

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de 8 Mai 1945-Guelma-
Faculté des Mathématiques, d'Informatique et des Sciences de la matière
Département d'Informatique



Mémoire de fin d'études Master

Filière : Informatique

Option :

Science et technologie de l'information et de la communication

Thème

**Une nouvelle structure d'indexation
pour les données du processus de suivi des objets
dans l'Internet des objets vidéo**

Encadré Par :
Dr Kouahla Zineddine

Présenté par :
Allele Imane

Membres du jury :
Dr Zendaoui Zakaria
Dr Lebsir Rabah

Septembre 2021

Remerciements

En premier lieu, on tient à remercier mon Dieu ALLAH qui à donner la force à achever ce projet.

Je tiens à exprimer toute ma reconnaissance à mon directeur de mémoire, Docteur Kouahla Zineddine. Je la remercie de m'avoir encadré, orienté, aidé et conseillé.

Je remercie en particulier Docteur Benrazek Alaeddine pour l'aide qu'il a fournie et les connaissances qu'il a su me transmettre. Je le remercie également pour sa disponibilité et la qualité de ses conseils.

Je tiens à remercier les membres du jury Docteur Zendaoui Zakaria et Docteur Labsir Rabah pour leur présence, leur lecture attentive de mon mémoire ainsi que pour les remarques qu'ils m'adresseront lors de la soutenance afin d'améliorer mon travail.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je remercie mes très chers parents, Mouhamed et Ysmahen, qui ont toujours été là pour moi. Je remercie ma sœur Nour elhouda, et mes frères Abd elrahmen et Ilyes, pour leurs encouragements.

Je remercie mes amis Ibtissem, Ilham, Sabri et la promo STIC qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Finalement, je désire remercier au corps professoral et administratif de département d'informatique université 08 mai 1945 Guelma pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants des bonnes formations.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Allele Imane

Résumé

Ce projet examine la problématique du suivi en temps réel des objets mobiles dans les systèmes de vidéosurveillance à grande échelle. Au cours du processus de suivi, le système attribue un identifiant ou une étiquette à chaque objet suivi pour le distinguer des autres objets. Dans ce type de tâche, il est important de conserver ce même identifiant pour les mêmes objets, quels que soient la zone, le moment de leur apparition et la caméra de détection. Cela permet de stocker un maximum d'informations sur l'objet suivi, de diminuer le nombre de changements d'identifiant (**#ID-Sw**), et de diminuer le taux de fragmentation de la trajectoire (**#Frag**) de ces objets. Par conséquent, la qualité du suivi des objets augmente parallèlement.

Pour réaliser l'étiquetage des objets, une quantité massive de données collectées par les caméras doit être recherchée pour trouver l'identifiant d'objet le plus similaire (le plus proche voisin). Malgré la simplicité de cette tâche, elle devient très complexe dans les réseaux de systèmes de vidéosurveillance à grande échelle où les données deviennent très volumineuses et nécessitent beaucoup de temps pour être accomplies, ce qui affecte négativement les performances du système de suivi en temps réel.

Pour résoudre ces problèmes, nous proposons une nouvelle solution pour étiqueter automatiquement des objets multiples pour un suivi efficace en temps réel en utilisant le mécanisme d'indexation. Ce mécanisme permet d'organiser les métadonnées des objets extraits lors de la phase de détection en un arbre binaire inspiré de l'arbre BCCF et la classification hiérarchique ascendante (CAH). Les principaux avantages de cette structure sont (1) sa capacité à indexer des méta-données massives générées par des caméras multiples, (2) sa complexité de recherche logarithmique, qui réduit de manière implicite le temps de réponse de la recherche, et (3) la qualité des résultats de recherche, qui garantit un étiquetage cohérent des objets suivis. La charge est répartie dans l'architecture de l'Internet des objets vidéo (IoVT) afin d'améliorer le traitement des données pour un système à grande échelle. L'évaluation expérimentale a été réalisée sur la base de données MOT16 et un jeu de données public généré par plusieurs caméras au niveau de département informatique de l'Université 8 Mai 1945 Guelma.

Mots clés : Caméras Intelligente, Index, Distribution, Vidéosurveillance, Optimisation.

Abstract

This project addresses the problem of real-time tracking of moving objects in large-scale video surveillance systems. During the tracking process, the system assigns an identifier or label to each tracked object to distinguish it from other objects. In this type of task, it is important to keep the same identifier for the same objects, regardless of the area, the time of their appearance and the detection camera. This allows to store a maximum of information on the tracked object, to decrease the number of identifier changes (**ID-Sw**), and to decrease the fragmentation rate of the trajectory (**Frag**) of these objects. As a consequence, the quality of object tracking increases in the same time.

To perform object labeling, a massive amount of data collected by the cameras must be searched to find the most similar (nearest neighbor) object identifier. Although this task is simple, it becomes very complex in large-scale video surveillance system networks where the data becomes very large and requires a lot of time to be accomplished, which negatively affects the performance of the real-time tracking system.

To solve these problems, we propose a novel solution to automatically label multiple objects for real-time efficient tracking using the indexing mechanism. This mechanism organizes the metadata of the objects extracted during the detection phase into a binary tree inspired by the BCCF tree and the hierarchical ascending classification (HAC). The main advantages of this structure are (1) its ability to index massive metadata generated by multiple cameras, (2) its logarithmic search complexity, which implicitly reduces the search response time, and (3) the quality of the search results, which ensures consistent labeling of the tracked objects. The experimental evaluation was performed on the MOT16 database and a public dataset generated by several cameras at the computer science department of the University, 8 Mai 1945 Guelma.

Keywords : Smart Camera, Index, Distribution, Video Surveillance, Optimization.

Table des matières

Remerciements	I
Résumé	II
Abstract	III
Introduction générale	1
Contexte	1
Problématique	1
Objectifs	2
Organisation du mémoire	2
1 Fondements de l'IdO et l'évolution des paradigmes modernes	3
1.1 Introduction	4
1.2 Internet des Objets -IdO-	4
1.2.1 Définitions	4
1.2.2 Architecture	4
1.2.3 Caractéristiques essentielles	6
1.2.4 Les domaines d'application	7
1.3 Cloud computing	8
1.3.1 Les principes du Cloud computing	9
1.3.2 Les avantages et les inconvénients du Cloud computing	12
1.4 Paradigmes Post-Cloud émergents	13
1.4.1 Les défis du Cloud computing dans l'IdO	13
1.4.2 Le paradigme Cloudlet	14
1.4.3 Le paradigme Fog computing	14
1.4.4 Le paradigme Edge computing	15
1.4.5 Le paradigme Dew computing	15
1.4.6 Le paradigme Mist computing	15
1.5 Analyse comparative des paradigmes de l'IdO	16
1.6 Conclusion	18
2 Les Systèmes de Vidéo-surveillance Intelligents -SVSI-	19
2.1 Introduction	20
2.2 Évolution des systèmes de vidéosurveillance	20
2.2.1 La première génération (1GSS 1960-1980) : Tout analogique	20
2.2.2 La deuxième génération (2GSS 1980-2000) : Le système hybride	20
2.2.3 La troisième génération (3GSS 2000-aujourd'hui) : Tout numérique	21
2.3 Description des techniques d'analytique vidéo dans les systèmes de surveillance	22
2.3.1 Détection des objets	22
2.3.2 Classification des objets	24

2.3.3	Suivi des objets	26
2.3.4	Analyse de comportement	27
2.4	Nouvelle génération de système de vidéo-surveillance : Point de vue infrastructurelle	27
2.4.1	Les limites des systèmes de vidéo-surveillance conventionnels	27
2.4.2	Internet des Objets Vidéo -IdOV-	28
2.5	Systèmes de vidéo-surveillance distribués à grande échelle	29
2.6	Exigences et défis	31
2.7	Conclusion	31
3	Techniques d'indexation	32
3.1	Introduction	33
3.2	Exigences d'indexation des données massives	33
3.3	Les structures d'indexation arborescentes récentes	34
3.3.1	Techniques d'indexation multidimensionnelle	34
3.3.2	Techniques d'indexation métrique	37
3.4	Conclusion	41
4	Conception	42
4.1	Introduction	43
4.2	Architecture du système proposé	43
4.3	Aperçu du fonctionnement du système	44
4.3.1	Détection d'objet	44
4.3.2	Extraction de premier vecteur de caractéristique	45
4.3.3	Identification d'objet	45
4.3.4	Suivi d'objet	45
4.3.5	Insertion d'objet suivi	46
4.4	La structure d'indexation proposée	46
4.4.1	Définition de la structure proposée	46
4.4.2	Identification des objets détecte	46
4.4.3	Insertion des objets identifier	48
4.5	Conclusion	48
	Conclusion et perspectives	49

Table des figures

- 1.1 Définition de l'Internet des Objets [4] 5
- 1.2 L'architecture trois couches de l'IdO [9] 6
- 1.3 L'architecture cinq couches de l'IdO [9] 7
- 1.4 Quelques domaines d'application de l'IdO 8
- 1.5 La définition du Cloud Computing par NIST 11
- 1.6 Les avantages de l'IdO 12
- 1.7 Les inconvénients de l'IdO 12
- 1.8 Les paradigmes Post-Cloud 14

- 2.1 Étapes de base pour la détection et le suivi des objets [47] 23
- 2.2 Exemple de soustraction de l'arrière-plan 23
- 2.3 Exemple de détection d'objets à l'aide de la technique de différenciation temporelle 24
- 2.4 Exemple de détection d'objets à l'aide de la technique de flot optique 24
- 2.5 l'IdOV est une partie de l'IdO 29
- 2.6 Infrastructure de l'IdOV [61] 30
- 2.7 Les exigences relatives à la distribution [64] 31

- 3.1 Exigences d'indexation des données massives [65] 34
- 3.2 Taxonomie des techniques d'indexation 41

- 4.1 Le système de détection YOLO [122] 45
- 4.2 Les différents types de nœuds de l'arbre-UD 46

Liste des tableaux

1.1	Cloud computing vs. Post-Cloud computing [39].	16
1.2	Comparaison entre les paradigmes de Post-Cloud.	17
2.1	Résumé de l'évolution technique des systèmes de surveillance intelligents [45]. . .	22
2.2	Étude comparative des méthodes de détection d'objet [47].	25
2.3	Étude comparative des méthodes de classification d'objet [47].	26
3.1	Tableau comparatif des approches multidimensionnelles	40
3.2	Tableau comparatif des approches métriques [71]	41

Liste des algorithmes

Liste des sigles et acronymes

Introduction générale

Contexte

Aujourd'hui, la protection des lieux publics et privés est devenue une préoccupation majeure. Cette tâche nécessite l'installation de plusieurs caméras de surveillance pour détecter les événements suspects et garantir la sécurité de ces lieux grâce à des systèmes intelligents qui prennent des décisions basées sur l'analyse de la séquence des scènes.

Selon la société d'information économique IHS Markit, le nombre de caméras utilisées à des fins de surveillance dépassera le milliard d'ici fin 2021, soit une augmentation de près de 30 % par rapport aux caméras installées en 2019 [1]. Une autre étude estime que le nombre de caméras dans le monde dépassera 13 milliards en 2030 [2]. En raison de l'énorme croissance du nombre de caméras et de l'énorme quantité d'informations (vidéo/images) collectées, stockées et traitées, la gestion des stations de surveillance (telles que la détection, le suivi et l'identification de personnes en temps réel) est devenue une tâche complexe qui nécessite des solutions robustes.

Problématique

Le suivi des objets en mouvement constitue un problème fondamental de la vision par ordinateur. De plus, il est important pour de nombreuses applications, notamment la sécurité, l'analyse du mouvement humain et la surveillance du trafic. En bref, Le but du suivi d'objets est de suivre la trajectoire des objets détectés par la localisation de leurs positions dans une période de temps spécifique. Pour accomplir ce processus, il est important de différencier les identités de chaque objet suivi afin de conserver le maximum d'informations sur celui-ci.

Afin de réaliser cette tâche et d'extraire les trajectoires des objets suivis, une étiquette ou un identifiant unique est attribué à chaque objet détecté, indépendamment de la zone, du moment de leur apparition ou de la caméra de détection. Le changement de cette étiquette entraîne une augmentation du nombre de changements d'identité ($\#ID-Sw$). Cela conduit à :

1. Augmenter le taux de fragmentation de la trajectoire ($\#Frag$).
2. Diminuer le taux des objets qui ont été suivis pendant au moins 80 % (les plus suivis -MT-) de leur durée de vie.
3. Augmenter le taux d'objets qui ont été suivis pendant au moins 20 % de leur durée de vie (les objets les plus perdus -ML-).

En résultat, la qualité du suivi est également diminuée.

En général, les systèmes actuels effectuent une recherche séquentielle sur toutes les métadonnées, sans prendre en compte le temps nécessaire pour trouver l'objet (complexité linéaire, $O(n)$).

La quantité de métadonnées augmente considérablement avec le temps, le nombre d'objets suivis et le nombre de caméras dans le réseau. Cette augmentation rend le temps de recherche beaucoup plus long. Ainsi, le système peut trouver l'étiquette après que l'objet détecté ait quitté le champ de vision (FoV) de la caméra détectrice. Ce problème est particulièrement présent dans les réseaux de vidéosurveillance en temps réel à grande échelle, où les métadonnées sont devenues des "grosses métadonnées". Dans cette situation, l'organisation des métadonnées pour un accès rapide est obligatoire.

Objectifs

Pour améliorer la qualité du suivi d'objets en temps réel, il est indispensable de remédier au problème décrit ci-dessus. À cette fin, dans cette mémoire, nous proposerons une nouvelle solution pour étiqueter automatiquement plusieurs objets en utilisant le mécanisme d'indexation. Ce mécanisme consiste à structurer les métadonnées extraites des objets pendant leur suivi dans une structure arborescente appelée Arbre-UD (en anglais : Up Down Tree ou UD-tree). Cette structure s'inspire de la méthode CHA (Classification Hiérarchique Ascendante) et de la récente structure d'indexation l'arbre BCCF [3].

Organisation du mémoire

Ce manuscrit est divisé en cinq chapitres et est organisé comme suit :

- **Chapitre 1 : Fondements de l'IdO et l'évolution des paradigmes modernes**
Ce chapitre présente les technologies, les concepts pertinents et les définitions des différents paradigmes informatiques modernes émergeant de l'IdO.
- **Chapitre 2 : Les systèmes de vidéo-surveillance intelligents -SVSI-**
Dans ce chapitre, nous présentons un état de l'art sur les systèmes de vidéosurveillance et leur évolution.
- **Chapitre 3 : Techniques d'indexation**
Ce chapitre est consacré à la gestion des données générées par l'IoT en général et les systèmes de vidéosurveillance en particulier.
- **Chapitre 4 : Conception**
Ce chapitre constitue le cœur de cette mémoire où nous allons présenter l'architecture globale de notre système, expliquer son fonctionnement et la conception de la structure d'indexation que nous proposons.
- **Chapitre 5 : Implémentation et résultats**
Dans ce dernier chapitre, nous validons notre système sur une base de données réelle en deux parties : la première partie vise à évaluer la structure d'indexation et la seconde partie vise à évaluer les performances du système basé sur la structure que nous proposons.

Cette mémoire se termine par une conclusion générale et quelques perspectives dans le domaine.

Chapitre 1

Fondements de l'Internet des Objets et l'évolution des paradigmes informatiques modernes

1.1 Introduction

Ce chapitre présente les technologies, les concepts pertinents et les définitions des différents paradigmes informatiques modernes émergeant de l'IdO. Tout d'abord, dans la Section 1.2, nous présentons en détail la technologie IdO, ses principales définitions et quelques domaines d'application. Ensuite, dans la Section 1.3, nous présentons la technologie du Cloud computing utilisée dans l'IdO comme une solution plus sécurisée avec une puissance de stockage et de calcul importante qui accélère l'évolution des objets connectés. Dans la Section 1.4, nous présenterons brièvement les différents paradigmes post-Cloud émergents pour surmonter les limites du Cloud dans l'IdO. Dans la Section 1.5, nous terminons cette étude par une analyse comparative de tous les paradigmes examinés.

1.2 Internet des Objets -IdO-

1.2.1 Définitions

Internet des Objets ou IdO (**en anglais** Internet of Things ou IoT), est le concept le plus récent et le plus populaire des technologies de l'information (TI). Cependant, malgré le développement rapide de cette technologie, il n'existe pas encore de définition universelle. Pour cela, nous citerons quelques définitions connues et récentes.

Définition 1.1 (IEEE¹ [4]) *L'IdO est un réseau qui relie à l'Internet des "objets" identifiables de manière unique. Les "objets" ont des capacités de détection/actionnement et de programmation potentielle. Grâce à l'exploitation de l'identification unique et de la détection, il est possible de collecter des informations sur le "objet" et de modifier l'état de le "objet" de n'importe où, n'importe quand et par n'importe qui.*

Définition 1.2 (Oracle [5]) *L'Internet des objets (IdO) décrit le réseau d'objets physiques - "choses" - qui sont équipés de capteurs, de logiciels et d'autres technologies dans le but de se connecter et d'échanger des données avec d'autres dispositifs et systèmes sur l'Internet. Ces dispositifs vont des objets domestiques ordinaires aux outils industriels sophistiqués. Avec plus de 7 milliards de dispositifs IdO connectés aujourd'hui, les experts s'attendent à ce que ce nombre passe à 10 milliards d'ici à 2020 et à 22 milliards d'ici à 2025.*

Définition 1.3 (ITU² [6]) *Une infrastructure globale pour la société de l'information, qui permet des services avancés lors de la connexion d'objets (physiques et virtuels) sur la base des informations existantes et des technologies de communication et de communication interopérables.*

Sur la base des définitions citées nous allons présenter l'IdO dans un schéma illustré dans la Figure 1.1 :

1.2.2 Architecture

Comme nous allons dit précédemment, il n'existe pas une définition standard pour l'IdO. Cela signifie qu'il n'existe pas une architecture unique pour ce concept. Plusieurs architectures ont été proposées depuis la naissance d'IdO, mais les plus utilisés sont les deux architectures : trois couches et cinq couches.

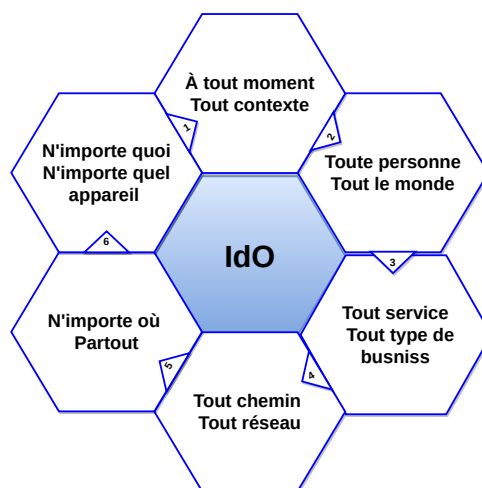


FIG. 1.1 : Définition de l'Internet des Objets [4]

Architecture de l'IdO à trois couches

L'architecture à trois couches est considérée comme une architecture IdO de base [7]. Comme le montre la Figure 1.2, cette architecture est essentiellement composée de trois couches : la couche de perception, la couche réseau et la couche application [8]. Dans ce qui suit, ces couches sont brièvement décrites :

1. **La couche Perception/Physique** : La principale fonction de cette couche est d'identifier sans ambiguïté des objets ou des entités du monde physique, de collecter des données sur leur état physique par des capteurs (comme GPS et le gyroscope pour la position, accéléromètre pour la vitesse, etc.), de transporter les données collectées vers le réseau central ou vers une base de données locale à travers l'utilisation des diverses technologies (tel que RFID³, RCSF⁴, CCP⁵, etc.), et finalement d'effectuer la même action sur l'objet en fonction d'un critère bien défini.
2. **La couche Réseau** : Appelée aussi la couche de transport, elle forme le réseau de communication qui connecte les objets, les dispositifs de réseau et les serveurs. Et aussi le lien de transmission des données collecter par la couche perception vers la couche application, à travers l'utilisation des diverses technologies (tel que BLE, WI-FI, ZigBee, etc.).
3. **La couche Application** : Il s'agit de la couche supérieure de l'architecture à trois couches. La couche application analyse et traite les informations provenant de la couche perception et de la couche réseau, réalisant ainsi l'application IdO. En bref, la couche d'application est l'interface entre l'IdO et divers types d'utilisateurs (une personne ou un système), combinés à des besoins spécifiques, pour réaliser diverses applications intelligentes de l'IdO.

