.
- [Ledoux and Gold, 2008] Ledoux, H. and Gold, C. M. (2008). Modelling three-dimensional geoscientific fields with the voronoi diagram and its dual. *Int. J. Geogr. Inf. Sci.*, 22(5) :547–574.
- [Lee and Chung, 2008] Lee, C.-H. and Chung, C.-W. (2008). Efficient storage scheme and query processing for supply chain management using rfid. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD’08*, pages 291–302, New York, NY, USA. ACM.
- [Lee et al., 2010] Lee, K., Lee, W.-C., and Leong, H. (2010). Nearest surround queries. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10) :1444–1458.
- [Leonhardt, 1998] Leonhardt, U. (1998). *Supporting Location-Awareness in Open Distributed Systems*. PhD thesis, Imperial College of Science, Technology and Medicine University of London.
- [Lew et al., 2006] Lew, M. S., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval : State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1) :1–19.
- [Li and Lee, 2008] Li, D. and Lee, D. (2008). A lattice-based semantic location model for indoor navigation. In *Mobile Data Management, 2008. MDM ’08. 9th International Conference on*, pages 17–24.
- [Liu et al., 2012] Liu, K., Li, Y., He, F., Xu, J., and Ding, Z. (2012). Effective map-matching on the most simplified road network. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 609–612.
- [Liu et al., 2009] Liu, X., Corner, M., and Shenoy, P. (2009). Seva : Sensor-enhanced video annotation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(3) :24 :1–24 :26.
- [Manzat, 2013] Manzat, A.-M. (2013). *Contribution à la Modélisation des Métadonnées Associées aux Documents Multimédias et à leur Enrichissement par l’usage*. PhD thesis, Université de Toulouse.
- [Mekni, 2010] Mekni, M. (2010). *Automated Generation of Geometrically-Precise and Semantically-Informed Virtual Geographic Environments Populated with Spatially-Reasoning Agents*. Universal-Publishers.
- [Merkus et al., 2004] Merkus, P., Desurmont, X., Jaspers, E. G. T., Wijnhoven, R. G. J., Caignart, O., f Delaigle, J., and Favoreel, W. (2004). Candela - integra-

- ted storage, analysis and distribution of video content for intelligent information systems.
- [Moore, 1991] Moore, A. (1991). An introductory tutorial on kd-trees. Technical Report Technical Report No. 209, Computer Laboratory, University of Cambridge, Pittsburgh, PA.
- [Mountain and Raper, 2001] Mountain, D. and Raper, J. (2001). Modelling human spatio-temporal behaviour : a challenge for location based services. In *Proceedings of The 6th International Conference on GeoComputation*.
- [Nick Kizoom, 2008] Nick Kizoom, P. M. (2008). A transmodel based xml schema for the google transit feed specification with a gdfs / transmodel comparison. Technical report, Kizoom.
- [Pajor, 2009] Pajor, T. (2009). Multi-modal route planning. Master's thesis, Karlsruhe Institute of Technology, Germany.
- [Papadias et al., 2003] Papadias, D., Zhang, J., Mamoulis, N., and Tao, Y. (2003). Query processing in spatial network databases. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03*, pages 802–813. VLDB Endowment.
- [Parent et al., 2013] Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M. L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., and Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4) :42 :1–42 :32.
- [Parent et al., 2006] Parent, C., Spaccapietra, S., and Zimányi, E. (2006). *Conceptual Modeling for Traditional and Spatio-Temporal Applications : The MADS Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [Patino et al., 2011] Patino, L., Bremond, F., and Thonnat, M. (2011). Incremental learning on trajectory clustering. In Remagnino, P., Monekosso, D., and Jain, L., editors, *Innovations in Defence Support Systems - 3*, volume 336 of *Studies in Computational Intelligence*, pages 47–70. Springer Berlin Heidelberg.
- [Peng and Kim, 2008] Peng, Z.-R. and Kim, E. (2008). A standard-based integration framework for distributed transit trip planning systems. *Journal of Intelligent Transportation Systems*, 12(1) :13–28.
- [Pereira et al., 2008] Pereira, F., Vetro, A., and Sikora, T. (2008). Multimedia retrieval and delivery : Essential metadata challenges and standards. *Proceedings of the IEEE*, 96(4) :721–744.