Principe de fonctionnement

En bref, le fonctionnement de l'architecture à trois couches est le suivant : la première étape consiste à collecter des données à partir des capteurs connectés par le biais de diverses technologies situées dans la couche de perception. Ces données sont transférées à la couche

³RFID : Radio-frequency identification

⁴RCSF : Réseaux de capteurs sans fil

⁵CCP : communication en champ proche

application via la couche réseau, qui joue le rôle de lien de communication entre elles. Enfin, la couche application traite, stocke et partage ces données.

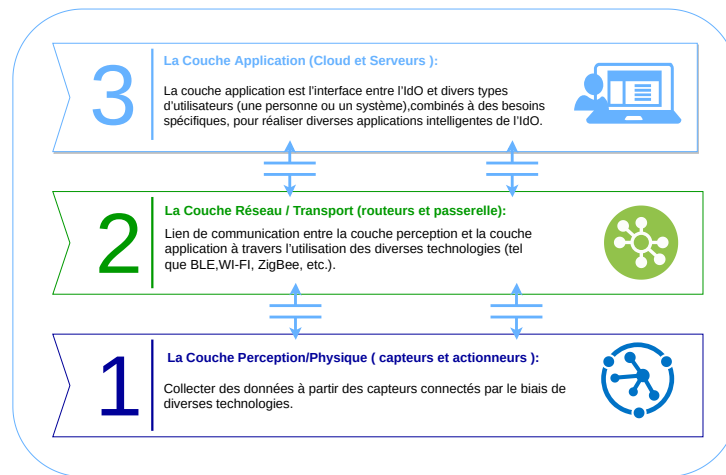


FIG. 1.2 : L'architecture trois couches de l'IdO [9]

Architecture de l'IdO à cinq couches

Avec l'innovation de l'IdO, l'architecture à trois couches reste incapable d'expliquer les caractéristiques et les connotations de l'IdO [10]. Par conséquent, les chercheurs proposent une autre architecture à cinq couches, comme le montre la Figure 1.3. Cette architecture est essentiellement basée sur l'architecture à trois couches. Il s'agit d'une architecture IdO centrée sur le cloud, dans laquelle presque tout le traitement des données IdO est effectué dans le Cloud ou sur un serveur distant [11]. La différence entre l'architecture à cinq couches et à trois couches réside dans l'ajout des éléments suivants :

4. **La couche de Traitement** : La couche de traitement, également appelée intergiciel, est chargée de traiter, de stocker et d'analyser les données massives reçues de la couche de transport grâce aux nouvelles technologies de traitement de l'information telles que les bases de données, le traitement intelligent, l'informatique en nuage et l'informatique omniprésente.
5. **La couche Métier (Business)** : La couche métier est la dernière couche de l'architecture IdO à cinq couches. Cette couche est désignée comme le gestionnaire de l'IdO (gestion des applications, modèle commercial pertinent, etc.) et d'autres activités telles que la vie prédictive et la confidentialité des utilisateurs. Cependant, il ne s'agit pas seulement de la gestion, mais aussi de la recherche de modèles commerciaux efficaces et de modèles de profit pour le développement à long terme de l'IdO.

1.2.3 Caractéristiques essentielles

L'IdO repose sur les sept caractéristiques fondamentales citées ci-dessous [12] :

- **Inter-connectivité** : L'une des caractéristiques essentielles d'IdO, c'est que tous les objets peuvent être interconnectés avec l'infrastructure mondiale d'information et de communication.

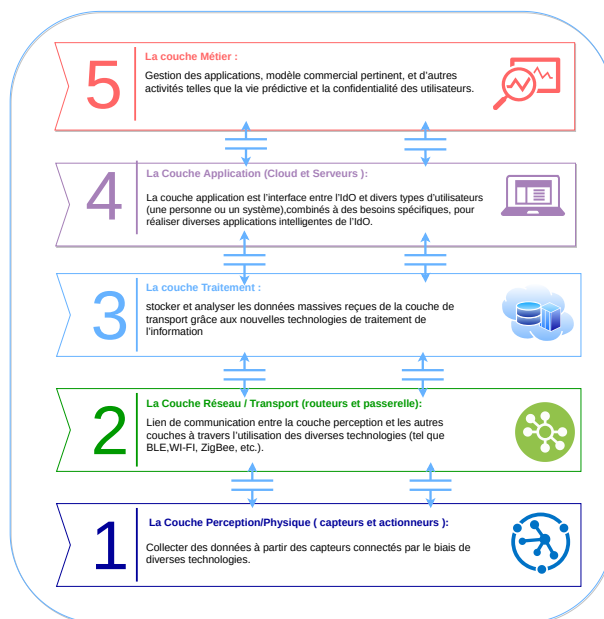


FIG. 1.3 : L'architecture cinq couches de l'IdO [9]

- **Services liés aux objets :** La capacité de l'IdO de garantir des services (tel que la protection de la vie privée et la cohérence sémantiques entre les objets physiques et les objets virtuels qui leur sont associés) associés aux objets dans leur cadre de limitation.
- **Hétérogénéité :** La variété des plateformes matérielles et des réseaux d'IdO conduit à cette hétérogénéité des dispositifs de l'IdO.
- **Changements dynamiques :** Le changement dynamique d'état des dispositifs (connecté/déconnecté, etc.), le contexte (la localisation et la vitesse, etc.), de même que le nombre de dispositifs peuvent changer de façon dynamique.
- **Une échelle énorme :** Avec plus de sept milliards de dispositifs IdO connectés aujourd'hui [5] qui ont besoin d'être gérés et qui communiquent entre eux, l'interprétation des données générer grâce à ce nombre exponentiel de ces dispositifs et la nécessité d'un traitement efficace de ces derniers seront encore plus critiques.
- **Sécurité :** L'une des caractéristiques les plus importantes de l'IdO, c'est la sécurité. Cela comprend la sécurité de nos données personnelles, la sécurité de notre confort physique, la sécurité des points terminaux, des réseaux, etc.
- **Connectivité :** La connectivité fait appel à l'accessibilité (accéder à un réseau) et à la compatibilité (la capacité de production et de consommation de données en général) du réseau.

1.2.4 Les domaines d'application

Aujourd'hui, on peut dire que l'IdO fait partie de notre monde, grâce au nombre de capteurs (les éléments physiques intégrés à l'IdO), les données sont collectées, traitées et réutilisées pour développer notre vie urbaine. Avec cette croissance, l'IdO touche presque tous les domaines, en particulier les domaines énumérés ci-dessous [13, 14] :

- **Les Environnements intelligents** : peut-être des maisons incluses des appareils électroniques tels que les téléviseurs, les machines à laver, les réfrigérateurs, les climatiseurs et de nombreux autres dispositifs sont contrôlés très efficacement. Les villes intelligentes et aussi peut-être des usines intelligentes [15].
- **Contrôle automatique, intelligent et à distance des unités de fabrication et de production** : Les environnements intelligents permettent également d'améliorer l'automatisation des installations industrielles grâce au déploiement massif d'étiquettes RFID associées aux pièces de production [16].
- **Santé et E-santé** : Les appareils de l'IdO peuvent aider les individus à contrôler et faire des diagnostics automatiques pour leur propre état de santé, comme le poids, la masse corporelle, les habitudes de sommeil et le taux d'activité quotidienne et aussi le contrôle intelligent et à distance des malades par les médecins [15].
- **Agriculture intelligente** : Grâce aux données relatives à l'atmosphère, à l'eau, au sol et d'autres données collectées en temps réel à travers les capteurs de l'IdO, le rendement et la qualité de la production agricole a été améliorée [15].

Il existe des autres domaines d'application de l'IdO présenté dans la Figure 1.4.

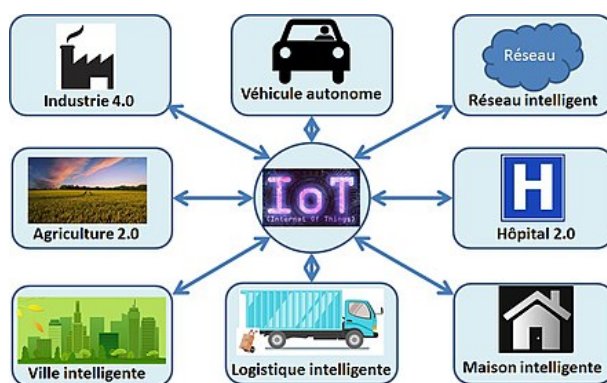


FIG. 1.4 : Quelques domaines d'application de l'IdO

1.3 Cloud computing

Le Cloud computing ou bien "informatique en nuage", est l'une des paradigmes bien connus dans le monde informatique depuis quelques années, qui offre différents niveaux de service à l'utilisateur, grâce à l'amélioration des réseaux, la puissance des serveurs et la rapidité de connexion.

Plusieurs communautés mondiales définissent ce qu'est le Cloud computing. Le reste de cette Section présente la définition de la communauté NIST ⁶ et quelques autres définitions.

Définition 1.4 (NIST, [17]) *Cloud computing est un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement approvisionnées et libérées avec un minimum d'efforts de gestion ou d'interaction avec le fournisseur de services. Ce modèle de nuage est composé de cinq caractéristiques essentielles, de trois modèles de service et de quatre modèles de déploiement.*

⁶NIST : National Institute of Standards and Technology

Définition 1.5 (Chandrasekaran, [18]) *Le Cloud computing est un modèle de traitement de l'information dans lequel les capacités informatiques administrées de manière centralisée sont fournies sous forme des services, en fonction des besoins, à travers le réseau à une variété de dispositifs en contact avec l'utilisateur.*

Définition 1.6 (Griffith, [19]) *En termes simples, le Cloud computing consiste à stocker et à accéder à des données et à des programmes sur l'internet plutôt que sur le disque dur de votre ordinateur. Le Cloud n'est qu'une métaphore de l'internet. Il remonte à l'époque des organigrammes et des présentations qui représentaient la gigantesque ferme de serveurs dans les bureaux de l'entreprise.*

1.3.1 Les principes du Cloud computing

Comme défini par NIST (dans la Section 1.3) le Cloud computing repose sur trois modèles de base d'offre de services, quatre modèles de déploiement et cinq caractéristiques fondamentales comme le montre la Figure 1.5.

Les trois modèles de services du Cloud computing

Selon la définition de NIST [17], trois modèles de services peuvent être offerts sur le Cloud, ces modèles sont brièvement décrites :

- *SaaS (Software as a Service)* : Dans ce modèle de service, la principale possibilité offerte au client est d'utiliser les applications du fournisseur qui fonctionnent sur le Cloud. Les applications sont accessibles à travers une interface client de petite taille, telle qu'un programme web, à partir de divers appareils clients.
- *PaaS (Plateforme as a Service)* : Ce modèle de service fournit aux clients un environnement d'exécution pour développer et déployer des logiciels et des applications sur le Web. Il offre également la liberté de contrôler ces applications et les paramètres de configuration de l'environnement, mais il ne gère ni ne contrôle l'infrastructure du Cloud.
- *IaaS (Infrastructure as service)* : Le modèle de service IaaS offre à ses clients les mêmes fonctionnalités que PaaS, Ainsi, le client du service IaaS est responsable de la sécurité de tous les éléments du système qu'il utilise, à l'exception la sécurité physique.

D'autres modèles de services Cloud sont apparus récemment, tels que [20] :

- *CaaS (Containers as a service)* : Les conteneurs en tant que service (CaaS) sont un autre modèle de service en Cloud qui permet aux utilisateurs de télécharger, d'organiser, de démarrer, d'arrêter, de mettre à l'échelle et de gérer les conteneurs, les applications et les clusters. Ces processus sont mis en œuvre au moyen d'une virtualisation basée sur des conteneurs, d'une interface de programmation d'applications (API) ou d'une interface de portail Web. CaaS aide les utilisateurs à créer des applications conteneurisées sécurisées et évolutives par le biais de centres de données sur site ou dans le Cloud. Les conteneurs et les clusters sont utilisés comme un service avec ce modèle et sont déployés dans le Cloud ou dans des centres de données sur site.

- *FaaS (Function as a service)* : Le fonctionnement en tant que service est un service destiné aux utilisateurs qui n'ont besoin du Cloud computing que pour des fonctions ou des composants individuels de leurs applications. Avec ce service, le matériel physique du client, le système d'exploitation de la machine virtuelle et la gestion du logiciel du serveur web sont tous gérés automatiquement par le fournisseur de services.

Les quatre modèles de déploiement du Cloud computing

Les trois types de services de Cloud abordés précédemment visent à répondre aux besoins des clients à différents niveaux d'engagement dans la gestion du matériel et des logiciels informatiques. Il existe une corrélation directe entre la taille de l'organisation et le choix du type de service de Cloud. Les modèles de déploiement des services sont liés aux types de services. Ces modèles de déploiement peuvent être classés en Cloud public, Cloud privé, Cloud hybride et Cloud communautaire [21].

- *Cloud Public* : Le Cloud public est accessible au grand public et fournit plusieurs ressources et services, tels que des applications, des serveurs et du stockage, à la demande, via Internet. Il peut être détenu, géré et exploité par une entreprise, une institution universitaire ou une organisation gouvernementale, ou une combinaison des deux. Il existe à l'intérieur du fournisseur de Cloud [17, 22]. Les principaux Clouds publics comprennent Amazon Web Services (AWS), Microsoft Azure, IBM Bluemix, Google Cloud [23].
- *Cloud Privé* : Le Cloud privé est un type spécifique de Cloud déployé pour une utilisation exclusive (un seul consommateur ou une organisation comprenant plusieurs consommateurs). Les organisations peuvent détenir et gérer unilatéralement des Clouds privés, ou attribuer cette responsabilité à un tiers (c'est-à-dire un fournisseur de Cloud) ou à une combinaison des deux. L'infrastructure de Cloud privé peut être interne ou externe à l'entreprise. Les principaux Clouds privés comprennent Amazon Virtual Private Cloud (Amazon VPC), Eucalyptus, OpenStack, et VMWare.
- *Cloud Hybride* : Ce type de Cloud est composé de deux ou plusieurs infrastructures distinctes (privé, public ou communautaire). Ces structures de Cloud uniques constituent toujours une seule entité, elles peuvent être contraintes ensemble grâce à une technologie propriétaire qui permet la portabilité des données et des applications.
- *Cloud Communautaire* : Le Cloud communautaire est une forme de Cloud public pour une communauté spécifique (des consommateurs partagent des intérêts communs.), par exemple mission, exigences de sécurité, politique et considérations de conformité. Elle peut être détenue, gérée et exploitée par une ou plusieurs organisations communautaires, des tiers ou une combinaison d'entre eux, et peut exister à l'intérieur ou à l'extérieur des locaux.

Les caractéristiques essentielles du Cloud computing

Comme défini (dans la Section 1.3), le Cloud computing repose sur cinq caractéristiques essentielles citées ci-dessous :

1. **Libre-service à la demande** : cette caractéristique offre aux utilisateurs la capacité d'utiliser les services de Cloud computing (tel que la capacité de calcul, stockage, etc.),

sans avoir besoin d'une interaction directe fournisseur-utilisateur, cela réduit le coût d'administration informatique et le temps entre la demande de service et la livraison de ce dernier.

2. **Large accès au réseau** : le Cloud computing fournit des services aux utilisateurs finaux à distance, dans tous les temps, et quelle que soit la plateforme utilisée par l'utilisateur (par exemple, les téléphones mobiles, les tablettes, les ordinateurs portables et les postes de travail.), grâce à des mécanismes standard qui favorisent l'utilisation hétérogène des plateformes.
3. **La mise en commun des ressources** : Grâce à cette caractéristique la gestion de Cloud computing devient automatique, les ressources des fournisseurs sont mises en commun pour servir plusieurs utilisateurs à l'aide d'un modèle multiloculaire. Les différentes ressources physiques et virtuelles sont allouées et réallouées dynamiquement en fonction des besoins des utilisateurs, sans connaître la localisation précise des ressources fournies et sans l'intervention de ces derniers pour les contrôler.
4. **Une élasticité rapide** : c'est la capacité de l'infrastructure de Cloud computing de s'adapter aux différentes charges de travail des utilisateurs, à travers l'augmentation et la diminution des ressources fournies automatiquement ou libérées ces ressources d'une manière élastique.
5. **Service mesuré ou paiement à l'usage** : le Cloud computing dispose d'un mécanisme de paiement à l'utilisation en fonction de la durée et de la quantité des ressources utilisées, afin de contrôler et optimiser automatiquement la gestion des ressources, et assure la transparence entre les fournisseurs et les utilisateurs finaux.

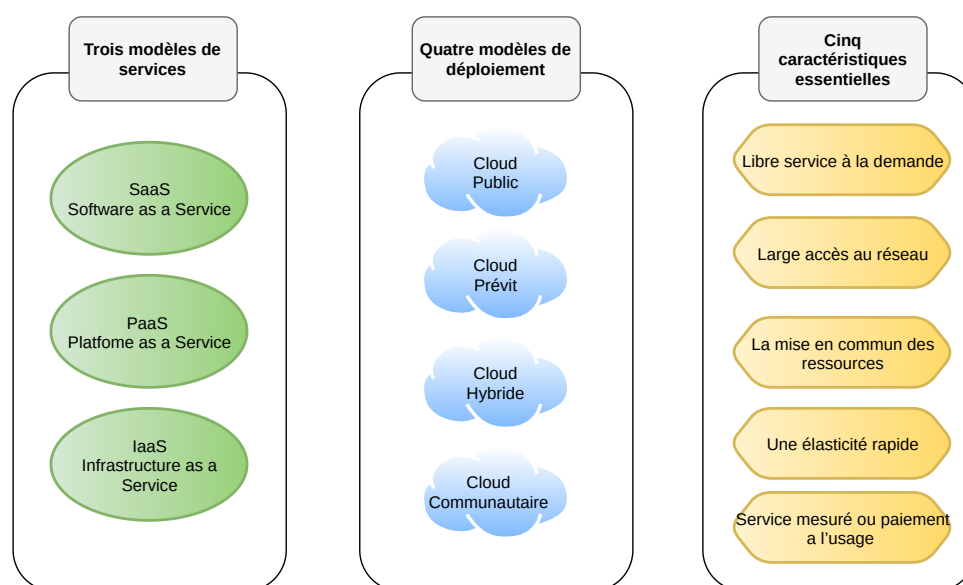


FIG. 1.5 : La définition du Cloud Computing par NIST

1.3.2 Les avantages et les inconvénients du Cloud computing

Les avantages

Le Cloud computing comprennent principalement sur les avantages citez ci-dessous [24], et présente dans la Figure 1.6 :

- La réduction des coûts grâce à la virtualisation des ressources.
- L'extensibilité et la flexibilité des ressources fournies.
- La fiabilité des services.
- La facilité de la maintenance.
- L'accès à distance aux ressources et aux services.

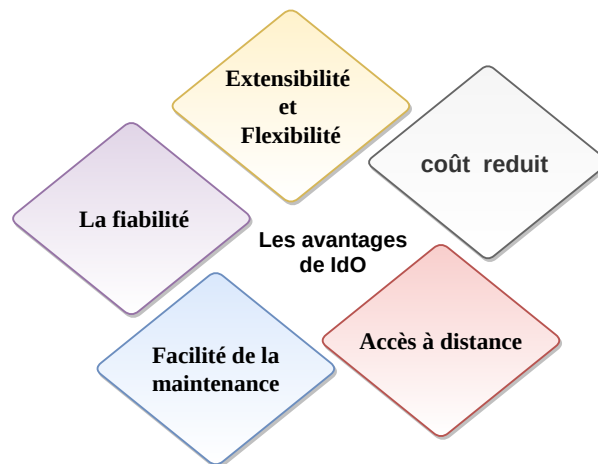


FIG. 1.6 : Les avantages de l'IdO

Les inconvénients

Malgré les avantages apportés par le Cloud, il introduit également des défis, tels que [25] :

- Problèmes de sécurité : Le problème majeur de Cloud est le risque de sécurité et la confidentialité, et jusqu'à maintenant la sécurité de Cloud sera amélioré, mais n'est pas encore idéal.
- L'accès au Cloud repose sur la disponibilité de la connexion internet.
- La conformité : C'est l'une des principaux inconvénients de Cloud pour toute personne utilisant des services de Cloud ou d'hébergement. Lorsqu'il s'agit de sauvegarder ou de stocker en Cloud, il faut respecter les réglementations et les lois du secteur.

FIG. 1.7 : Les inconvénients de l'IdO

1.4 Paradigmes Post-Cloud émergents

Le Cloud computing a été largement accepté au cours de la dernière décennie et son utilisation a augmenté rapidement. Cette Section donne un bref aperçu des défis du Cloud dans le cadre de l'IdO et des principaux paradigmes post-Cloud émergents pour surmonter ces limites. Ces paradigmes post-Cloud comprennent le CDEF computing, c'est-à-dire le Cloudlet, le Dew, le Edge, le Fog et le Mist computing.

1.4.1 Les défis du Cloud computing dans l'IdO

Avec le développement rapide de l'Internet des objets (IdO) que le monde a connu récemment, son intégration avec l'infrastructure du Cloud était nécessaire et inévitable. Cette intégration est connue sous le nom de " Cloud of Things " ou CoT. Elle constitue l'une des principales tendances de la recherche visant à relever les défis de l'IdO. Malgré les avantages apportés par l'utilisation du paradigme CoT, il introduit également de nouveaux défis pour les applications IdO qui ne peuvent être résolus par le Cloud computing, tels que :

- *Le temps de latence* : De nombreuses applications IdO, telles que systèmes de vidéosurveillance et les applications de santé, nécessitent une faible latence. Ces exigences sont en dehors des capacités des services du Cloud en raison de la grande distance entre les objets de l'IdO et le Cloud, où le nombre de sauts que le paquet doit faire pour voyager et obtenir la réponse est important et peut ne pas être souhaité.
- *L'interruption des services* : Dans les systèmes mentionnés précédemment, la connectivité réseau instable et intermittente avec les services Cloud est généralement la priorité absolue. Il n'est pas facile de la fournir en raison de la longue distance entre le Cloud et les dispositifs de l'IdO distants.
- *Les ressources limitées des objets IdO* : La plupart des objets de l'IdO ont des ressources limitées en termes de mémoire, de traitement et de puissance. Avec ces ressources limitées, ces objets ne seront pas en mesure d'interagir directement avec le Cloud sans inclure au moins un matériel puissant comme intermédiaire pour se connecter au Cloud.
- *La distribution géographique* : Dans l'IdO, les objets connectés sont déployés dans des zones géographiquement distribuées. Comme le nombre de ces objets a augmenté ces dernières années, leur espace est devenu de très grandes. En raison de cette répartition à grande échelle, il est très difficile de trouver un emplacement d'infrastructure de Cloud qui réponde à toutes les exigences des applications IdO.
- *Goulets d'étranglement des réseaux hétérogènes* : Dans une certaine mesure, le développement de la technologie des réseaux a offert un support technique solide et une certaine garantie pour l'application du Cloud computing. Néanmoins, en raison du traitement centralisé du Cloud computing, la bande passante et la vitesse du réseau limitent toujours ses performances et son efficacité.

Le secteur de la recherche et l'industrie ont déployé de grands efforts pour trouver des solutions à ces limitations. En conséquence, de nouveaux paradigmes ont été proposés. Comme le montre dans la Figure 1.8, ces paradigmes sont fondés sur le concept d'extension du Cloud computing du centre de données à l'endroit où se trouvent les objets, en éliminant les goulets d'étranglement du réseau et en améliorant les vitesses de traitement et l'efficacité des services.

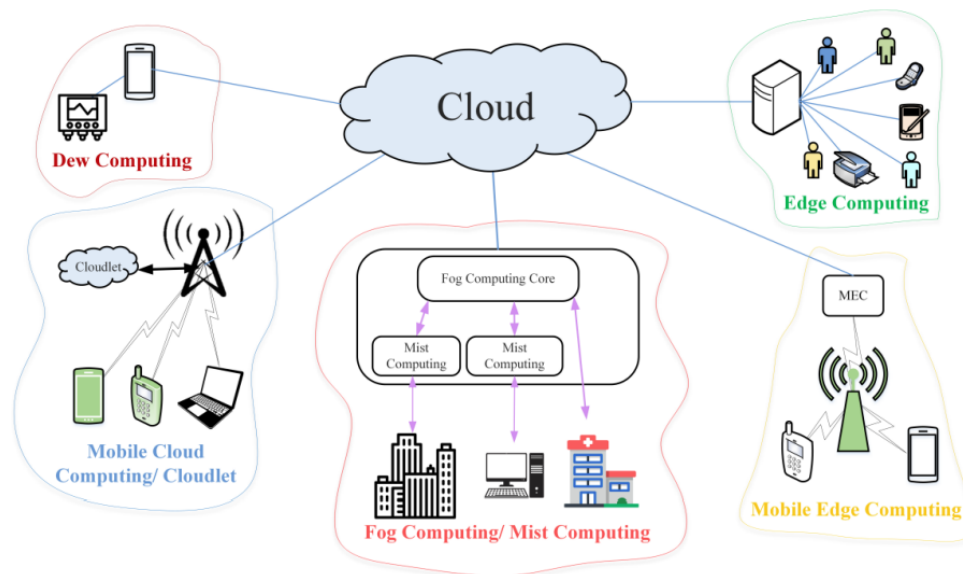


FIG. 1.8 : Les paradigmes Post-Cloud

1.4.2 Le paradigme Cloudlet

Le mot "cloudlet" existe depuis longtemps. Avec des nombreuses significations, il a commencé à être utilisé dans l'orchestration informatique en 2009 [26].

Définition 1.7 (Pan, [27]) *Cloudlet est un ordinateur de confiance ou un cluster d'ordinateurs avec des ressources abondantes et une bonne connexion à Internet, qui peut être utilisé par les objets à proximité.*

Selon la définition, le Cloudlet est un ordinateur ou cluster d'ordinateurs, ces derniers sont place au milieu d'une hiérarchie à trois niveaux entre les objets et le Cloud, afin de placer des centres de données du Cloud à petite échelle à la périphérie de réseau [27].

1.4.3 Le paradigme Fog computing

Le paradigme du Fog (**F**rom **c**ORE to **e**dGE) computing est l'un des plus récents paradigmes Post-Cloud. Il a été proposé pour la première fois par Flavio Bonomi, vice-président de Cisco Systems, en septembre 2011 lors d'une conférence [27] et a été expliqué en détail en 2012 lors de la conférence "Mobile Cloud computing" [28].

Définition 1.8 (Vaquero, [29]) *Le Fog computing est un paradigme informatique distribué, né de la nécessité de déplacer les services de Cloud computing (calcul, stockage et mise en réseau) vers les périphériques. Dans ce cas, les dispositifs de bord sont appelés "objets" et comprennent une variété de capteurs, actionneurs, dispositifs mobiles, systèmes intégrés, etc.*

Le principe de ce paradigme repose sur la nécessité de déplacer les ressources informatiques (réseau, calcul, stockage, etc.) placées sur le Cloud vers le cœur de réseau (comme les routeurs, les commutateurs de routage, les multiplexeurs, etc.) à proximité des objets de l'IdO [27].

1.4.4 Le paradigme Edge computing

Le premier article mentionne le terme "le calcul à la périphérie (computing at the edge)" a été publié en 2004 [30], mais il est considéré comme un article d'accumulation et ce n'est pas l'origine de "Edge computing" [27]. Selon [31], le "Edge computing" est apparu pour la première fois en octobre 2015.

Définition 1.9 (Shi, [32]) *Le Edge computing désigne les technologies permettant d'effectuer des calculs à la périphérie du réseau, sur des données en aval pour le compte de services en Cloud et des données en amont pour le compte de services IdO.*

Le principe de ce paradigme est de placer les applications, les données et les services depuis les serveurs centraux du Cloud vers la périphérie de l'internet, à proximité des appareils mobiles ou des capteurs. Dans ce paradigme, les dispositifs terminaux (tels que les capteurs intelligents, les smartphones, les véhicules intelligents et même les serveurs de périphérie spéciaux) agissent à la fois comme utilisateurs et comme producteurs de données [33], ce qui réduit la latence du réseau et l'utilisation de la bande passante associé au Cloud computing. Le Edge computing garantit la continuité des services et des opérations, même si la connexion au Cloud est intermittente [34].

1.4.5 Le paradigme Dew computing

Le Dew computing est un nouveau paradigme informatique qui est né après la large acceptation du Cloud computing. Il est apparu pour la première fois en 2015 [35].

Définition 1.10 (Wang, [36]) *Le Dew computing est un paradigme d'organisation logiciel-matériel de l'ordinateur sur site dans l'environnement de l'informatique en Cloud, dans lequel l'ordinateur sur site fournit des fonctionnalités indépendantes des services en nuage et collabore également avec ces derniers. L'objectif du Dew computing est d'exploiter pleinement le potentiel des ordinateurs sur site et des services du Cloud.*

Le Dew computing utilise des ordinateurs installés sur place pour fournir des micro-services décentralisés indépendante des services du Cloud et qu'ils collaborent également avec les services de ce dernier. Tandis que le Cloud computing utilise des serveurs pour fournir des services centralisés [27].

1.4.6 Le paradigme Mist computing

Cisco introduit un nouveau paradigme, appelé "Mist computing", qui est une version encore plus décentralisée de ce Fog. Il s'agit de placer les outils d'analyse non seulement au cœur et à la périphérie du réseau, mais aussi à l'extrême périphérie, c'est-à-dire au cœur de nombreux appareils. Ces dispositifs sont généralement des nœuds à faible puissance comme les microordinateurs et des microcontrôleurs [37].

Définition 1.11 ([38]) *Mist computing est une couche optionnelle et une sous-classe de Fog computing. Inspirée par la prolifération de l'IdO et l'introduction des applications sociales de l'IdO (par exemple, la détection des foules), l'architecture informatique Mist tente d'optimiser les plateformes évolutives et rentables, l'analyse des données distribuées, l'allocation et l'approvisionnement de ressources limitées et la réduction des temps de réponse.*

Le Mist computing consiste à placer la puissance de calcul directement à la périphérie du réseau, sur les objets de l'IdO. L'objectif de ce rapprochement est d'effectuer l'analyse aussi localement que possible, afin de supporter des actions immédiates, de diminuer la latence, d'augmenter le débit, d'économiser du temps et de la charge réseau, et d'augmenter l'autonomie du système.

1.5 Analyse comparative des paradigmes de l'IdO

Pour avoir une compréhension plus approfondie et plus claire des différents paradigmes de l'IdO, nous avons fait une analyse comparative sur la base de certaines études de chercheurs [39, 40, 41, 42, 27]. Cette analyse réalisée en deux étapes. Une analyse entre le paradigme du Cloud et les paradigmes post-Cloud (résumé dans le tableau 1.1), puis une analyse entre les paradigmes post-Cloud eux-mêmes (examiné dans le tableau 1.2).

Critères de comparaison	Cloud computing	Post-Cloud computing
Architecture (nombre de couches)	deux	plusieurs
Modèle de stockage	centralisé	petit centralisé ou distribué
Lieu d'exécution de calcul	grand centre de donnée	dispositif lui-même ou dispositifs adjacents ou les dispositifs de Edge computing
Séquence de temps d'exécution	par lot (série)	par lot (série) ou parallèle
Distance utilisateur-serveur	loin des utilisateurs	proche des utilisateurs
Utilisateur cible	utilisateurs courants d'internet	principalement des utilisateurs mobiles
Ressources matérielles	grand espace de stockage et capacité de calcul	capacités de stockage et de calcul limité
Géo-distribution	centralisé	petit centralisé ou distribué
Couverture géographique	global	zone locale ou plus large
Types de services	services d'informations collectées à l'échelle globale	services d'information locale pour un environnement de déploiement particulier
Emplacement du service	au sein de l'Internet	a la périphérie des réseaux locaux
Entreprise d'assistance	grande entreprise de services Internet	petits opérateurs et fabricants d'équipements

TAB. 1.1 : Cloud computing vs. Post-Cloud computing [39].

TAB. 1.2 : Comparaison entre les paradigmes de Post-Cloud.

	Cloudlet	Fog	Edge	Dew	Mist
utilisation des appareils terminaux	Non	Oui	Oui	Oui	Oui
Nombre d'utilisateurs /dispositifs	Des milliers	Des milliers aux millions	Des centaines à des milliers	Quelques dispositifs	Des centaines
Latence	Faible	Faible	Faible	Faible	Faible
Réseau d'accès	sans fils	sans fils	sans fils	sans fils et filière	sans fils et filière
Matériels déployés	centre de données	Routeurs, commutateurs, points d'accès, passerelles, etc.	Points d'accès radio, stations de base, etc.	Ordinateurs à domicile	Micro-contrôleur
Distance	Très proche	Très proche	Très proche	Très proche	Très proche
Type de service	Service local	Service local	Service local	Service local	Service local
Architecture	Décentralisé	Hierarchique, Décentralisé, Distribué	Hierarchique, Décentralisé, Distribué	Décentralisé	Centralisé
Mode de fonctionnement	Autonome ou connecté au Cloud	Autonome	Autonome	Autonome	Autonome

De l'analyse comparative entre le paradigme du Cloud et les paradigmes post-Cloud, nous pouvons conclure que l'évolution rapide des technologies matérielles, logicielles et de réseau, a conduit à une augmentation du nombre de dispositifs qui communiquent entre eux ainsi que les données volumineuses collectées à travers eux. Cette augmentation a entraîné de nombreuses difficultés pour le paradigme du Cloud centralisé, ce qui encourage les chercheurs à trouver des solutions efficaces à ces problèmes. Les solutions proposées sont basées sur les paradigmes de post-Cloud(cités dans la Section 1.4). Ces paradigmes diffèrent complètement du paradigme du Cloud computing dans de nombreuses caractéristiques essentielles (telles que l'architecture, le calcul et le stockage des données, etc.). Par contre, ces paradigmes post-Cloud jouent un rôle d'intermédiaire entre les objets ou capteurs IdO et le Cloud, ce qui présente un grand avantage, notamment en termes de latence pour les systèmes nécessitant un traitement en temps réel.

Malgré toutes ces différences, ces paradigmes se complètent. En effet, les tâches qui nécessitent des calculs intensifs et qui sont moins sensibles aux délais peuvent être exécutées soit dans les nœuds de Fog, soit dans les nœuds Edge. D'autre part, les tâches moins intensives en calcul et plus sensibles aux délais peuvent être exécutées dans les nœuds de Mist ou de Dew. Le traitement et la collecte des données sont toujours effectués et stockés dans le Cloud ou Cloudlet.

1.6 Conclusion

Dans ce chapitre, nous avons abordé presque tous les aspects de notre contexte de travail pour notre mémoire. Nous avons d'abord abordé le concept de l'IdO, sa définition, son architecture et les principales applications qui l'utilisent comme infrastructure dans la Section 1.2. Ensuite, dans les Sections 1.3 et 1.4, nous avons passé en revue les différents paradigmes modernes (CDEF computing) qui ont émergé pour soutenir le déploiement d'applications basées sur l'IdO. Nous terminons cette étude par une analyse comparative de tous les paradigmes examinés dans la Section 1.5.

Chapitre 2

Les Systèmes de Vidéo-surveillance Intelligents -SVSI-

2.1 Introduction

Aujourd'hui, les systèmes de vidéosurveillance sont devenus une partie essentielle de notre infrastructure. Ces systèmes jouent un rôle indispensable dans nos vies en raison de leurs énormes avantages, tels que la sécurité des lieux publics et privés et l'amélioration de la sécurité dans notre communauté. La vidéosurveillance consiste à placer des caméras dans un environnement, en gardant la trace des personnes observées et en détectant les comportements suspects par des opérateurs qualifiés. Au fur et à mesure que le nombre de caméras augmente, cette mission devient très complexe et parfois impossible. C'est pour cette raison qu'un aspect particulier de l'intelligence artificielle a vu le jour. Il s'agit du développement d'algorithmes de vision par ordinateur. Ces algorithmes permettent de traiter des données visuelles et de fournir des observations similaires à celles d'un être humain, rendant ce système plus intelligent.

Dans ce chapitre, nous présentons un état de l'art sur les systèmes de vidéosurveillance et leur évolution. Dans la section 2.2, nous introduirons quelques rappels sur le développement du système de vidéosurveillance. La section 2.3 présentera les différents composants de système de suivi en point de vue analytique. Ensuite, dans la section 2.4, nous décrirons la motivation pour utiliser l'IdOV comme infrastructure robuste de la prochaine génération du système de vidéosurveillance intelligent. Avant de conclure ce chapitre, nous présentons dans la section 2.6 les exigences et les défis auxquels est confrontée cette nouvelle génération.

2.2 Évolution des systèmes de vidéosurveillance

Les systèmes de vidéosurveillance sont utilisés depuis longtemps pour surveiller les zones de haute sécurité. Dans l'histoire de la vidéosurveillance, trois générations de systèmes, appelés 1GSS, 2GSS et 3GSS [43, 44]. Ils sont brièvement décrits ci-dessous.

2.2.1 La première génération (1GSS 1960-1980) : Tout analogique

Les systèmes de vidéosurveillance analogiques représentent la première génération de systèmes de surveillance. Ces systèmes étaient constitués de groupes de caméras reliées directement aux moniteurs via des câbles coaxiaux [45, 46]. Cette génération utilise des sous-systèmes analogiques pour l'acquisition, la transmission et le traitement des flux de données. Le processus de cette génération commence par la phase d'acquisition par les caméras. Les images numériques sont ensuite converties en un signal vidéo analogique, avec une certaine dégradation de l'image. Ce signal est ensuite transmis à tous les moniteurs et équipements d'enregistrement de la vidéosurveillance pour être visualisé et stocké [46].

2.2.2 La deuxième génération (2GSS 1980-2000) : Le système hybride

La conversion numérique-analogique provoque une certaine dégradation de l'image causée par la sensibilité du signal analogique au bruit. Il est possible d'avoir des systèmes numériques de vidéosurveillance en profitant du format numérique initial des images capturées et en utilisant des ordinateurs performants. L'amélioration technologique apportée par ces systèmes a conduit au développement de systèmes semi-automatiques, appelés systèmes de surveillance de deuxième génération [45]. Ces systèmes améliorent considérablement les applications de surveillance grâce à une certaine automatisation, comme la détection des mouvements et la minimisation de l'enregistrement des scènes. Les techniques de traitement d'image en général vont de la simple

détection des changements ou de l'élimination du bruit de l'image à des tâches plus complexes comme la reconnaissance et le suivi des objets [46].

2.2.3 La troisième génération (3GSS 2000-aujourd'hui) : Tout numérique

La troisième génération de systèmes de vidéosurveillance correspond aux systèmes qui gèrent un grand nombre de caméras avec des ressources distribuées et de nombreux points de contrôle, reflétant la nature hiérarchique et distribuée du processus de surveillance humaine. Cette génération repose essentiellement sur des capacités de traitement complètement autonome et distribuées sur le réseau et sur des dispositifs de traitement du signal embarqués afin de tirer parti de l'évolutivité et de la robustesse potentielles des systèmes distribués. Les principaux objectifs attendus de la troisième génération sont de fournir une bonne compréhension de la scène en temps réel et à faible coût, notamment dans les réseaux de surveillance à grande échelle [45].

Le tableau 2.1 présente un résumé de l'évolution de ces générations.

1 ^{re} génération	1960-1980
Techniques	Systèmes CCTV analogiques
Avantages	Donne de bonnes performances dans certaines situations Technologie avancée
Inconvénients	Utiliser des techniques analogiques pour la distribution et le stockage des images
Recherche en cours	Numérique ou analogique Enregistrement vidéo numérique Compression vidéo CCTV

2 ^{me} génération	1980-2000
Techniques	Surveillance visuelle automatisée en combinant la technologie de la vision par ordinateur avec les systèmes CCTV
Avantages	Augmenter l'efficacité de surveillance des systèmes CCTV
Inconvénients	Les algorithmes de détection et de suivi robustes nécessaires pour l'analyse du comportement
Recherche en cours	Algorithmes robustes de vision par ordinateur en temps réel Apprentissage automatique de la variabilité de la scène et des modèles de comportement Réduction de l'écart entre l'analyse statistique d'une scène et la production des interprétations en langage naturel

3 ^{me} génération	2000-aujourd'hui
Techniques	Système automatisé de surveillance à grande échelle
Avantages	Des informations plus précises grâce à la combinaison de plusieurs types de capteurs Distribution
Inconvénients	Distribution de l'information (intégration et communication) Méthodologie de la conception Plates-formes mobiles Plates-formes multi-capteurs
Recherche en cours	Intelligence distribuée ou centralisée Fusion de données Cadre de raisonnement probabiliste Techniques de surveillance multi-caméras

TAB. 2.1 : Résumé de l'évolution technique des systèmes de surveillance intelligents [45].