- [Pfoser and Theodoridis, 2003] Pfoser, D. and Theodoridis, Y. (2003). Generating semantics-based trajectories of moving objects. *Computers, Environment and Urban Systems*, 27(3) :243 – 263.
- [Raimond et al., 2012] Raimond, A.-M. O., Couronné, T., Fen-Chong, J., and Smoreda, Z. (2012). Le paris des visiteurs étrangers, qu'en disent les téléphones mobiles - inférence des pratiques spatiales et fréquentations des sites touristiques en ile-de-france. *Revue Internationale de Géomatique*, 22(3) :413–437.
- [Raphael Troncy, 2011] Raphael Troncy, Benoit Huet, S. S. (2011). *Multimedia Semantics : Metadata, Analysis and Interaction*. Wiley.
- [Retscher, 2007] Retscher, G. (2007). Ubiquitous positioning solutions for pedestrian navigation. In *International Society for Photogrammetry and Remote Sensing*.
- [Saglio and Moreira, 2000] Saglio, J.-M. and Moreira, J. (2000). Oporto : A realistic scenario generator for moving objects. *GEOINFORMATICA*, 5 :426–432.
- [Sana, 2013] Sana (2013). A survey of indoor localization techniques. *Journal of Electrical and Electronics Engineering (IOSR-JEEE)*, 6(3) :69–76.
- [Sandu-Popa and Zeitouni, 2012] Sandu-Popa, I. and Zeitouni, K. (2012). Modeling and querying mobile location sensor data. In *The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2012)*, page 18pp, Espagne.
- [Sandu Popa et al., 2011] Sandu Popa, I., Zeitouni, K., Oria, V., Barth, D., and Vial, S. (2011). Indexing in-network trajectory flows. *The VLDB Journal*, 20(5) :643–669.
- [SanMiguel et al., 2009] SanMiguel, J., Martinez, J., and Garcia, A. (2009). An ontology for event detection and its application in surveillance video. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 220–225.
- [Sato, 2005] Sato, I. (2005). A location model for pervasive computing environments. In *Proceedings of IEEE 3rd International Conference on Pervasive Computing and Communications (PerCom'05)*, pp,215-224, IEEE Computer Society. Press.
- [Schneider, 2009] Schneider, M. (2009). Moving objects in databases and gis : State-of-the-art and open problems. In Navratil, G., editor, *Research Trends in Geographic Information Science*, Lecture Notes in Geoinformation and Cartography, pages 169–187. Springer Berlin Heidelberg.

- [Scuturici and Ejigu, 2006] Scuturici, V.-M. and Ejigu, D. (2006). Positioning support in pervasive environments. In *Pervasive Services, 2006 ACS/IEEE International Conference on*, pages 19–26.
- [Sèdes, 1998] Sèdes, F. (1998). *Bases documentaires-Hyperbases : Proposition d'un modèle générique et contribution à la spécification d'un langage pour l'intégration et la manipulation d'informations semi-structurées*. PhD thesis, Université Paul Sabatier, Toulouse, France.
- [Sèdes et al., 2012] Sèdes, F., Calvary, G., Delot, T., and Tigli, J.-Y. (2012). *Computer Science and Ambient Intelligence*. ISBN 9781848214378. Wiley Interscience Publications, <http://www.wiley.com>.
- [Sedes et al., 2012] Sedes, F., Sulzer, J., Marraud, D., Mulat, C., and Cepas, B. (2012). *Intelligent Video Surveillance Systems*, chapter A Posteriori Analysis for Investigative Purposes. Wiley.
- [Shahabi et al., 2010] Shahabi, C., Banaei-Kashani, F., Khoshgozaran, A., Nocera, L., and Xing, S. (2010). Geodec : A framework to visualize and query geospatial data for decision-making. *IEEE Multimedia*, 17(3) :14–23.
- [Shen et al., 2011] Shen, Z., Arslan Ay, S., Kim, S. H., and Zimmermann, R. (2011). Automatic tag generation and ranking for sensor-rich outdoor videos. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM'11, pages 93–102, New York, NY, USA. ACM.
- [Simon and Fröhlich, 2007] Simon, R. and Fröhlich, P. (2007). A mobile application framework for the geospatial web. In *Proceedings of the 16th International Conference on World Wide Web*, WWW'07, pages 381–390. ACM.
- [Snoek and Worring, 2009] Snoek, C. G. M. and Worring, M. (2009). Concept-based video retrieval. In *Foundations and Trends in Information Retrieval*, volume 2, pages 215–322.
- [Song and Roussopoulos, 2001] Song, Z. and Roussopoulos, N. (2001). K-nearest neighbor search for moving query point. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, SSTD '01, pages 79–96, London, UK, UK. Springer-Verlag.
- [Spaccapietra and Parent, 2011] Spaccapietra, S. and Parent, C. (2011). Adding meaning to your steps (keynote paper). In Jeusfeld, M., Delcambre, L., and Ling, T.-W., editors, *Conceptual Modeling - ER 2011*, volume 6998 of *Lecture Notes in Computer Science*, pages 13–31. Springer Berlin Heidelberg.
- [Spaccapietra et al., 2008] Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., and Vangenot, C. (2008). A conceptual view on trajectories. *Data Knowl. Eng.*, 65(1) :126–146.

- [Stojanovic et al., 2005] Stojanovic, D., Djordjevic-Kajan, S., and Predic, B. (2005). Incremental evaluation of continuous range queries over objects moving on known network paths. In Li, K.-J. and Vangenot, C., editors, *Web and Wireless Geographical Information Systems*, volume 3833 of *Lecture Notes in Computer Science*, pages 168–182. Springer Berlin Heidelberg.
- [Tao et al., 2002] Tao, Y., Papadias, D., and Shen, Q. (2002). Continuous nearest neighbor search. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 287–298. VLDB Endowment.
- [Tian et al., 2008] Tian, Y.-l., Brown, L., Hampapur, A., Lu, M., Senior, A., and Shu, C.-f. (2008). Ibm smart surveillance system (s3) : event based video surveillance system with an open and extensible framework. *Machine Vision and Applications*, 19(5-6) :315–327.
- [Tulp and Siklóssy, 1991] Tulp, E. and Siklóssy, L. (1991). Searching time-table networks. *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 5 :189–198.
- [Vazirgiannis and Wolfson, 2001] Vazirgiannis, M. and Wolfson, O. (2001). A spatio-temporal model and language for moving objects on road networks. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases*, SSTD'01, pages 20–35, London, UK, UK. Springer-Verlag.
- [Wåhslén et al., 2010] Wåhslén, J., Lindh, T., and Eriksson, M. (2010). A novel approach to multi-sensor data synchronization using mobile phones. In *Proceedings of the Fifth International Conference on Body Area Networks*, BodyNets '10, pages 171–174, New York, NY, USA. ACM.
- [Wang et al., 2012] Wang, H., Jin, P., Zhao, L., Zhang, L., and Yue, L. (2012). Generating semantic-based trajectories for indoor moving objects. In Wang, L., Jiang, J., Lu, J., Hong, L., and Liu, B., editors, *Web-Age Information Management*, volume 7142 of *Lecture Notes in Computer Science*, pages 13–25. Springer Berlin Heidelberg.
- [Wannous, 2014] Wannous, R. (2014). *Trajectory ontology inference considering domain, temporal and spatial dimensions : Application to marine mammals*. PhD thesis, University of La Rochelle, France.
- [Want, 2006] Want, R. (2006). Rfid explained : A primer on radio frequency identification technologies. In *Synthesis Lectures on Mobile and Pervasive Computing*, volume 1, pages 1–94.
- [Want et al., 1992] Want, R., Hopper, A., Falcão, V., and Gibbons, J. (1992). The active badge location system. *ACM Trans. Inf. Syst.*, 10(1) :91–102.



- [Wong, 2013] Wong, J. C. (2013). Use of the general transit feed specification (gtfs) in transit performance measurement.
- [Yuan and Schneider, 2010] Yuan, W. and Schneider, M. (2010). Supporting continuous range queries in indoor space. In *Proceedings 11th International Conference on Mobile Data Management (MDM)*, pages 209–214. IEEE Computer Society.
- [Zandbergen, 2009] Zandbergen, P. A. (2009). Accuracy of iphone locations : A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 13 :5–25.
- [Zhang et al., 2011a] Zhang, J., Liao, F., Arentze, T., and Timmermans, H. (2011a). A multimodal transport network model for advanced traveler information systems. *Procedia - Social and Behavioral Sciences*, 20(0) :313 – 322.
- [Zhang et al., 2003] Zhang, J., Zhu, M., Papadias, D., Tao, Y., and Lee, D. L. (2003). Location-based spatial queries. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD’03, pages 443–454, New York, NY, USA. ACM.
- [Zhang et al., 2011b] Zhang, L., Jing-Quan, L., Zhou, K., Gupta, S. D., Li, M., Wei-Bin, Z., A., M. M., and A., M. J. (2011b). Traveler information tool with integrated real-time transit information and multimodal trip planning : Design and implementation. Technical report, National Research Council, Washington, DC, ETATS-UNIS.
- [Zhao et al., 2011] Zhao, L., Jin, P., Zhang, L., Wang, H., and Lin, S. (2011). Developing an oracle-based spatio-temporal information management system. In *Proceedings of the 16th International Conference on Database Systems for Advanced Applications*, DASFAA’11, pages 168–176. Springer-Verlag.
- [Zheng et al., 2010] Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y. (2010). Understanding transportation modes based on gps data for web applications. *ACM Trans. Web*, 4(1) :1–36.
- [Zheng et al., 2009] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW’09, pages 791–800, New York, NY, USA. ACM.