2.3 Description des techniques d'analytique vidéo dans les systèmes de surveillance

Avec la croissance énorme du nombre des caméras et le flux des vidéos collectée par ces derniers, le traitement manuel par les opérateurs humains devient très difficile. Pour cela, les systèmes de vidéo surveillance ont été apparus comme une solution pour un traitement automatique des flux des vidéos. Ces algorithmes nous permettent de traiter les données visuelles et de fournir des observations similaires à celles d'un être humain grâce à des méthodes d'analyse intelligentes, ou ce que nous appelons les techniques d'analyse vidéo. La Figure 2.1 montre la configuration standard des différents modules d'analyse vidéo. Chacune des sous-sections suivantes présente les techniques les plus courantes utilisées dans chacun de ces modules.

2.3.1 Détection des objets

La détection des objets en mouvement est l'étape la plus importante pour analyse de vidéo. Cette étape permet de détecter toute activité spécifique dans la zone de surveillance, comme le mouvement, l'apparition ou la disparition d'objets. Les techniques suivantes se focalisent essentiellement sur la détection des objets en mouvement [47].

Soustraction de l'arrière-plan

La soustraction de l'arrière-plan est une étape essentielle de tout traitement vidéo pour détecter les objets en mouvement [48]. Elle permet de classer les pixels en arrière-plan et en premier plan. Le processus de soustraction d'arrière-plan prend en entrée une image codée dans un espace couleur quelconque et produit à la fin une image en noir et blanc comme indiqué dans la Figure 2.2. La robustesse des systèmes de soustraction d'arrière-plan dépend de l'exploitation

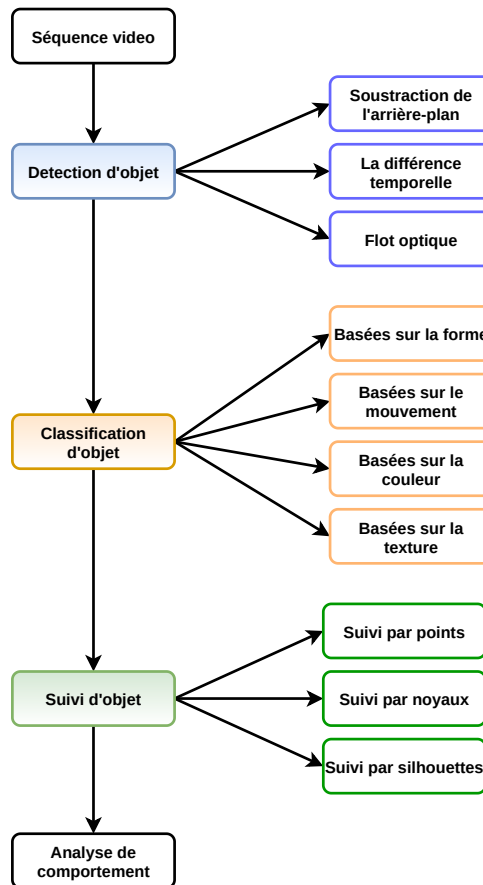


FIG. 2.1 : Étapes de base pour la détection et le suivi des objets [47]



FIG. 2.2 : Exemple de soustraction de l'arrière-plan

d'une méthode efficace qui assure une bonne séparation entre les pixels d'arrière-plan et d'avant-plan. Le principe de cette technique est de mesurer la différence entre l'image actuelle et l'image d'arrière-plan. Si la différence entre les pixels est supérieure à la valeur seuil (T), on considère qu'il s'agit des pixels de l'objet en mouvement, sinon, il s'agit des pixels de l'arrière-plan [49]. De nombreuses techniques de soustraction d'arrière-plan ont été proposées plus récemment pour améliorer la précision dans un environnement complexe [50, 51, 52].

Méthode de la différence temporelle

La méthode de différenciation temporelle identifie l'objet en mouvement basé sur le résultat de différence au niveau des pixels entre deux ou trois images successives (voir la Figure 2.3) [47]. Mathématiquement, la différence entre deux images consécutives est défini comme suit [53] :

$$I_{d(k,k+1)} = |I_{k+1} - I_k|$$

Où :

I_k : est supposé être la valeur de la k -ième image dans les séquences des images.

I_{k+1} : est la valeur de la $(k + 1)$ -ième image dans les séquences des images.

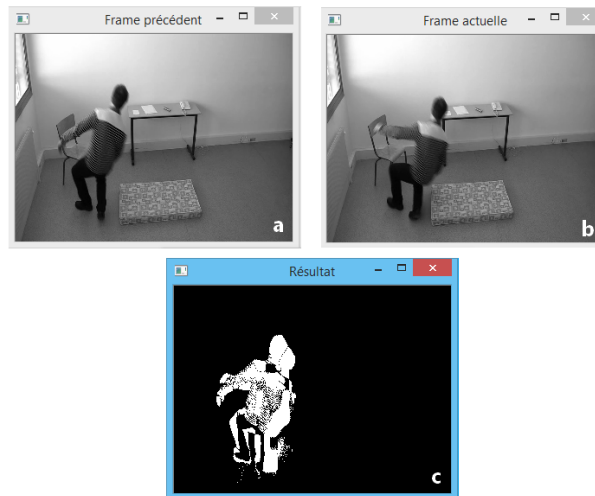


FIG. 2.3 : Exemple de détection d'objets à l'aide de la technique de différenciation temporelle

Le flot optique

Cette méthode basée sur un processus de regroupement en fonction de distribution de Flot optique de l'image [47]. Cette technique tente de calculer le mouvement entre deux images qui sont prises aux instants t et $t + \Delta t$. Cette méthode est dite différentielle, car elle est basée sur la dérivée partielle par rapport aux coordonnées spatiales et temporelles.

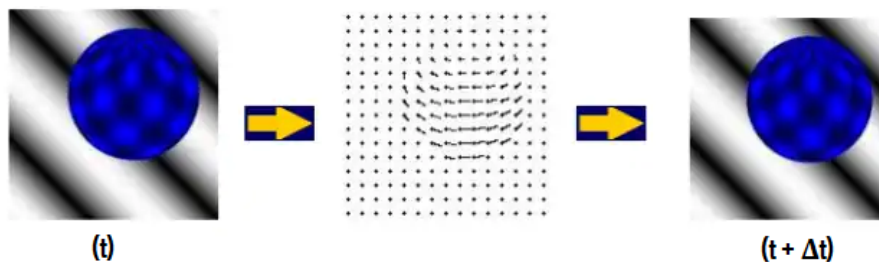


FIG. 2.4 : Exemple de détection d'objets à l'aide de la technique de flot optique

Le Tableau 2.2 présente une étude comparative entre les méthodes de détection des objets en mouvement exposées précédemment.

2.3.2 Classification des objets

Cette étape consiste à reconnaître la nature de l'objet en mouvement en fonction de leurs caractéristiques dans la région de mouvement [47] pour analyser et classer cet objet en différentes catégories prédéfinies (exemple : humains, véhicules, animaux, etc.). Ce classement peut s'effectuer préalablement au suivi pour ne conserver que les trajectoires des objets pertinents pour les fins de surveillance. Les approches des classifications sont généralement classées en fonction de la

Méthodes	Avantages	Inconvénients
Soustraction de l'arrière-plan	<ol style="list-style-type: none"> 1. Le moyen le plus simple 2. Performant pour l'arrière-plan statique 	<ol style="list-style-type: none"> 1. Il faut un arrière-plan sans objets en mouvement
Différence temporelle	<ol style="list-style-type: none"> 1. Faible besoin en mémoire 2. Il ne nécessite pas de sous-échantillonnage des images pour créer un modèle d'arrière-plan adéquat 	<ol style="list-style-type: none"> 1. Le calcul nécessite un tampon contenant les valeurs récentes des pixels 2. Elle ne prend pas en compte l'arrière-plan multimodal
Flot optique	<ol style="list-style-type: none"> 1. Il peut produire l'information complète sur le mouvement 	<ol style="list-style-type: none"> 1. Nécessite une grande quantité de calculs.

TAB. 2.2 : Étude comparative des méthodes de détection d'objet [47].

forme, du mouvement, de la couleur et de la texture [54]. Les sous-sections suivantes présentent brièvement ces approches.

Approches basées sur la forme

Cette approche est basée sur l'analyse de la forme des objets détectés. L'analyse de la forme est basée sur la géométrie des régions (tel que la boîte d'élimination, les contours externes, etc.), afin de découvrir des objets de formes semblables en les comparant au contenu d'une base de données de référence [55].

Approches basées sur le mouvement

Les systèmes de classification d'objets basés sur cette approche dépendent de la périodicité du mouvement (ils créent un système qui comprend comment les objets se déplacent), en prenant en compte les objets mobiles et non mobiles [55]. Le principal avantage de cette approche est la non-présence de forme de classification, mais une faible précision pour les objets non-mobiles [54].

Approches basées sur la couleur

Contrairement à d'autres approches, les couleurs sont faciles à acquérir et relativement stables sous des angles de vision changeants. Le faible coût de calcul de cette technique en fait une technique idéale qui peut être utilisée en cas de besoin. Afin de détecter et de suivre les véhicules en temps réel, des algorithmes basés sur des histogrammes de couleurs sont utilisés. Afin de segmenter l'arrière-plan et les images cibles et de décrire la distribution des couleurs dans la séquence d'images, une méthode de mélange gaussien est utilisée pour créer un modèle [47].

Approches basées sur la texture

L'objectif principal de cette approche est l'extraction des caractéristiques pertinente d'image texturée afin d'attribuer chaque image à une classe de texture prédéfinie [54]. Ces méthodes donnent une meilleure précision, mais avec un temps de calcul supplémentaire.

Le Tableau 2.3 présente une étude comparative entre les méthodes de classification des objets en mouvement discuté précédemment.

Méthodes	Précision	Observations
Basé sur le mouvement	Moyenne	Ne nécessite pas des modèles prédéfinis, mais a des difficultés à identifier une personne non mobile
Basé sur les textures	Haute	La qualité est améliorée, mais le temps de calcul est plus long
Basé sur la forme	Médium	Une approche simple de comparaison de motifs peut être appliquée avec des modèles appropriés
Basée sur la couleur	Haute	Il crée un modèle de mélange gaussien pour décrire la distribution des couleurs dans la séquence d'images et pour segmenter l'image en deux catégories : arrière-plan et objets

TAB. 2.3 : Étude comparative des méthodes de classification d'objet [47].

2.3.3 Suivi des objets

Le suivi des objets a pour but de localiser un objet dans une séquence vidéo, en fonction de son emplacement dans la première image. Les techniques de suivi des objets reposent principalement sur la représentation d'objets et sont regroupées en trois catégories : (i) suivi par points, (ii) suivi par noyaux et (iii) suivi par silhouettes [48].

Suivi par points

Dans cette méthode, le suivi des objets est effectué en fonction de la représentation des points caractéristiques des objets en mouvement essentiels. Cette méthode présente deux phases importantes : la prédiction qui consiste à utiliser toutes les observations actuelles et à les mettre à jour pour l'étape suivante. La deuxième phase est la correction qui fournit une meilleure estimation de l'étape suivante [47]. Ces méthodes souffrent d'un problème majeur, notamment dans le cas de la détection de faux objets et d'occlusions [56].

Suivi par noyaux

Le suivi de noyau est typiquement réalisé en calculant le mouvement de l'objet lorsqu'il se déplace d'une image à l'autre. Le mouvement de l'objet se présente généralement sous la forme d'un mouvement paramétrique (translation, conformation, affine, etc.) ou du champ de flux dense calculé dans les images suivantes. Les algorithmes de cette méthode sont classés selon plusieurs critères : nombre des objets suivis, la méthode utilisée pour estimer le mouvement de l'objet et la représentation d'apparence utilisée [57].

Suivi par silhouettes

Dans la plupart des cas, les objets n'ont pas de contours géométriques simples, comme un corps humain avec des formes géométriques complexes. L'objectif de ce mécanisme de suivi est de trouver la région de l'objet (tel que les mains, la tête, les épaules, etc.) dans chaque image à l'aide d'un modèle d'objet généré par les images précédentes [47]. Le principal défi de création

de cette méthode est les formes composites des objets qui ne sont pas facilement définies par des formes géométriques.

2.3.4 Analyse de comportement

L'analyse comportementale est la tâche de plus haut niveau qui consiste à interpréter le comportement et l'interaction des objets à partir de toute information collectée pendant le processus de suivi. Cette tâche nécessite une analyse sémantique, parfois complexe, en fonction du contexte de l'application et de l'événement détecté [54].

2.4 Nouvelle génération de système de vidéo-surveillance : Point de vue infrastructurelle

Au cours des dernières décennies, une grande quantité de données est générée par les réseaux de surveillance multi-caméras. Conformément au Cisco Visual Networking Index, le trafic vidéo IP atteindra 82% du trafic Internet grand public en 2020, alors qu'il était de 70% en 2015. Sur la base de cette évolution, Cisco suppose que le trafic vidéo IP dépassera 95% du trafic Internet d'ici à 2030 [58]. Cette évolution rapide des données générées par les caméras pose de nombreux défis importants aux systèmes de vidéosurveillance conventionnels (2.4.1). Face à cette évolution, une nouvelle génération de systèmes de vidéosurveillance appelée l'Internet des Objets Vidéo ou IdOV émerge pour renforcer la flexibilité et remédier aux défis des systèmes conventionnels.

2.4.1 Les limites des systèmes de vidéo-surveillance conventionnels

Les problèmes rencontrés par les systèmes de vidéo-surveillance conventionnels sont résumé ci-dessous [59, 60] :

Réseau

Du point de vue de la communication réseau, les systèmes classiques ne sont pas capables de répondre aux besoins de la prochaine génération de systèmes de vidéosurveillance, en raison des divers problèmes de latence, de sécurité et de traitement centralisé des données. Pour relever ces défis, l'intégration de nouvelles technologies de communication sans fil telles que la 5G et la 6G est l'une des solutions prometteuses pour améliorer les performances du haut débit et offrir de nouveaux niveaux de fiabilité, de latence, d'efficacité et de sécurité.

Architecture

L'architecture centralisée des systèmes de vidéo-surveillance conventionnels pose plusieurs problèmes tels que :

- Tous les traitements sont effectués dans la même unité de contrôle, ce qui peut entraîner une interruption des services en raison d'une connectivité intermittente ou de la défaillance complète de ce système centralisé.
- L'analyse des flux vidéo nécessite beaucoup de puissance de calcul.

- La transmission des flux de caméras à une seule station dans un réseau centralisé nécessite de la bande passante, surtout dans les réseaux comportant de nombreuses caméras. Elle provoque également des goulets d'étranglement qui réduisent les performances des systèmes, notamment en termes de traitement en temps réel.

Optimisation

Dans les systèmes de vidéo-surveillance conventionnels, les caméras sont connectées à un serveur centralisé via des passerelles pour la diffusion des données collectées et les analyser. Mais avec la croissance des flux vidéo et la quantité énorme des données stocker (pertinente ou non pertinente), il est nécessaire d'intégrer les calculs sur les capteurs visuels lui-même, grâce aux technologies récentes de semi-conducteurs pour améliorer la qualité de traitement et l'optimisation d'espace de stockage.

Temps réel

La grande complexité des tâches du système de vidéosurveillance intelligent (détection, reconnaissance, suivi et analyse) sur un serveur peut entraîner des retards dans le traitement des images ou des vidéos, ce qui signifie que le système n'est pas adapté au fonctionnement en temps réel. Par conséquent, il est nécessaire de garantir les performances du système en temps réel en assurant le traitement immédiat des actions ou des décisions prises par les entités du système.

2.4.2 Internet des Objets Vidéo -IdOV-

Le développement mondial de l'Internet des objets (IdO) au cours de la dernière décennie a conduit à la création de nombreuses nouvelles applications grâce à la mise en réseau de divers appareils et capteurs. Les capteurs visuels ont récemment connu un véritable succès dans les systèmes IdO, car ils peuvent fournir des informations plus riches et plus diversifiées. Le réseau de capteurs visuels à grande échelle basé sur l'IdO est appelé Internet des Objets Vidéo.

Définition d'Internet des Objets Vidéo -IdOV-

Selon la Figure 2.5, l'IdOV est une partie de l'IdO capable de traiter efficacement de grands volumes de données, comme des images et des vidéos. Par rapport aux systèmes classiques, la vidéosurveillance dans un cadre IdOV fournit plusieurs couches basées sur de nouveaux paradigmes informatiques tels que le Mist, le Fog et le cloud computing en tant qu'infrastructure de communication et de prise de décision en capturant et en analysant de riches informations contextuelles et comportementales. L'IdOV vise à développer un système de vidéosurveillance plus efficace, plus flexible et plus rentable, adapté aux nouvelles exigences des villes intelligentes en termes de sûreté et de sécurité des citoyens dans les lieux publics et privés.

Définition 2.1 *Internet des objets vidéo -IdOV" (en anglais : Internet of Video Things -IdOV-) est l'intégration des caméras intelligentes avec les paradigmes informatiques modernes de l'IdO tels que cloud, mist, fog computing, pour traiter efficacement de grands volumes de données, telles que des images et des vidéos.*

Définition 2.2 (Benrazek, [61]) *Un réseau de capteurs visuels ou de caméras intelligentes distribués, qui sont identifiés de manière unique sans ambiguïté, fonctionnant dans un environnement IdO. Ces caméras peuvent interagir et communiquer entre elles et/ou avec d'autres objets*

de l'IdO et humains en utilisant les technologies de l'information et de la communication (TIC) pour le traitement distribué, l'échange/partage de données et l'accroissement de l'autonomie du système.

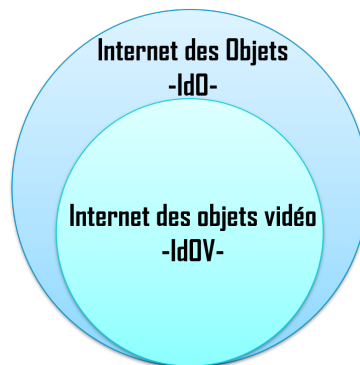


FIG. 2.5 : l'IdOV est une partie de l'IdO

En terme général, L'IdOV est un système distribué, capable de traiter efficacement de grands volumes de données visuelles collectées par des caméras distribuées et analysées par des ressources informatiques distribuées basées sur différents paradigmes informatiques modernes tels que Cloud, Mist, Fog computing.

Infrastructure de l'Internet des Objets Vidéo (IdOV)

À partir de la Section (2.4.2), nous pouvons définir l'infrastructure de l'Internet des Objets Vidéo (IdOV) comme suite :

- *Les caméras intelligentes* : Ces dispositifs intelligents se composent de caméra et de micro-contrôleur permettant d'acquérir, stocker, traiter des informations, et même communiquer entre eux et/ou avec d'autres objets de l'environnement IdOV.
- *Le réseau* : Les réseaux de communication qui connecte les caméras intelligentes entre eux ou avec d'autres et transmettre les données collectées à travers différentes technologies récentes (tel que BLE, Wifi, ZigBee, etc.).
- *Les applications* : La liaison entre l'interface de IdOV et les utilisateurs finaux (personnes ou systèmes), il s'agit d'une couche de traitement, stockage et analyse les grands volumes de donnée reçus des caméras intelligentes.
- *Paradigmes informatiques* : Les paradigmes modernes de IdO cité dans le chapitre 1 tel que le Cloud computing qui permet de stocker et à analyser de grosses quantités de données grâce à leur capacité notable de stockage et à leur puissance de calcul élevée. Le Fog computing qui gère et contrôle efficacement un ensemble de caméras intelligentes situées dans sa zone géographique.

L'infrastructure d'IdOV est illustrée dans la Figure 2.6.

2.5 Systèmes de vidéo-surveillance distribués à grande échelle

La croissance considérable du trafic visuel collecté par les caméras de vidéosurveillance nécessite de nouvelles solutions pour améliorer la capacité de stockage et assurer un traitement efficace

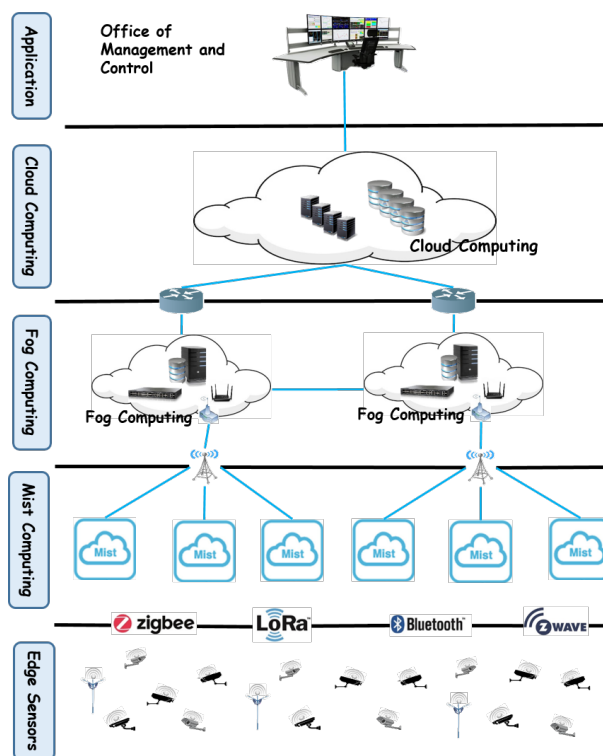


FIG. 2.6 : Infrastructure de l'IdOV [61]

et en temps réel. Après la défaillance du système centralisé, une nouvelle architecture distribuée de systèmes de vidéosurveillance a été récemment créée dans le cadre de la troisième génération de vidéosurveillance [62]. Selon Valera et Velastin [62], un système de vidéosurveillance distribué avec un stockage et une communication réseau distribués pourrait permettre une surveillance intelligente en temps réel, améliorer la fiabilité et la stabilité des systèmes de vidéosurveillance et répondre à divers besoins des systèmes centralisés. Pour cette raison, les systèmes de vidéosurveillance intelligents distribués sont de plus en plus courants dans les systèmes de surveillance actuels. L'idée de ces systèmes est de migrer le traitement et le stockage autant que possible à la périphérie du réseau par la distribution de caméras intelligentes qui ont des capacités de traitement et de stockage locales et des capacités de communication [63].

Dans les systèmes de vidéosurveillance intelligents distribués, chacune des caméras du réseau agit comme une entité autonome pouvant stocker et analyser les données localement. Cependant, le coût d'accès aux données dépend des distances physiques et topologiques entre les nœuds du réseau, ce qui est une propriété inhérente aux systèmes distribués en général. Ainsi, lors de l'exécution d'opérations de traitement dans un réseau de caméras intelligentes intégrées, nous pouvons déplacer les étapes de calcul vers la périphérie du réseau et laisser chaque nœud effectuer les actions requises. Ensuite, chaque capteur n'échange que les informations appropriées avec ses voisins pour finalement atteindre un objectif partagé et global de manière entièrement distribuée [63].

Le but de cette distribution est de répartir l'intelligence et, en particulier, de rapprocher une partie de l'intelligence du réseau périphérique (c'est-à-dire vers les caméras). Cette approximation est expliquée par les exigences illustrées à la Figure 2.7.

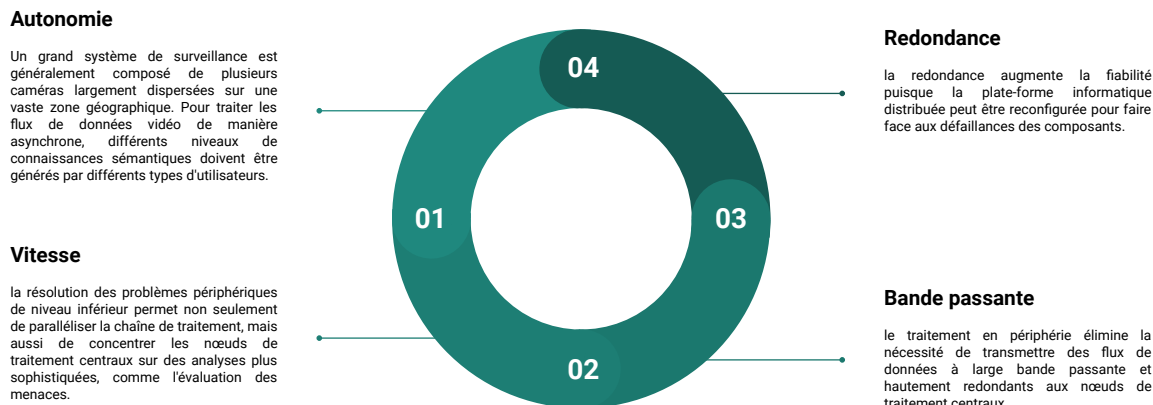


FIG. 2.7 : Les exigences relatives à la distribution [64]

2.6 Exigences et défis

L'Internet des objets vidéo a des caractéristiques uniques en matière de détection, de transmission, de stockage et d'analyse, et est fondamentalement différent de l'Internet des objets traditionnel. Ces nouvelles fonctionnalités de l'IdOV devraient poser des défis techniques importants que nous n'avons jamais connus auparavant, qui nécessitent des solutions techniques entièrement nouvelles. Dans cette section, nous aborderons les défis techniques suivants [60] :

- Intégrez le traitement visuel des données, le stockage, la mise en réseau et l'efficacité énergétique.
- La fiabilité et auto-adaptatives d'IdOV permettant de répondre à l'application gourmande en termes de bande passante des données des capteurs visuels.
- Intégrez la compression et l'analyse des données visuelles pour une recherche et une récupération efficaces pour assurer un système de recherche et de récupération efficace pour une large gamme d'applications IdOV.
- Des problèmes de sécurité et de confidentialité pour le partage des données de capteurs visuels et le traitement distribué dans le Cloud.
- Construction d'une plate-forme standardisée pour transmettre les données des capteurs visuelle sur le réseau d'Internet des objets à grande échelle afin de faciliter la coopération entre les différents propriétaires de données IdOV.

2.7 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art à propos le domaine d'application de notre système. Dans la Section 2.2 et la Section 2.3, nous avons abordé l'évolution des systèmes de vidéosurveillance et leurs composants d'un point de vue analytique, ainsi que dans la Section 2.4 nous avons abordé la prochaine génération des systèmes vidéosurveillance intelligents comme un système distribué capable de traiter les grands volumes des données visuelles. Et en termine ce chapitre avec les exigences et les défis de cette nouvelle génération de vidéosurveillance. Dans le chapitre qui se suit, nous avons présenté les différents concepts d'indexation arborescente.

Chapitre 3

Techniques d'indexation

3.1 Introduction

Selon les études actuelles [65, 2], l'utilisation des objets de l'IdO sera augmentée jour après jour et dépassera 1 trillion d'objets connectés en 2030. Cette croissance exponentielle des objets provoque notamment l'émergence de données massives. Malheureusement, la quantité massive de données générées par ces objets et leurs caractéristiques distinctives rendent l'utilisation des systèmes traditionnels de gestion de bases de données inadaptée à la gestion en temps réel de ces volumes de données, qui sont générés en continu et variables dans le temps [66]. C'est pourquoi l'IdO et notamment les systèmes de vidéosurveillance en tant que source importante de données nécessitent de nouvelles solutions pour les applications à grande échelle afin de permettre une recherche et une découverte rapides et efficaces de leurs données et services. Dans notre contexte, l'indexation et la recherche efficaces dans les collectes de données volumineuses constituent l'un des paradigmes les plus prometteurs pour résoudre ces problèmes [67]. L'indexation est une étape d'organisation des données qui doit permettre un accès efficace aux données. Les techniques de ce mécanisme visent à construire une structure de données qui organise les données [68], pour fournir un accès rapide aux objets d'une base de données en réduisant l'espace de recherche, le coût des entrées-sorties et le nombre de calculs de distances entre les objets. En d'autres termes, l'indexation fournit une mise en œuvre efficace de la recherche associative [69].

Dans le chapitre précédent, nous avons abordé notre domaine d'application. Ce chapitre se concentrera sur la gestion des données générées par l'IdO en général et les systèmes de vidéosurveillance en particulier. Dans la section 3.2, nous présentons brièvement les exigences d'indexation des données massives. Ensuite, nous nous concentrons sur la gestion et l'indexation des big data dans le domaine de l'IdO. Dans la section 3.3, nous présentons une étude analytique approfondie des techniques d'indexation existantes dans la littérature.

3.2 Exigences d'indexation des données massives

L'indexation des grands volumes des données de l'IdO, nécessite des mécanismes distribués, dynamique, robuste et évolutif qui peuvent fournir un accès plus rapide à l'information pour répondre aux exigences des utilisateurs. Pour résoudre ce problème, nous avons besoin d'un nouveau mécanisme qui nous permette d'organiser efficacement les données et de réduire autant que possible le temps de recherche pour améliorer le temps de recherche son influence sur le processus de recherche en temps réel comme dans notre problème d'étiquetage des objets dans les systèmes de suivi. Dans le cadre de nos travaux, les mécanismes d'indexation des données sont l'un des mécanismes les plus prometteurs pour résoudre ces problèmes. Ce mécanisme se concentre sur l'organisation de données massives grâce à des structures de données afin de fournir un accès rapide aux données en réduisant l'espace de recherche ainsi que le temps de recherche.

Afin d'évaluer l'efficacité et de déterminer l'applicabilité à grande échelle des structures d'indexation, certaines contraintes techniques sont préalablement imposées [65, 70]. Ces contraintes sont présentées à la Figure 3.1.

- **Évolutivité** : La structure de l'index doit être capable d'indexer des volumes de données en croissance continue sans perte de performance (efficacité à répondre aux requêtes).
- **Efficacité** : Contrairement aux structures conventionnelles d'accès physique aux entrées/sorties, qui tentent "seulement" de diminuer le nombre d'entrées/sorties sur disque, un index de recherche par similarité doit également prendre en compte les coûts de calcul,

car les calculs de similarité entre deux objets peuvent être très coûteux que les temps de lecture du disque.

- **Dynamisme** : La structure de l'index doit être dynamique par rapport aux modifications de la base de données.
- **Indépendance des données** : Les structures d'accès doivent être performantes quelle que soit la distribution des données, c'est-à-dire résoudre efficacement les problèmes liés à la "malédiction des dimensions" ou à l'inefficacité dans les grandes dimensions.

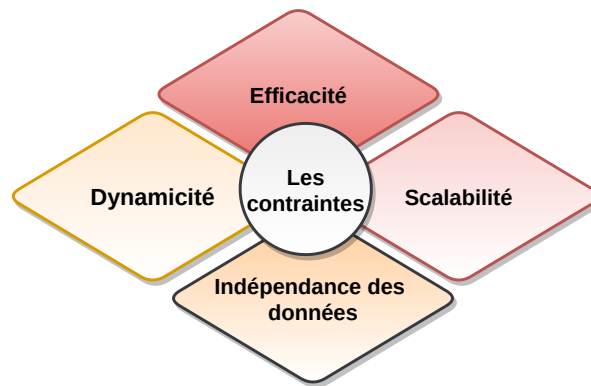


FIG. 3.1 : Exigences d'indexation des données massives [65]

3.3 Les structures d'indexation arborescentes récentes

Plusieurs structures d'indexation sont présentées dans la littérature pour améliorer la qualité et le temps de la recherche d'information dans un ensemble de données massives. D'après Kouahla dans [70], les structures d'indexation sont regroupées dans deux grandes catégories selon l'espace dans lequel elles sont créées : l'espace multidimensionnel et l'espace métrique.

3.3.1 Techniques d'indexation multidimensionnelle

Définition 3.1 (Kouahla, [71]) *Un espace multidimensionnel est défini lorsque les éléments de l'ensemble étudié sont des vecteurs, homogènes ou hétérogènes, et que leurs composantes sont totalement ordonnées.*

Les techniques d'indexation dans les espaces multidimensionnels peuvent être classées en trois grands types selon le type de structure utilisé : i) les techniques basées sur le hachage, ii) les techniques basées sur les arbres et iii) les techniques basées sur les bitmaps. Dans ce qui suit, nous allons présenter ces trois types de techniques d'indexation multidimensionnelle.

i) Techniques basées sur le hachage

Les méthodes d'indexation basées sur le hachage sont des techniques plus populaires dans le domaine de l'indexation de données multidimensionnelles. Ces techniques consistent à partitionner de manière répétitive l'ensemble des données et à dériver un seul "bit" de hachage de chaque partitionnement. Dans le hachage basé sur le partitionnement binaire, les données d'entrée sont

mis en correspondance avec un espace de code discret appelé espace de Hamming, où chaque échantillon est représenté par un code binaire [72].

En terme général, ces techniques permettent de transformer des données de taille quelconque en valeurs plus courtes de longueur fixe qui réduisent d'une manière considérable le temps d'exécution, la consommation de stockage et la complexité d'accès aux données [73]. De nombreuses méthodes basées sur la technique du hachage sont appliquées à plusieurs applications réelles, telles que la vision par ordinateur, la recherche et l'analyse d'informations. Selon Wu-Jun LI [74], ces méthodes divisées en deux familles principales : (1) *les méthodes indépendantes des données* et (2) *les méthodes dépendantes des données*. Si l'ensemble des fonctions de hachage est défini de manière unique et indépendante des données à traiter, nous pouvons le classer comme une méthode de hachage indépendante des données. Sinon, elle est classée comme une méthode de hachage dépendant des données [75].

- Méthodes indépendantes des données

Si l'ensemble des fonctions de hachage est défini de manière unique et indépendante des données à traiter, nous pouvons le classer comme une méthode de hachage indépendante des données. Parmi les méthodes de hachage indépendantes des données, la méthode LSH (Locality-Sensitive Hashing) [76] la plus connue dans la littérature. L'un des principaux critères de cette technique est la fonction de hachage qui renvoie avec de fortes probabilités, le même bit pour des points de données proches dans l'espace original [72]. Depuis la proposition de LSH, plusieurs variantes ont été proposées pour améliorer la méthode LSH comme : BayesLSH [77], Super-bit LSH [78], Non-metric LSH [79], Kernelized LSH [80] et Asymmetric LSH [81]. Cependant, ces techniques nécessitent généralement un code très long dans chaque table de hachage pour garantir une précision acceptable, ce qui augmente considérablement les besoins de stockage, notamment pour les applications à très grande échelle [75, 72]. En général, les méthodes de hachage indépendantes des données sont bien adaptées aux petites données, mais elles ne sont pas suffisantes pour traiter les grandes données.

- Méthodes dépendantes des données

Contrairement aux méthodes indépendantes des données, les méthodes de hachage dépendantes des données (également appelé apprentissage des méthodes de hachage) sont définies de manière unique uniquement pour un ensemble de données d'apprentissage donné. Les fonctions de hachage impliquent généralement des comparaisons de similarité avec certaines caractéristiques de l'ensemble de données d'apprentissage [75]. Les méthodes de référence de cette famille comprennent : Graph Hashing [82], Multiple Hashing [83], Fast Supervised Hashing [84], Fast Supervised Discrete Hashing [85], Semi-supervised Topology-Preserving Hashing [86], semi-supervised multi-view discrete hashing [87], Asymmetric Deep Supervised Hashing [88] and Hashing as Tie-Aware Learning to Rank [89].

L'objectif de ces méthodes est d'ajuster précisément la distribution des données dans l'espace des caractéristiques pour obtenir une meilleure précision tout en préservant autant que possible la localité [75]. Cependant, le principal inconvénient des techniques de cette famille réside dans la nécessité d'utiliser un grand nombre de tables de hachage et de longs mots de code, ce qui rend difficile l'obtention de performances satisfaisantes. Par conséquent, de nouvelles solutions sont nécessaires pour résoudre le problème de l'optimisation de l'apprentissage des fonctions de hachage et des codes de hachage.

Pour plus de détails sur l'étude de ces techniques d'indexation, nous vous recommandons de faire référence aux articles suivants [75, 90, 91, 92, 74].

ii) Techniques basées sur les arbres

Certains auteurs considèrent les techniques d'indexation multidimensionnelle comme des méthodes de classification non supervisée. Il est important de noter que dans une classification, les classes ne sont pas de même cardinal et que, dans une classification hiérarchique, toutes les classes feuilles ne sont pas situées à la même profondeur. Les techniques d'indexation peuvent être classées selon deux approches principales :

- **Partitionnement de l'espace :**

Cette catégorie des méthodes d'indexation multidimensionnelle basée sur le partitionnement de l'espace en sous-espaces dans toutes les intersections entre ces sous-espaces sont nulle. Plusieurs techniques d'indexation s'appuient sur ce principe tel que : l'arbre KD [93], l'arbre PH [94], et récemment l'arbre BB [95].

L'arbre KD proposé par Jon Louis Bentley en 1975 [93] basé sur le partitionnement spatial de l'espace à K dimensions pour structurer les données selon leur répartition dans l'espace. Cette structure est très utile pour de nombreuses applications, notamment pour accélérer la recherche de données dans un espace multidimensionnel, la recherche par intervalle ou la recherche des plus proches voisins [96]. Malgré les avantages offerts par cette structure, elle est confrontée au problème du coût de recherche le plus élevé dû à la nécessité de visiter toutes les régions voisines pour trouver des objets similaires.

L'arbre-PH ou l'arbre PATRICIA-hypercube, est une structure d'indexation et de stockage de données multidimensionnelle proposé par Zäschke et al. en 2014 [94]. Elle est basée sur la représentation binaire des données sous forme de chaîne de bits [97] et sur la structure Quadtree [98]. Celle-ci utilise un hypercube pour le partitionnement de l'espace dans toutes les dimensions à chaque nœud de l'arbre [99]. Ce partitionnement permet de naviguer plus efficacement vers le sous-nœud et de stocker les données plus efficacement par rapport aux arbres binaires [94]. Dans la littérature, plusieurs autres améliorations ont été proposées pour améliorer l'efficacité de la structure de l'arbre-PH [100, 101, 102].

Récemment, Stefan Sprenger et al. en 2019 proposent une nouvelle structure d'indexation multidimensionnelle appelée l'arbre-BB [95]. Cette structure combine l'arbre-KD et l'arbre-X [103]. L'arbre-BB se compose de deux éléments : Un arbre de recherche k -aire et un ensemble de seaux de bulles. Les nœuds internes de l'arbre de recherche divisent récursivement l'espace de données en k sous-ensembles disjoint comme l'arbre-KD et toutes les données sont conservées dans des seaux à bulles dans les nœuds feuilles de l'arbre avec une capacité limitée inspiré de l'arbre-X. BB-tree montre une meilleure efficacité pour les requêtes de plage, mais les requêtes de recherche de similarité knn ne sont pas prises en compte.

- **Partitionnement des données :**

Le concept principal de cette deuxième catégorie des méthodes d'indexation multidimensionnelle est la création des ensembles ou des clusters de données appelées "Forme englobante". Les techniques les plus connues de cette catégorie sont l'arbre-B [104] et ses variantes (telles que l'arbre-B⁺ [105]), l'arbre-R [106], l'arbre-X [103] et l'arbre-SR [103].

L'arbre-R [106] est une structure hiérarchique basée sur l'arbre-B. L'idée principale de cette structure est la génération des rectangles englobante minimale (REM) autour des objets. L'avantage principal de l'arbre-R repose sur la structure dynamique de l'arbre et le décapage hiérarchique, mais il souffre d'un problème de malédiction dimensionnelle et d'inefficacité dans les

grandes dimensions [71]. De plus, le chevauchement des boîtes englobantes dans le répertoire augmente avec la croissance de la dimension [103], ce qui influence négativement le processus de recherche, en particulier la recherche k-*nn*.

L'arbre-SR (pour sphère et rectangle) est une structure d'indexation proposée en 1997 pour éviter le problème de chevauchement entre les cellules [107]. Le principe de l'arbre SR basé sur l'utilisation des formes englobante provenant de l'intersection d'un hyper-rectangle et d'une hyper-sphère [107]. Malgré la solution apportée par cet arbre, il souffre aussi du problème de la complexité des formes englobantes qui augmente le coût des opérations d'insertion et des recherches.

L'arbre-X est une nouvelle structure d'indexation basée sur L'arbre-R développée pour éviter le chevauchement dans les grandes quantités de données ponctuelles et spatiales dans un espace à haute dimension [103]. Cet arbre utilise le concept de régions superposées et le concept de super-nœuds. Le principe de base l'arbre-X est de garder le répertoire hiérarchique le plus que possible, et en même temps d'éviter le problème de sur-partitionnement. L'arbre-X est une structure variable dont la taille et la complexité sont difficiles à calculer en raison de leur sensibilité à la taille et à la distribution des données et ses performances restent limitées à la dimension des données [108].

Techniques basées sur le bitmap

Les techniques d'indexations basées sur le bitmap a été largement utilisée dans diverses applications en raison de sa rapidité dans les opérations par bit. Par contre, cette méthode peut consommer de grandes quantités de mémoire [109]. Un index bitmap est composé d'un grand nombre de séquences de bits. L'unité d'opération utilisée dans les techniques basées sur le bitmap peut se présenter sous la forme d'un octet, d'un mot, d'un Dword ou d'un qword, afin de s'adapter aux différents processeurs [109]. Un index bitmap sur un attribut indexé consiste en un vecteur de bits (c'est-à-dire un bitmap) de sorte que la séquence de bits a un 1 en position i si le i^{ieme} élément de données satisfait à la propriété, et 0 sinon [110].

Les méthodes basées sur l'index bitmap sont des techniques très efficaces pour répondre à des requêtes complexes pour des systèmes en lecture seule et pour des données qui ne sont pas fréquemment mises à jour, mais elles sont moins efficaces dans d'autres cas. Ce problème est causé par le processus de compression où ce dernier est utilisé pour réduire l'espace de stockage, mais à chaque opération de mise à jour, il est nécessaire de décoder et d'encoder le bitmap, et cette opération est coûteuse [111]. En outre, les index Bitmap sont généralement destinés à l'optimisation de la recherche et de l'extraction de données à faible variabilité, à faible cardinalité et à petites valeurs distinctes (Famille, Humain, Vrai, Faux, etc.) et non à des données complexes pour lesquelles aucune propriété particulière liée à la nature des données ne peut être exploitée.

3.3.2 Techniques d'indexation métrique

Les structures d'indexation dans un espace multidimensionnel sont plus rigides en raison de leur forte dépendance par type ou, plus précisément, par leurs propriétés géométriques. Cette dépendance rend très difficile l'implémentation d'une structure flexible pour un système fiable. Par conséquent, la focalisation de l'indexation s'est partiellement déplacée des espaces multidimensionnels vers les espaces métriques. L'espace métrique a été proposé comme une abstraction universelle pour les grands volumes de données [112]. Parmi les principales caractéristiques de cet espace, il ne nécessite pas la structure intrinsèque des données, mais seulement une fonction de

distance, avec les propriétés de non-négativité, de symétrie et d'inégalité triangulaire des paires de points de données [113]. L'avantage le plus important de l'indexation dans l'espace métrique est que davantage de types de données peuvent être indexés, car l'approche est basée uniquement sur le calcul des distances entre les objets et non sur leur contenu [67]. En d'autres termes, l'espace métrique est plus flexible et plus simple que l'espace multidimensionnel, ce qui nous permet de créer une structure d'indexation capable de traiter n'importe quel type de données.

Définition 3.2 ([114]) *Un espace métrique est un couple (E, d) où E est un ensemble non vide et d est une distance sur E , c'est-à-dire une application $d : E \rightarrow \mathbb{R}_+$ qui vérifie les quatre propriétés suivantes :*

- *positivité* : $\forall x, y \in E, d(x, y) \geq 0$.
- *Symétrie* : $\forall x, y \in E, d(x, y) = d(y, x)$.
- *Identité* : $\forall x, y \in E, d(x, y) = 0 \Leftrightarrow x = y$.
- *Inégalité triangulaire* : $\forall x, y, z \in E, d(x, y) \leq d(x, z) + d(z, y)$.

Les techniques d'indexation dans les espaces métrique disparu la notion de l'axe utilisé dans les techniques basé sur les arbres dans les espaces multidimensionnels [71]. Deux types de ces techniques ont été développées. La première basée sur le partitionnement de l'espace et la deuxième basée sur le non-partitionnement de l'espace.

Techniques basées sur le partitionnement de l'espace

Dans la littérature, on trouve deux approches des techniques partitionnement de l'espace : une approche qui utilise des boules (ball partitioning), comme l'arbre-VP [115], l'arbre-mVP [116], etc. La deuxième approche utilise des hyper-plans (hyper-plane partitioning), tels que l'arbre-GH [117] et l'arbre-GNAT [118], etc.

L'arbre-VP (vantage point) est une structure d'indexation hiérarchique basée sur le partitionnement de l'espace à travers des boules. L'arbre-VP choisi d'une manière aléatoire un point d'observation appelée pivot (p), puis calcule la distance médiane d_m entre le pivot choisi et les points de l'espace étudié pour définir une boucle $B(p, d_m)$ qui va partitionner l'espace en deux espaces disjoint[71, 115].

L'arbre-mVP (multiple Vantage Points) est une évaluation de l'arbre-VP qui utilise plusieurs points d'observation (les pivots). L'arbre-mVP est similaire a l'arbre-VP dans le sens où les deux structures utilisent les distances relatives à partir d'un point d'observation pour partitionner l'espace étudié. Cette structure a été proposée comme une solution au problème lie à la réponse efficace aux requêtes basées sur la similarité pour les espaces métriques à haute dimension [116].

L'arbre GH (Generalised Hyper-plane) utilise le principe du partitionnement dans les espaces métriques à l'aide d'hyper-plans. L'arbre GH est défini par les deux points le plus éloignés $p1$ et $p2$ (les pivots) avec $p1$ différent de $p2$ et le reste des points sont partitionnées selon leur distance avec les deux pivots[117].

L'arbre-GNAT [118] est un arbre m-aire et une généralisation de l'arbre-GH qui utilise k pivots dans chaque nœud interne au lieu de deux. L'idée de l'arbre-GNAT repose sur le choix de k points de séparation (les pivots) à partir de l'ensemble que nous désirons indexer et le reste des données sont partitionnées selon la fonction de la distance la plus proche par rapport les pivots choisis [71, 118].

Techniques basées sur le non-partitionnement de l'espace

Le principe de base des techniques d'indexation basée sur le non-partitionnement de l'espace (le partitionnement des données) est l'utilisation d'une fonction de distance pour regrouper les données selon leurs proximités. Dans cette catégorie, nous citons l'arbre-M [119] et ses variantes telles que l'arbre-slim [120] et récemment l'arbre-BCCF [3].

L'arbre-M est un arbre équilibré, capable de traiter des fichiers de données dynamique sans réorganisations périodiques [119]. Elle est basée sur le regroupement des données dans des boules (ou hyper-sphères) à travers l'utilisation d'une fonction de distance d [71].

L'arbre-slim est un arbre équilibré et dynamique a été développé pour réduire le chevauchement entre les régions de chaque niveau. Les objets sont insérés dans un Slim-tree de la manière suivante : en commençant par le nœud racine, cet algorithme cherche à localiser un nœud qui couvre le nouvel objet. Si aucun ne se qualifie, il faut sélectionner le nœud dont le centre est le plus proche du nouvel objet. Si plusieurs nœuds sont qualifiés, l'algorithme ChooseSubtree est appliqué afin de sélectionner l'un d'entre eux. Ce processus est appliqué de manière récursive sur tous les niveaux de l'arbre [120].

Murgante et al. [121] ont proposé une nouvelle structure d'indexation métrique appelée arbre-M^X basée sur l'arbre-M originale. Cette structure adapte les super-nœuds inspirés de l'arbre-X qui évite la division insatisfaisante des nœuds. Ainsi, elle réduit le coût de calcul et l'étend complètement à l'espace métrique où la complexité temporelle est réduite sans réglage d'aucun paramètre.

L'arbre-BCCF une nouvelle structure d'indexation efficace pour indexer les données IdO massives. Cette structure est fondée sur le partitionnement récursif de l'espace à travers l'algorithme de regroupement k-means qui permet de séparer de manière efficace l'espace en sous-espaces qui ne se chevauchent pas dans le but d'améliorer la qualité des résultats des algorithmes de recherche et de découverte [3]. L'inconvénient de cette structure réside dans la complexité de leur construction en raison du coût de convergence de l'algorithme k-means.

En conclusion de cette étude, nous pouvons affirmer que la technique d'indexation universelle capable de gérer des ensembles de données arbitraires n'a pas encore été trouvée et sa réalisation, qui répond aux exigences de l'indexation des données massives mentionnées dans la section 3.2, est presque irréalisable.

Notre étude montre que les structures d'indexation basées sur des arbres ont gagné en popularité ces dernières années. Ces structures sont des structures dynamiques avec des changements de données. Ces structures ne nécessitent pas de ré-organisations périodiques de la structure qui exige des ressources plus coûteuses. Au contraire, elles assurent une indexation continue avec une complexité logarithmique de l'insertion et de la recherche. Sur la base de ses avantages, nous pensons que l'utilisation de la structure arborescente sera le meilleur choix pour notre index.

Pour avoir une compréhension plus claire sur les structures d'indexation arborescence, nous allons faire une analyse comparative entre ces techniques. Les approches multidimensionnelles sont résumées dans le tableau 3.1 tandis que les approches métriques sont résumées dans le tableau 3.2.

Proposition	Avantages	Inconvénients
Arbre-kD	Accélérer la recherche de données dans un espace multidimensionnel Accélérer la recherche par intervalle ou la recherche des plus proches voisins	Coût de recherche le plus élevé
Arbre-X	Évite le problème de sur-partitionnement Réduction du taux de chevauchement	Taille et complexité sont difficiles à calculer
Arbre-SR	Construction simple Réduction du taux de chevauchement	Complexité des formes Algorithmes de recherche et d'insertion coûteux
Arbre-R	Structure dynamique Décapage hiérarchique	Inefficacité dans les grande dimension Augmentation du taux de chevauchement avec la croissance de la dimension
Arbre-BB	Efficace pour les requêtes de plage	Ne sont pas prises en compte la recherche knn
Arbre-PH	Plus rapide et plus efficace en termes d'efficacité de l'espace, de performance des requêtes par rapport aux arbres binaires	Dégradation au passage à l'échelle

TAB. 3.1 : Tableau comparatif des approches multidimensionnelles

Proposition	Avantages	Inconvénients
Arbre-M	Structure dynamique Réduction des calculs de distances	Non adapté aux données fortement groupées Problème de chevauchement Dégradation au passage à l'échelle
Arbre-slim	Performante par rapport arbre-M Réduction du taux de chevauchement	La complexité globale de calcul
Arbre-GNAT	Affiner la recherche Pas de chevauchement	Forme compliquée à manipuler Structure statique
Arbre-GH	Partitionnement simple Structure dynamique Réduction du taux de chevauchement	Forme compliquée à manipuler Dégradation au passage à l'échelle

Arbre-mVP	Réduction du coût de la recherche Faible impact à grande échelle	Structure statique
Arbre-VP	Implémentation simple	Dégradation au passage à l'échelle
Arbre-BCCF	Éliminer le problème de chevauchement Améliore la qualité des processus de recherche	La complexité de construction
Arbre-M ^X	Réduction de coût de calcul Réduction de la complexité temporelle	Dégradation au passage à l'échelle

TAB. 3.2 : Tableau comparatif des approches métriques [71]

En se basant sur les études précédentes, nous pouvons introduire une simple taxonomie, comme il est illustré dans la Figure 3.2, qui représente les techniques d'indexation dans les deux espaces multidimensionnels et métriques.

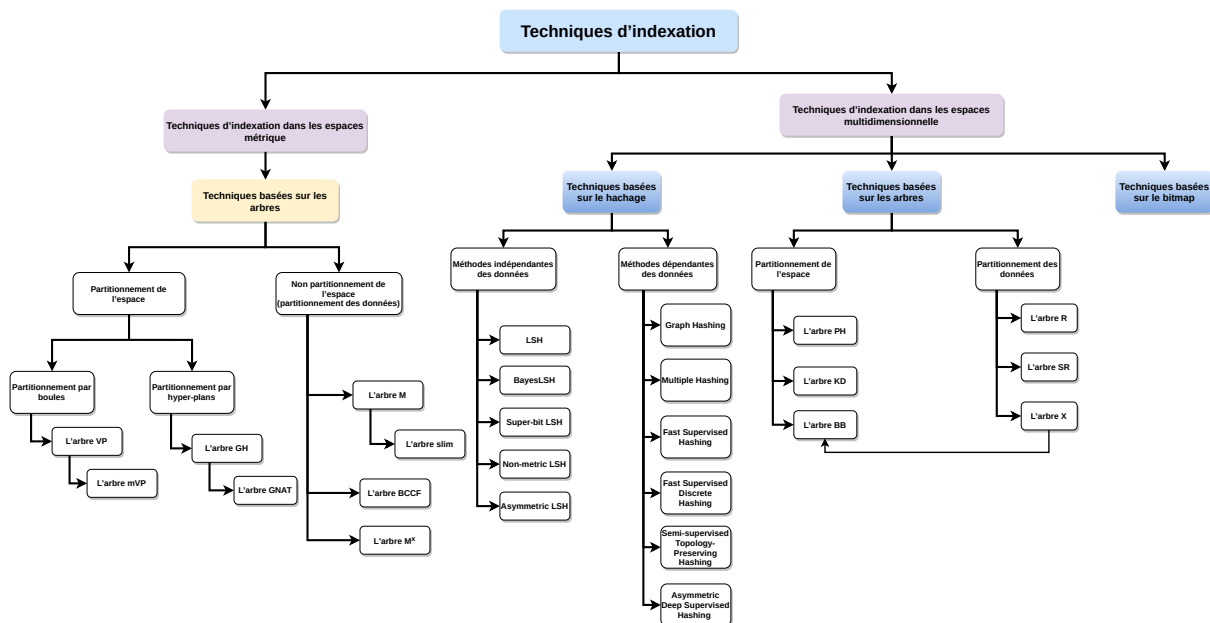


FIG. 3.2 : Taxonomie des techniques d'indexation

3.4 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art à propos les techniques d'indexation dans les deux espaces métriques et multidimensionnelles. Nous avons commencé par les exigences d'indexation des données massives dans la Section 3.2, puis dans la Section 3.3 nous avons présenté un résumé sur les différentes techniques d'indexation métrique et multidimensionnel. Nous terminons cette étude par une analyse comparative et une taxonomie des techniques d'indexation arborescente mentionnées dans le chapitre.

Chapitre 4

Conception

4.1 Introduction

Les systèmes de vidéosurveillance constituent désormais l'une des plus grandes sources de données visuelles dans le monde réel. La vidéosurveillance distribuée est devenue omniprésente en raison de la nécessité d'assurer la sécurité à grande échelle dans tous les domaines, notamment dans les villes intelligentes. Dans de nombreuses applications de vidéosurveillance, il est nécessaire d'attribuer un identifiant ou une étiquette à chaque objet suivi pour le distinguer des autres objets. Dans une telle mission, il est essentiel de conserver cet identifiant pour les mêmes objets, quelle que soit la zone, le moment de leur apparition ou la caméra de détection. L'augmentation de ces changements entraîne l'augmentation du problème de changement d'identifiant (ID-Sw) qui influence négativement la qualité du suivi.

Afin de pallier les problèmes décrits précédemment, nous allons présenter dans ce chapitre une nouvelle solution pour étiqueter automatiquement plusieurs objets pour un suivi efficace en temps réel en utilisant le mécanisme d'indexation.

4.2 Architecture du système proposé

L'objectif principal de notre système est d'améliorer la qualité du suivi d'objets en temps réel en utilisant des structures d'indexation basées sur des arbres et des caméras intelligentes, ainsi que les paradigmes informatiques modernes de l'IoT mentionnés dans le Chapitre 1. Cette combinaison représente la prochaine génération de systèmes de vidéosurveillance, appelée "Internet des objets vidéo ou IdOV"

Cette section présente l'architecture de notre système de vidéo-surveillance distribué basé sur les paradigmes de l'IdOV pour améliorer la qualité du suivi des objets en temps réel. Le système proposé est réparti sur quatre couches. Chaque couche a des tâches spécifiques parmi les tâches de suivi en temps réel d'objets en mouvement. Ces tâches sont attribuées en fonction des exigences de la tâche et des ressources disponibles dans la couche.

Le mécanisme de système proposé commence dans la première couche des caméras intelligentes. Ces caméras ont des capacités de traitement, de stockage et de communication grâce à l'intégration de technologies modernes de microcontrôleurs. Du point de vue architectural, ce type de caméra est divisé en deux couches : (i) la couche du capteur visuel et (ii) la couche de Mist computing. Le rôle de la couche de capteurs visuels est de capturer les événements se situant dans leur champ de vision. La principale tâche assignée à la couche de Mist computing est de détecter les objets présents dans les images entrantes. Ensuite, elle extrait les vecteurs de caractéristiques de chaque objet détecté et les envoie à la couche supérieure pour les identifier ou les étiqueter. Les données collectées et stockées sont ensuite transmises aux nœuds Fog correspondants, qui représentent la troisième couche. Chaque nœud Fog gère et contrôle un ensemble de caméras intelligentes situées dans sa zone géographique. Le rôle principal du Fog computing est l'indexation des données collectées par la couche inférieure dans la structure d'indexation proposée afin d'identifier (étiqueter) les objets demandés et de les partager avec d'autres caméras ou nœuds Fog voisins si nécessaire. Notre structure d'indexation a été créée dans cette couche pour concevoir un mécanisme d'indexation distribué en temps réel qui permet des recherches en temps réel. Concernant les données brutes indexées (images/vidéos), elles seront transmises à la couche Cloud. Le rôle de cette couche consiste à stocker et à analyser de grandes quantités de données grâce à leur grande capacité de stockage et à leur puissance de calcul élevée.

4.3 Aperçu du fonctionnement du système

Pour atteindre notre objectif, le système passe par quatre étapes principales :

1. Détection d'objet en mouvement et extraction de premier vecteur de caractéristique par les caméras intelligentes (les première et deuxième couches). Ce dernier envoie ensuite une demande d'identification au nœud Fog correspondant pour étiqueter cet objet sur la base du vecteur de caractéristiques extrait.
2. À la réception de cette demande, le nœud Fog lance une recherche dans ses données locales –structuré dans la structure proposée– pour trouver l'étiquette de l'objet en fonction de la métrique de similarité.
3. Une fois que la réponse d'étiquetage est reçue par les caméras, le processus de suivi est déclenché (au niveau de Mist computing).
4. À la fin du processus de suivi, chaque caméra envoie ses données extraites des objets suivis aux nœuds Fog pour les indexer.

Dans la suite, nous abordons chacune de ces étapes en détail.

4.3.1 Détection d'objet

La détection d'objets est l'une des tâches principales de notre système, qui consiste à déterminer l'emplacement sur l'image où certains objets sont présents, ainsi qu'à classer ces objets. Plusieurs approches ont été proposées dans la littérature, comme nous l'avons vu dans l'état de l'art. Parmi ces approches, nous nous intéressons à utiliser YOLO¹. Cette technique a gagné en popularité ces dernières années en raison de ses performances supérieures à celles des autres techniques de détection d'objets.

YOLO est une nouvelle approche de la détection d'objets en temps réel basée sur les réseaux neuronaux convolutionnels (CNN). En bref, cette approche utilise les caractéristiques de l'image entière et un seul réseau convolutif pour prédire simultanément plusieurs boîtes englobantes et les probabilités de classe pour ces boîtes. La première étape consiste à redimensionner l'image d'entrée à 448×448 , puis exécute un seul réseau convolutif sur l'image, et en enfin applique un seuil aux détections résultantes en fonction de la confiance du modèle. La Figure 4.1 présente un exemple de détection d'objet avec YOLO. Pour plus de détails sur cette technique, nous vous recommandons de consulter la référence [122].

Notre motivation pour utiliser cette technique repose sur [123] :

- *Rapidité* : cet algorithme améliore la vitesse de détection, car il peut prédire les objets en temps réel.
- *Haute précision* : YOLO est une technique de prédiction qui fournit des résultats précis avec des erreurs de fond minimales.
- *Capacités d'apprentissage* : L'algorithme possède d'excellentes capacités d'apprentissage qui lui permettent d'apprendre les représentations des objets et de les appliquer à la détection des objets.

¹YOLO est l'abréviation de l'expression "You Only Look Once"

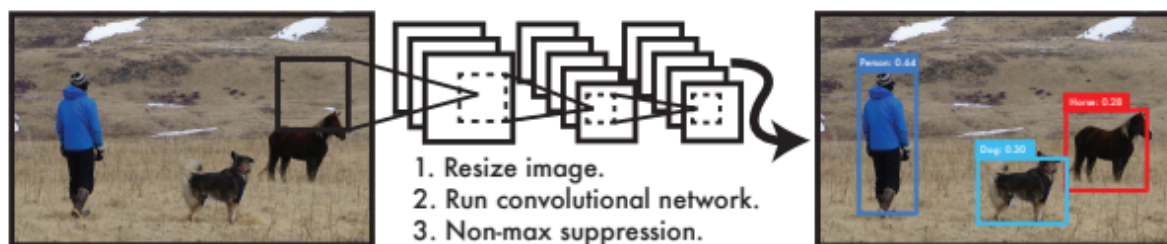


FIG. 4.1 : Le système de détection YOLO [122]

4.3.2 Extraction de premier vecteur de caractéristique

Pour l'extraction de vecteur de caractéristique, le système utilise une architecture CNN entraînée sur un ensemble de données de ré-identification de personnes à grande échelle [124] qui contient plus de 1 100 000 images de 1 261 piétons. L'architecture CNN se compose d'un réseau résiduel large avec deux couches conventionnelles suivies de six blocs résiduels. La carte globale des caractéristiques de dimension 128 est calculée dans la couche dense 10. Le réseau CNN prend en entrée une image couleur BGR et une matrice de boîtes de délimitation au format (x, y, w, h) et renvoie une matrice de vecteurs de caractéristiques correspondants [125].

4.3.3 Identification d'objet

Après la détection de l'objet et l'extraction de premier vecteur de caractéristique, on commence la première étape de cœur de notre travail, c'est l'identification ou l'étiquetage des objets détectés. Cette étape se fait avec une requête de recherche 1-*nn* (parfois appelée recherche exacte) en utilisant le premier vecteur de caractéristique dans la structure d'indexation créée avec notre technique proposée qui contient des objets étiquetés auparavant pour donner un identificateur unique pour chaque nouvel objet détecté. La description et l'algorithme de cette étape sont décrits dans la Section 4.4.2.

4.3.4 Suivi d'objet

Cette étape permet de suivre la trajectoire d'un objet détecté dans une période de temps bien définie en repérant leurs positions à chaque instant. Après l'identification de l'objet, ce processus sera déclenché afin de collecter un maximum d'informations sur l'objet concerné. Le modèle utilisé dans notre système pour le suivi est le filtre de Kalman standard avec un mouvement à vitesse constante et un modèle d'observation linéaire, dans lequel nous prenons les coordonnées de la boîte englobante (u, v, γ, h) comme des observations directes de l'état de l'objet. Le filtre de Kalman se compose de deux étapes [125, 126] :

1. l'étape de prédiction, qui permet de prédire grâce au modèle dynamique.
2. l'étape de correction, qui permet de corriger avec le modèle d'observation, de sorte que la covariance d'erreur de l'estimateur est minimisée.

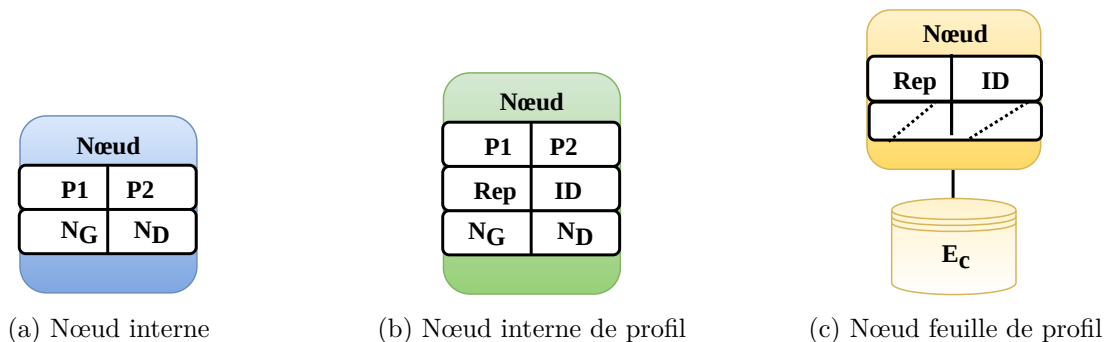


FIG. 4.2 : Les différents types de nœuds de l'arbre-UD

4.3.5 Insertion d'objet suivi

Cette étape consiste à insérer et indexer l'ensemble des métadonnées (les vecteurs de caractéristiques) collecté pendant le suivi des objets dans notre structure d'indexation arborescente. L'objectif de cette structure dans cette étape est de garder autant que possible la collection de métadonnées d'un même objet au même endroit et d'éviter leur dispersion pour améliorer le temps de recherche et la qualité de l'identification des objets en temps réel. La description et l'algorithme de l'étape d'insertion sont décrits dans la Section 4.4.3.

4.4 La structure d'indexation proposée

4.4.1 Définition de la structure proposée

L'approche proposée est appelée Arbre-UD (en anglais : Up Down Tree ou UD-tree). L'arbre-UD est une structure arborescente proposée pour indexer les données massives de l'IdOV inspirer de la méthode CAH [127](classification hiérarchique ascendance) et l'arbre BCCF [3]. Cette dernière est basée sur le regroupement des données les plus similaires – formant ce que l'on appelle un profil– d'une manière ascendante ou descendante. À chaque étape de processus de création de l'arbre-UD, les deux pivots sont présentés par les deux premiers vecteurs de caractéristique de chaque deux profils regrouper.

La construction de l'arbre-UD se fait au fur et à mesure du déroulement du système. Les deux étapes qui interviennent dans cette construction sont décrites ci-dessus. La première étape est l'identification des objets détectés et l'insertion des métadonnées collectées de l'objet identifié pendant leur suivi. Ces deux étapes sans décrits dans les Sections 4.4.2 et 4.4.3.

4.4.2 Identification des objets détecte

Comme nous l'avons vu précédemment, cette étape est effectuée au niveau de la couche Fog et est déclenchée une fois que la caméra détectrice envoie une requête d'identification d'objet après avoir extrait le vecteur de caractéristiques de l'objet détecté.

L'objectif principal de cette étape est de trouver l'identifiant ou l'étiquette de l'objet détecté s'il est déjà suivi précédemment dans le but de conserver le même identifiant autant que possible. Si l'objet est apparu pour la première fois, dans ce cas, cette étape assure la détermination d'un identifiant unique. Pour ce faire, le Fog lance une recherche exacte ou 1-*nn* sur son arbre-UD local. La description algorithmique de ce processus est représentée dans l'algorithme ??.

Durant cette étape, l'algorithme de recherche se fait en calculant la distance $d(o_q, p_i)$ entre le point de requête o_q (le premier vecteur caractéristique de l'objet détecté) et les deux pivots (p_1, p_2) de nœud actuel, tout en descendant dans l'arbre jusqu'à atteindre le nœud feuille approprié qui contient les métadonnées d'un objet identifié le plus similaire. Au niveau des feuilles une deuxième distance $d(o_q, rep_i)$ sera calculé entre le point de requête o_q et le représentant rep_i de nœud feuille qui a été atteint. La distance calculée sera comparé avec un seuil de similarité β défini auparavant pour déterminer si cet objet appartient ou non au même profil atteint.

Si la distance est inférieure au seuil $d(o_q, rep_i) \leq \beta$, nous considérons que l'objet en question appartient au même profil de nœud feuille atteint. Dans ce cas, le même identifiant de ce nœud feuille sera transmis au nœud de mist correspondant pour achever le suivi de l'objet. Si ce n'est pas le cas, c'est-à-dire si la distance est supérieure au seuil $d(o_q, rep_i) \geq \beta$, nous déterminons que cet objet n'est pas le même objet du profil atteint, mais qu'il est similaire. Dans ce cas, un nouvel identifiant unique a été généré pour cet objet et transmis aussi au nœud de mist correspondant, ensuite, le Fog essaie de réserver un nœud feuille de profil dans son arbre-UD pour cet objet au bon endroit en fonction d'un deuxième seuil de similarité T . Le but de ce deuxième seuil est de déterminer si cet objet doit être resté proche du nœud feuille de profil atteint ou non selon les deux stratégies d'insertion : insertion en bas (Down) ou insertion en haut (UP) présentées respectivement dans les algorithmes ?? et ??.

Insertion en bas (Down)

Dans le cas d'une forte similitude entre le nouvel objet et le profil atteint pendant la recherche ($d(N.rep, o_q) < T$), le processus d'insertion en bas sur cet objet est déclenché. Cette stratégie d'insertion consiste à regrouper le nouvel objet détecté avec le profil le plus proche de celui-ci. L'objectif est de garantir un bon positionnement des profils indexés, ainsi que de réduire autant que possible l'espace de recherche, et donc le temps de recherche. Le pseudo code de cette stratégie est présenté dans l'algorithme ??.

Le principe de cette stratégie est de créer un sous-arbre enraciné à un nouveau nœud interne (*internalnode*) pointant respectivement vers le nœud interne de profil de l'objet le plus similaire (N) à l'objet inséré et un nouveau nœud feuille de profil ($node_{(o_q)}$) qui correspond ou représente l'objet inséré. Les deux pivots de la racine de ce sous-arbre (*internalnode*) sont respectivement les représentants de nouvel objet (o_q) et le profil le plus similaire ($N.rep$).

Insertion en haut (UP)

Dans le cas d'une forte dis-similarité entre le nouvel objet et le profil atteint pendant la recherche ($d(N.rep, o_q) > T$), un regroupement de ce dernier avec la racine est appliqué (insertion en haut), ce qui permet de créer des groupes distincts de profils qui ne sont pas similaires suivant l'algorithme ??.

Comme le montre dans l'algorithme, une distance $d(rep_i, P_i)$ entre le représentant de nouvel profil et les deux pivots (p_1, p_2) de la racine de l'arbre-UD sera calculée. Si la distance $d(rep_i, p_1)$ est inférieure ou égale à la distance $d(rep_i, p_2)$ une nouvelle racine sera créée. Les deux pivots de la nouvelle racine (p_1, p_2) sont respectivement le représentant du nouveau profil (o_q) et le pivot gauche (p_1) de l'ancienne racine. Concernant les pointeurs de la nouvelle racine (N_G, N_D) , le pointeur de gauche (N_G) est pointé vers un nouveau nœud feuille de profil qui correspond à la nouvelle objet inséré ($node_{(o_q)}$), tandis que le pointeur de droite (N_D) est pointé vers l'ancienne racine (N).

Dans le cas contraire, c'est-à-dire la distance $d(rep_i, p_1)$ est supérieure à la distance $d(rep_i, p_2)$. Dans ce cas aussi une nouvelle racine sera créée de telle sorte que les deux pivots (p_1, p_2) de nouvelle racine sont le représentant du nouveau profil (o_q) et le pivot droit (p_2) de l'ancienne racine. Concernant les pointeurs de la nouvelle racine (N_G, N_D), c'est l'inverse du cas précédent où le pointeur de gauche (N_G) est pointé vers l'ancienne racine (N), tandis que le pointeur de droite (N_D), est pointé vers un nouveau nœud feuille de profil qui correspond à la nouvelle objet inséré ($node_{(o_q)}$).

4.4.3 Insertion des objets identifier

L'arbre-UD est un répertoire hiérarchique où l'insertion des objets se fait dans les deux sens de haut en bas et de bas en haut selon le critère de similarité comme nous l'avons vu dans la section précédente. La description du processus d'insertion dans l'arbre-UD des métadonnées collectées sur les objets pendant leur suivi (voir la Section 4.3.5) est représentée par l'algorithme ??.

Après que les objets ont quitté le champ de vision des caméras qui les suivent, Chaque caméra envoie les métadonnées extraites sur ces objets au nœud mist correspondant pour les indexer et les utiliser plus tard. À la réception de ces métadonnées au nœud mist, ce dernier lance le processus d'insertion décrit dans l'algorithme ??.

Selon l'algorithme, si la taille actuelle de conteneur de nœud feuille correspondant avec la taille de la nouvelle collection de données (sac) ne sont pas supérieures à la limite cardinale des conteneurs c_{max} , la nouvelle collection de métadonnées est insérée directement dans le même conteneur de nœud feuille. Sinon, le l'algorithme de partitionnement est déclenché sur ces données (c'est-à-dire, les métadonnées reçues et les métadonnées du nœud de feuille de profil correspondant). Après la convergence de ce dernier, deux clusters et deux centres sont obtenus. Les deux centres deviennent les représentants et les deux clusters deviennent les centenaires des deux nouveaux nœuds feuilles enracinés au nœud partitionné.

4.5 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle structure d'indexation distribuée sur une infrastructure IdOV pour organiser les données massive du système de vidéosurveillance afin d'améliorer le processus de suivi des objets en temps réel. Cette structure s'appelle Arbre-UD (en anglais : Up Down Tree ou UD-tree). L'arbre-UD est une structure arborescente inspirer de la méthode CAH (classification hiérarchique ascendance) et l'arbre BCCF. L'arbre-UD est distribué sur deux niveaux. Au niveau du Fog computing, qui contient les métadonnées ou les vecteurs de caractéristiques indexés dans des index locaux, et au niveau du Cloud, qui contient les données réelles des métadonnées indexées.

Conclusion et perspectives

Conclusion générale

Au cours de la dernière décennie, la quantité d'informations numérisées recueillies par les caméras de vidéosurveillance a augmenté de façon considérable. Cette évolution a conduit les chercheurs à concevoir de nouveaux systèmes pour répondre aux besoins actuels des systèmes de vidéosurveillance. Parmi ces besoins, en particulier :

- La nécessité de créer une plateforme flexible, robuste et auto-adaptable, capable de prendre en charge un réseau de vidéosurveillance à grande échelle en exploitant l'infrastructure moderne existante.
- Le besoin de nouvelles solutions pour augmenter la qualité du suivi et le fonctionnement du système de suivi en temps réel en raison de la complexité croissante des tâches du système, assurant le traitement immédiat des actions ou des décisions prises par les entités du système.

Le suivi des objets en mouvement est l'étape la plus importante des systèmes de vidéosurveillance intelligents. Il consiste à suivre la trajectoire des objets détectés dans la zone de surveillance grâce à la localisation de leur position à chaque instant. Pour accomplir ce processus, il est important de différencier les identités de chaque objet suivi afin de conserver le maximum d'informations sur celui-ci. Afin de réaliser cette tâche, une étiquette ou un identifiant unique est attribué à chaque objet détecté, indépendamment de la zone, du moment de leur apparition ou de la caméra de détection. Le changement de cette étiquette entraîne une augmentation du nombre de changements d'identité ($\#ID-Sw$). Cela conduit à une diminution de la qualité du suivi en raison de :

1. L'augmentation de la fragmentation de la trajectoire ($\#Frag$) des objets suivis.
2. La diminution du taux des objets qui ont été suivis pendant au moins 80% de leur durée de vie (les plus suivis -MT-).
3. L'augmentation du taux d'objets qui ont été suivis pendant au moins 20% de leur durée de vie (les objets les plus perdus -ML-).

Par conséquent, l'objectif principal de notre projet est de proposer un système d'identification ou d'étiquetage efficace et automatique dans des données augmentant avec le temps, le nombre de caméras et le nombre d'objets suivis, en utilisant des structures d'indexation arborescentes. Par conséquent, nous avons proposé une nouvelle structure d'indexation appelée Arbre-UD (en anglais : Up Down Tree ou UD-tree). Cette structure est principalement basée sur le principe de la classification hiérarchique ascendante (CHA) et de l'arbre-BCCF [3].

Initialement, notre système commence par l'étape de détection de l'objet en mouvement et l'extraction du premier vecteur de caractéristiques par la première et la deuxième couche qui représentent les caméras intelligentes. Ensuite, pour chaque objet détecté, nous exécutons une requête de recherche 1- nn au niveau de la couche Fog dans le but d'identifier l'objet selon la métrique de similarité utilisé et l'indexer dans l'arbre-UD local du nœud Fog correspondant. Après l'identification du nouvel objet, les caméras intelligentes commencent leur suivi pour recueillir le maximum d'informations sur celui-ci. L'ensemble des métadonnées collectées pendant le suivi de cet objet sera ensuite indexé au niveau du Fog computing à la fin du processus de suivi.

Les résultats de notre évaluation montrent que la structure réduit de manière significative le taux de changement des identifiants (**#ID-Sw**) des objets suivis. De même, le taux de fragmentation de la trajectoire (**#Frag**) est réduit. Par conséquent, la qualité du suivi est également améliorée. De manière générale, le DU-tree a un impact très positif sur l'amélioration de l'étiquetage automatique des objets en temps réel.

Les perspectives que nous pouvons dégager de ce travail sont :

- Tester notre solution sur un réseau réel multi-caméras basé sur l'architecture que nous proposons.
- Le choix du bon modèle d'extraction des caractéristiques qui influence de manière positive le choix du représentant pour chaque profil créé.
- Un seuil de similarité dynamique sur la base d'une phase d'apprentissage est une excellente solution pour rendre notre système complètement autonome.

Bibliographie

- [2] Anup MOHAN et al. “Internet of video things in 2030: A world with many cameras”. In : *2017 IEEE international symposium on circuits and systems (ISCAS)*. IEEE. 2017, p. 1-4.
- [3] Ala-Eddine BENRAZEK et al. “An efficient indexing for Internet of Things massive data based on cloud-fog computing”. In : *Transactions on emerging telecommunications technologies* 31.3 (2020), e3868.
- [4] Roberto MINERVA, Abyi BIRU et Domenico ROTONDI. “Towards a definition of the Internet of Things (IoT)”. In : *IEEE Internet Initiative* 1.1 (2015), p. 1-86.
- [7] Sarah A AL-QASEEMI et al. “IoT architecture challenges and issues : Lack of standardization”. In : *2016 Future technologies conference (FTC)*. IEEE. 2016, p. 731-738.
- [8] Miao YUN et Bu YUXIN. “Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid”. In : *2010 International Conference on Advances in Energy Engineering*. IEEE. 2010, p. 69-72.
- [9] Charith PERERA et al. “Context aware computing for the internet of things : A survey”. In : *IEEE communications surveys & tutorials* 16.1 (2013), p. 414-454.
- [10] Miao WU et al. “Research on the architecture of Internet of Things”. In : *2010 3rd international conference on advanced computer theory and engineering (ICACTE)*. T. 5. IEEE. 2010, p. V5-484.
- [12] Keyur K PATEL, Sunil M PATEL et al. “Internet of things-IOT : definition, characteristics, architecture, enabling technologies, application & future challenges”. In : *International journal of engineering science and computing* 6.5 (2016).
- [13] Carles GOMEZ et al. “Internet of Things for enabling smart environments : A technology-centric perspective”. In : *Journal of Ambient Intelligence and Smart Environments* 11.1 (2019), p. 23-43.
- [14] Imad SALEH. “Internet des Objets (IdO) : Concepts, enjeux, défis et perspectives”. In : *Revue Internet des objets* 2.10.21494 (2018).
- [15] Figen BALO et Belkız TORĞUL. “Internet of Things : A Survey”. In : *International Journal of Applied Mathematics, Electronics and Computers* (déc. 2016), p. 104-110. DOI : [10.18100/ijamec.267197](https://doi.org/10.18100/ijamec.267197).
- [16] Luigi ATZORI, Antonio IERA et Giacomo MORABITO. “The internet of things : A survey”. In : *Computer networks* 54.15 (2010), p. 2787-2805.
- [17] Peter MELL, Tim GRANCE et al. “The NIST definition of cloud computing”. In : (2011).
- [18] K. CHANDRASEKARAN. *Essentials of Cloud Computing*. CRC Press, déc. 2014.

- [19] Eric GRIFFITH et al. “What is cloud computing”. In : *Retrieved from PC Mag : <http://au.pcmag.com/networking-communications-software-products/29902/feature/what-is-cloud-computing>* (2016).
- [21] Srini SRINIVASAN. *Cloud computing basics*. Springer, 2014.
- [24] Keke GAI et Saier LI. “Towards Cloud Computing : A Literature Review on Cloud Computing and Its Development Trends”. In : *2012 Fourth International Conference on Multimedia Information Networking and Security*. 2012, p. 142-146. DOI : [10.1109/MINES.2012.240](https://doi.org/10.1109/MINES.2012.240).
- [25] S DURCEVIC. “Cloud computing risks & challenges businesses are facing in these days”. In : *Accessed : Sep 4* (2019), p. 10.
- [26] Mahadev SATYANARAYANAN et al. “The case for vm-based cloudlets in mobile computing”. In : *IEEE pervasive Computing* 8.4 (2009), p. 14-23.
- [27] Yi PAN, Parimala THULASIRAMAN et Yingwei WANG. “Overview of cloudlet, fog computing, edge computing, and dew computing”. In : *Proceedings of The 3rd International Workshop on Dew Computing*. 2018, p. 20-23.
- [28] Flavio BONOMI et al. “Fog computing and its role in the internet of things”. In : *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, p. 13-16.
- [29] Luis M VAQUERO et Luis RODERO-MERINO. “Finding your way in the fog : Towards a comprehensive definition of fog computing”. In : *ACM SIGCOMM Computer Communication Review* 44.5 (2014), p. 27-32.
- [30] Michael RABINOVICH, Zhen XIAO et Amit AGGARWAL. “Computing on the edge : A platform for replicating internet applications”. In : *Web content caching and distribution*. Springer, 2004, p. 57-77.
- [32] Weisong SHI et al. “Edge computing : Vision and challenges”. In : *IEEE internet of things journal* 3.5 (2016), p. 637-646.
- [33] Weisong SHI et Schahram DUSTDAR. “The promise of edge computing”. In : *Computer* 49.5 (2016), p. 78-81.
- [35] Yingwei WANG et al. “Dew computing and transition of internet computing paradigms”. In : *ZTE COMMUNICATIONS* 15.4 (2017).
- [36] Yingwei WANG. “Definition and categorization of dew computing”. In : *Open Journal of Cloud Computing (OJCC)* 3.1 (2016), p. 1-7.
- [38] Hamid Reza ARKIAN, Abolfazl DIYANAT et Atefe POURKHALILI. “MIST : Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications”. In : *Journal of Network and Computer Applications* 82 (2017), p. 152-165.
- [39] Yuezhi ZHOU, Di ZHANG et Naixue XIONG. “Post-cloud computing paradigms : a survey and comparison”. In : *Tsinghua Science and Technology* 22.6 (2017), p. 714-732.
- [40] Pengfei HU et al. “Survey on fog computing : architecture, key technologies, applications and open issues”. In : *Journal of network and computer applications* 98 (2017), p. 27-42.
- [41] G REKHA, Amit Kumar TYAGI et Nandula ANURADHA. “Integration of Fog Computing and Internet of Things : An Useful Overview”. In : *Proceedings of ICRIC 2019*. Springer, 2020, p. 91-102.
- [42] Yingwei WANG. “Cloud-dew architecture”. In : *International Journal of Cloud Computing* 4.3 (2015), p. 199-210.

- [43] Siyuan GAO. “An intelligent video surveillance system”. In : *2010 International Conference on E-Product E-Service and E-Entertainment*. IEEE. 2010, p. 1-4.
- [44] Brahim FAROU. “Multimédia mining Reconnaissance des formes dans une vidéo”. Thèse de doct. Université Badji Mokhtar-Annaba, 2016.
- [45] Sergio A VELASTIN et Paolo REMAGNINO. *Intelligent distributed video surveillance systems*. T. 5. IET, 2006.
- [46] Maria Valera ESPINA. “An approach for designing a real-time intelligent distributed surveillance system.” Thèse de doct. Citeseer, 2006.
- [47] SR BALAJI et S KARTHIKEYAN. “A survey on moving object tracking using image processing”. In : *2017 11th international conference on intelligent systems and control (ISCO)*. IEEE. 2017, p. 469-474.
- [48] A MANJU et P VALARMATHIE. “Video analytics for semantic substance extraction using OpenCV in python”. In : *Journal of Ambient Intelligence and Humanized Computing* 12.3 (2021), p. 4057-4066.
- [49] Ms Jyoti J JADHAV et J JYOTI. “Moving object detection and tracking for video surveillance”. In : *International journal of engineering research and general science* 2.4 (2014), p. 372-378.
- [50] Brahim FAROU et al. “Efficient local monitoring approach for the task of background subtraction”. In : *Engineering Applications of Artificial Intelligence* 64 (2017), p. 1-12.
- [51] Wafa NEBILI, Brahim FAROU et Hamid SERIDI. “Background subtraction using artificial immune recognition system and single gaussian (airs-sg)”. In : *Multimedia Tools and Applications* 79.35 (2020), p. 26099-26121.
- [52] WAFA NEBILI, Seridi HAMID et Kouahla Mohamed NADJIB. “A new process for selecting the best background representatives based on gmm”. In : *International Journal of Informatics and Applied Mathematics* 1.1 (2019), p. 35-46.
- [53] Nishu SINGLA. “Motion detection based on frame difference method”. In : *International Journal of Information & Computation Technology* 4.15 (2014), p. 1559-1565.
- [54] Amal BEN HAMIDA. “Vers une nouvelle architecture de vidéosurveillance basée sur la scalabilité orientée vers l’application”. Theses. Université de Bordeaux, oct. 2016.
- [55] Ali Fadhil YASEEN. “A Survey on the Tracking of Object in Sequence Image”. In : (2018).
- [56] Patrick SEBASTIAN, Yap Vooi VOON et Richard COMLEY. “Colour space effect on tracking in video surveillance”. In : *International Journal on Electrical Engineering and Informatics* 2.4 (2010), p. 298.
- [57] Alper YILMAZ, Omar JAVED et Mubarak SHAH. “Object tracking : A survey”. In : *Acm computing surveys (CSUR)* 38.4 (2006), 13-es.
- [58] Cisco Visual Networking INDEX. “Cisco visual networking index : Forecast and methodology 2015-2020”. In : *White paper, CISCO* (2015).
- [59] Aleksandar MILOSAVLJEVIĆ et al. “Integration of GIS and video surveillance”. In : *International Journal of Geographical Information Science* 30.10 (2016), p. 2089-2107.
- [60] Chang Wen CHEN. “Internet of Video Things : Next-Generation IoT With Visual Sensors”. In : *IEEE Internet of Things Journal* 7.8 (2020), p. 6676-6685.
- [62] Maria VALERA et Sergio A VELASTIN. “Intelligent distributed surveillance systems : a review”. In : *IEE Proceedings-Vision, Image and Signal Processing* 152.2 (2005), p. 192-204.

- [63] Niki MARTINEL. “On a Distributed Video Surveillance System to Track Persons in Camera Networks”. In : *ELCVIA Electronic Letters on Computer Vision and Image Analysis* 14.3 (2015), p. 39-41.
- [64] Paolo REMAGNINO, AI SHIHAB et Graeme A JONES. “Distributed intelligence for multi-camera visual surveillance”. In : *Pattern recognition* 37.4 (2004), p. 675-689.
- [65] Mohamed Amine FERRAG et al. “Big IoT Data Indexing : Architecture, Techniques and Open Research Challenges”. In : *2019 International Conference on Networking and Advanced Systems (ICNAS)*. IEEE. 2019, p. 1-6.
- [66] Mayur ARORA et al. “MavVStream : Expressing and Processing Situations on Videos Using the Stream Processing Paradigm”. Thèse de doct. 2018.
- [67] Pavel ZEZULA et al. *Similarity search : the metric space approach*. T. 32. Springer Science & Business Media, 2006.
- [68] Yihe DONG et al. “Learning sublinear-time indexing for nearest neighbor search”. In : (2019).
- [69] Volker GAEDE et Oliver GÜNTHER. “Multidimensional access methods”. In : *ACM Computing Surveys (CSUR)* 30.2 (1998), p. 170-231.
- [70] Zineddine KOUAHLA, Adeel ANJUM et Hamid SERIDI. “Indexing through separable partitioning for complex data sharing in P2P systems”. In : *Journal of Intelligent & Fuzzy Systems* 35.6 (2018), p. 6469-6478.
- [71] Zineddine KOUAHLA. “Indexation dans les espaces métriques Index arborescent et parallélisation”. Thèse de doct. Université de Nantes, 2013.
- [72] Jun WANG et al. “Learning to hash for indexing big data—A survey”. In : *Proceedings of the IEEE* 104.1 (2015), p. 34-57.
- [73] Mitali DESAI, Rupa G MEHTA et Dipti P RANA. “A survey on techniques for indexing and hashing in big data”. In : *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE. 2018, p. 1-6.
- [75] Lianhua CHI. “Hashing for large-scale structured data classification”. Thèse de doct. 2015.
- [76] Aristides GIONIS, Piotr INDYK, Rajeev MOTWANI et al. “Similarity search in high dimensions via hashing”. In : *Vldb*. T. 99. 6. 1999, p. 518-529.
- [77] Venu SATULURI et Srinivasan PARTHASARATHY. “Bayesian locality sensitive hashing for fast similarity search”. In : *arXiv preprint arXiv :1110.1328* (2011).
- [78] Jianqiu JI et al. “Super-bit locality-sensitive hashing”. In : *Advances in neural information processing systems*. 2012, p. 108-116.
- [79] Yadong MU et Shuicheng YAN. “Non-metric locality-sensitive hashing”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 24. 1. 2010.
- [80] Brian KULIS et Kristen GRAUMAN. “Kernelized locality-sensitive hashing for scalable image search.” In : *ICCV*. T. 9. 2009, p. 2130-2137.
- [81] Anshumali SHRIVASTAVA et Ping LI. “Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS)”. In : *Advances in Neural Information Processing Systems*. 2014, p. 2321-2329.
- [82] Qing-Yuan JIANG et Wu-Jun LI. “Scalable graph hashing with feature transformation”. In : *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.
- [83] Lei TIAN, Chunxiao FAN et Yue MING. “Learning spherical hashing based binary codes for face recognition”. In : *Multimedia Tools and Applications* 76.11 (2017), p. 13271-13299.

- [84] Guosheng LIN et al. “Fast supervised hashing with decision trees for high-dimensional data”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, p. 1963-1970.
- [85] Jie GUI et al. “Fast supervised discrete hashing”. In : *IEEE transactions on pattern analysis and machine intelligence* 40.2 (2017), p. 490-496.
- [86] Lei ZHANG et al. “Scalable similarity search with topology preserving hashing”. In : *IEEE Transactions on Image Processing* 23.7 (2014), p. 3025-3039.
- [87] Chenghao ZHANG et Wei-Shi ZHENG. “Semi-supervised multi-view discrete hashing for fast image search”. In : *IEEE Transactions on Image Processing* 26.6 (2017), p. 2604-2617.
- [88] Qing-Yuan JIANG et Wu-Jun LI. “Asymmetric Deep Supervised Hashing”. In : *arXiv e-prints*, arXiv :1707.08325 (juil. 2017), arXiv :1707.08325. arXiv : [1707.08325 \[cs.LG\]](https://arxiv.org/abs/1707.08325).
- [89] Kun HE et al. “Hashing as tie-aware learning to rank”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 4023-4032.
- [90] Zhongyu LI et al. “Large-scale retrieval for medical image analytics : A comprehensive review”. In : *Medical image analysis* 43 (2018), p. 66-84.
- [91] Tim KRASKA et al. “The case for learned index structures”. In : *Proceedings of the 2018 International Conference on Management of Data*. 2018, p. 489-504.
- [92] Foram S PATEL et Dipali KASAT. “Hashing based indexing techniques for content based image retrieval : A survey”. In : *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE. 2017, p. 279-283.
- [93] Jon Louis BENTLEY. “Multidimensional binary search trees in database applications”. In : *IEEE Transactions on Software Engineering* 4 (1979), p. 333-340.
- [94] Tilmann ZÄSCHKE, Christoph ZIMMERLI et Moira C NORRIE. “The PH-tree : a space-efficient storage structure and multi-dimensional index”. In : *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 2014, p. 397-408.
- [95] Stefan SPRENGER, Patrick SCHÄFER et Ulf LESER. “BB-Tree : A practical and efficient main-memory index structure for multidimensional workloads.” In : *EDBT*. 2019, p. 169-180.
- [96] Cédric FLEURY. “Le KD-Tree : une methode de subdivision spatiale”. In : *Universite de Rennes 1* (2008).
- [97] Ulrich GERMANN, Eric JOANIS et Samuel LARKIN. “Tightly packed tries : How to fit large models into memory, and make them load fast, too”. In : *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*. 2009, p. 31-39.
- [98] Raphael A FINKEL et Jon Louis BENTLEY. “Quad trees a data structure for retrieval on composite keys”. In : *Acta informatica* 4.1 (1974), p. 1-9.
- [99] Ahmed R MAHMOOD, Sri PUNNI et Walid G AREF. “Spatio-temporal access methods : a survey (2010-2017)”. In : *GeoInformatica* 23.1 (2019), p. 1-36.
- [100] Adrien FAVRE-BULLY. “Data preprocessing and other improvements for the multi-dimensional ph-index”. Mém. de mast. ETH Zurich, 2014.
- [102] Bogdan Aurel VANCEA. “Cluster-Computing and Parallelization for the Multi-Dimensional PH-Index”. Mém. de mast. ETH Zurich, 2015.
- [103] Stefan BERCHTOLD, Daniel A KEIM et Hans-Peter KRIEGEL. “The X-tree : An index structure for high-dimensional data”. In : *Very Large Data-Bases*. 1996, p. 28-39.

- [104] Venkathachary SRINIVASAN et Michael J CAREY. “Performance of B-tree concurrency control algorithms”. In : *Proceedings of the 1991 ACM SIGMOD international conference on Management of data*. 1991, p. 416-425.
- [105] Venkathachary SRINIVASAN et Michael J CAREY. “Performance of B+ tree concurrency control algorithms”. In : *The VLDB Journal 2.4* (1993), p. 361-406.
- [106] Antonin GUTTMAN. “R-trees : A dynamic index structure for spatial searching”. In : *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*. 1984, p. 47-57.
- [107] Norio KATAYAMA et Shin’ichi SATOH. “The SR-tree : An index structure for high-dimensional nearest neighbor queries”. In : *ACM Sigmod Record 26.2* (1997), p. 369-380.
- [108] Christian BÖHM, Stefan BERCHTOLD et Daniel A KEIM. “Searching in high-dimensional spaces : Index structures for improving the performance of multimedia databases”. In : *ACM Computing Surveys (CSUR) 33.3* (2001), p. 322-373.
- [109] Yinjun WU et al. “COMBAT : a new bitmap index coding algorithm for big data”. In : *Tsinghua Science and Technology 21.2* (2016), p. 136-145.
- [110] Chee-Yong CHAN et Yannis E IOANNIDIS. “Bitmap index design and evaluation”. In : *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. 1998, p. 355-366.
- [111] Manos ATHANASSOULIS, Zheng YAN et Stratos IDREOS. “Upbit : Scalable in-memory updatable bitmap indexing”. In : *Proceedings of the 2016 International Conference on Management of Data*. 2016, p. 1319-1332.
- [112] Rui MAO et al. “Overcoming the challenge of variety : big data abstraction, the next evolution of data management for AAL communication systems”. In : *IEEE Communications Magazine 53.1* (2015), p. 42-47.
- [113] Steven ROMAN, S AXLER et FW GEHRING. *Advanced linear algebra*. T. 3. Springer, 2005.
- [114] “Espace métrique”. In : juil. 2021.
- [115] Peter N YIANILOS. “Data structures and algorithms for nearest neighbor search in general metric spaces”. In : *Soda*. T. 93. 194. 1993, p. 311-21.
- [116] Tolga BOZKAYA et Meral OZSOYOGLU. “Distance-based indexing for high-dimensional metric spaces”. In : *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 1997, p. 357-368.
- [117] Jeffrey K UHLMANN. “Satisfying general proximity/similarity queries with metric trees”. In : *Information processing letters 40.4* (1991), p. 175-179.
- [118] Sergey BRIN. “Near neighbor search in large metric spaces”. In : (1995).
- [119] Paolo CIACCIA, Marco PATELLA et Pavel ZEZULA. “M-tree : An efficient access method for similarity search in metric spaces”. In : *Vldb*. T. 97. Citeseer. 1997, p. 426-435.
- [120] Caetano TRAINA et al. “Slim-trees : High performance metric trees minimizing overlap between nodes”. In : *International Conference on Extending Database Technology*. Springer. 2000, p. 51-65.
- [121] Shichao JIN, Okhee KIM et Wenya FENG. “MX-tree : A Double Hierarchical Metric Index with Overlap Reduction”. In : *International Conference on Computational Science and Its Applications*. Springer. 2013, p. 574-589.

- [122] Joseph REDMON et al. “You only look once : Unified, real-time object detection”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 779-788.
- [124] Liang ZHENG et al. “Mars : A video benchmark for large-scale person re-identification”. In : *European Conference on Computer Vision*. Springer. 2016, p. 868-884.
- [125] Nicolai WOJKE, Alex BEWLEY et Dietrich PAULUS. “Simple online and realtime tracking with a deep association metric”. In : *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, p. 3645-3649.
- [126] Nasser H ALI et Ghassan M HASSAN. “Kalman filter tracking”. In : *International Journal of Computer Applications* 89.9 (2014).
- [127] M JAMBU. “Sur l’utilisation conjointe d’une classification hiérarchique et de l’analyse factorielle en composantes principales”. In : *Revue de Statistique Appliquée* 25.4 (1977), p. 5-35.

Webographie

- [1] L'Opinion LIZA LIN ET NEWLEY PURNELL. *En 2021, le monde comptera 1 milliard de caméras de surveillance au coin de la rue*. Déc. 2019. URL : <https://www.lopinion.fr/edition/wsaj/en-2021-monde-comptera-1-milliard-cameras-surveillance-coin-rue-205359>.
- [5] ORACLE. *What is the Internet of Things (IoT)?* Oracle, 2020. URL : <https://www.oracle.com/internet-of-things/what-is-iot>.
- [6] ITU-T. *Overview of the Internet of things*. International Telecommunication Union, 2012. URL : <https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060> (visité le 07/05/2021).
- [11] Adam CALIHMAN. *IoT Architectures - Common Approaches and Ways to Design IoT at Scale*. NetBurner, 2019. URL : <https://www.netburner.com/learn/architectural-frameworks-in-the-iot-civilization/> (visité le 07/05/2021).
- [20] *What is Containers as a service (CaaS)?* www.ibm.com. URL : <https://www.ibm.com/services/cloud/containers-as-a-service> (visité le 17/05/2021).
- [22] Neha T. *What are Principles of Cloud Computing?* Binary Terms, jan. 2021. URL : <https://binaryterms.com/principles-of-cloud-computing.html#FourCloudDeploymentModel> (visité le 17/05/2021).
- [23] Emilie RAVET. *Qui sont les principaux fournisseurs de cloud public?* www.scalair.fr, oct. 2017. URL : <https://www.scalair.fr/blog/quels-sont-les-principaux-fournisseurs-de-cloud> (visité le 17/05/2021).
- [31] Pedro GARCIA LOPEZ et al. *Edge-centric computing : Vision and challenges*. 2015.
- [34] Victor BAHL et al. *Intelligent Edge*. Microsoft Research, août 2018. URL : <https://www.microsoft.com/en-us/research/video/intelligent-edge/> (visité le 10/05/2021).
- [37] Raka MAHESA. *How cloud, fog, and mist computing can work together*. IBM Developer, mar. 2018. URL : <https://developer.ibm.com/technologies/iot/articles/how-cloud-fog-and-mist-computing-can-work-together> (visité le 09/05/2021).
- [61] Ala-Eddine BENRAZEK. *Internet of Video Things (IoVT) : Next generation of video surveillance systems*. Déc. 2020. URL : <https://benrazekalaeddine.medium.com/itecturewhat-is-internet-of-video-things-iovt-1f2323e02a0b>.
- [74] LI WU-JUN. *Learning to Hash Paper, Code and Dataset*. 2015. URL : <https://cs.nju.edu.cn/lwj/L2H.html>.
- [101] Tilmann ZÄSCHKE. *The PH-Tree Revisited*. 2015.
- [123] Karimi GRACE. *Introduction to YOLO Algorithm for Object Detection*. section.io, 2021. URL : <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection>.