



Full Length Research Paper

Optimisation du stockage des données dans un système de vidéosurveillance par l'utilisation d'un algorithme prédictif de chemin par l'apprentissage

Papa Samour DIOP^{1*}, Ahmath Bamba MBACKE¹, Claude LISHOU¹, Gervais MENDY¹¹ École Supérieure Polytechnique de Dakar (ESP/UCAD), Dakar, Sénégal

Received October 2023 – Accepted December 2023

*Corresponding author. papasamour.diop@esp.sn

Author(s) agree that this article remain permanently open access under the terms of the Creative Commons Attribution License 4.0 International License.

Résumé :

Le nombre de systèmes de vidéosurveillance a augmenté rapidement au cours de la dernière décennie. Les systèmes de vidéosurveillance étant de grands consommateurs de volume de données de stockages, le problème de l'optimisation du stockage dans les systèmes de vidéosurveillance est d'actualité. Dans cette étude, nous analysons à l'aide de modèles mathématiques la consommation du volume de stockage d'une architecture qui met en œuvre un algorithme de prédiction de parcours par apprentissage qui prédit le parcours et la destination d'un mobile dans un réseau de vidéosurveillance afin de réduire l'utilisation de ces ressources. Cette méthode réduit considérablement le volume de données du trafic en temps réel. Un système expérimental est conçu pour évaluer la méthode et des expériences sont menées pour en démontrer la validité. Les résultats expérimentaux montrent que la méthode a non seulement amélioré de manière significative l'utilisation des ressources et réduit le volume de stockage.

Mots clés/Keyword : Transmission de données ; mobile ; surveillance ; prédiction ; apprentissage distribué ville intelligente ; système de vidéosurveillance ; stockage de données.

Cite this article:

Papa Samour DIOP, Ahmath Bamba MBACKE, Claude LISHOU, Gervais MENDY. (2023). Optimisation du stockage des données dans un système de vidéosurveillance par l'utilisation d'un algorithme prédictif de chemin par l'apprentissage. *Revue RAMReS – Sci. Appl. & de l'Ing.*, Vol. 5(1), pp. 130-138. ISSN 2630-1164.

1. Introduction

Ces dernières années, le marché de la vidéosurveillance s'est considérablement développé en raison de la nécessité croissante de sécurité publique. Un réseau de capteurs multimédias sans fil (WMSN) se compose essentiellement de quelques nœuds de caméra et de nombreux nœuds classiques déployés dans des zones sensibles pour des applications de surveillance telles que la santé, la sécurité à domicile, la détection d'intrus, la surveillance du trafic, etc. Les objectifs les plus importants des villes intelligentes sont de fournir une sécurité automatisée et améliorée avec un minimum de frais généraux. Toute ville sécurisée doit être équipée de milliers, voire de millions, de caméras de surveillance intelligentes. Avec le développement de la technologie de traitement vidéo numérique, la vidéosurveillance joue un rôle important dans la sécurité intelligente.

Les vidéos de surveillance doivent généralement être stockées pour analyse ou transmises aux opérateurs sur un site distant, où la surveillance des événements est effectuée. Le problème de la surveillance visuelle à distance a donc fait l'objet d'une grande attention ces derniers temps. Les principaux défis de la surveillance visuelle à distance sont l'optimisation du débit et du stockage de la vidéo capturée par les caméras de surveillance.

L'application de vidéosurveillance exige que la vidéo capturée soit transmise de manière efficace et sécurisée au site de surveillance. Pour les applications de sécurité à domicile et de détection d'intrusion, la caméra est déclenchée chaque fois que le mouvement est détecté et que la vidéo capturée est transmise.

2. État de l'art sur l'optimisation du stockage des données

Avec le développement récent de la technologie de réseau, la surveillance du trafic dans les villes intelligentes fait de plus en plus l'objet d'étude et de recherche. Actuellement, de nombreux travaux de recherche ont été consacrés à l'architecture de transmission de systèmes de surveillance en temps réel et de haute qualité. La technologie de transmission adaptative de vidéos mobiles peut réduire les débits vidéo au prix d'une perte de qualité acceptable. En outre, le schéma adaptatif réduit le délai de démarrage au stade initial et le délai de transmission pendant toute la lecture. Les deux peuvent améliorer les expériences des utilisateurs. L'un des principaux problèmes pour améliorer l'expérience des utilisateurs est la stratégie permettant de distribuer équitablement les données vidéo aux clients et de sélectionner le périphérique mobile le plus approprié pour transmettre les données en coopération. Récemment, des études ont porté sur l'intégration du stockage collaboratif avec la transmission coopérative. Pour minimiser les coûts totaux du système, les chercheurs [1,2] ont examiné la fréquence de distribution en fonction des caractéristiques des vidéos sur mobile.

Dans les systèmes de surveillance existants, la vidéo capturée par CCTV est généralement codée à l'aide d'une technologie de compression classique, telle que MPEG-1/2 ou H.264 / AVC. Ces systèmes encodent le signal vidéo indépendamment de l'importance de la vidéo, comme le déterminent les modules d'analyse de contenu vidéo (VCA) utilisés pour la reconnaissance automatique, la détection, le suivi, etc. La raison en est évidente : lors de l'utilisation d'un codeur conventionnel, il est presque impossible d'adapter le flux compressé aux propriétés de la scène en termes d'activité et de degré de signification de l'événement. Ainsi, la sortie du VCA ne peut pas être utilisée directement après ou pendant le codage et la transmission du signal, à moins d'effectuer un processus de transcodage coûteux et parfois irréalisable. Par exemple, dans de nombreuses situations de surveillance, la scène reste essentiellement statique pendant quelques minutes et même quelques heures dans certains cas. Au cours de ces périodes, rien d'intéressant ne se produit du point de vue de la surveillance, et la vidéo ressemble à une image fixe pendant de longues périodes sans autre activité que le mouvement aléatoire de l'environnement. C'est le cas de la surveillance dans le trafic où une route peut ne pas être fréquentée par une voiture pour une durée longue où les événements habituels sont les voitures qui vont et viennent de temps en temps. Ces événements, pertinents pour les applications de surveillance, sont détectables par le VCA agissant sur le signal de sortie des caméras.

Plusieurs recherches ont été faites dans le passé pour optimiser le débit de la vidéo de surveillance codée en tenant compte de l'importance du contenu dans les applications de surveillance. Dans [1], seules les régions spatiales de la vidéo identifiées comme étant au premier plan ont été codées à l'aide du codage à base

d'objet MPEG-4. L'arrière-plan a été codé en tant qu'image statique (ou actualisé périodiquement) et répété tout au long de la séquence. Ici, les zones de premier plan sont obtenues par des algorithmes de soustraction d'arrière-plan et affinées en analysant leur mouvement. Dans [2] cette approche a été encore étendue en dérivant le modèle d'efficacité de compression qui considère le nombre et la taille des objets de premier plan. Ce travail a été motivé par l'observation selon laquelle, si la taille et le nombre d'objets de premier plan sont élevés, le codage basé sur les objets peut produire des résultats pires que le codage conventionnel basé sur des images. Par conséquent, en utilisant le modèle de compression dérivé, le codeur peut choisir de manière adaptative d'effectuer un codage basé sur une image ou sur un objet pour un segment temporel spécifique d'une vidéo de surveillance. Dans [3,4], une approche similaire a été proposée, qui combine la soustraction de fond et un cadre de codage à base d'objet développé par l'utilisateur.

Dans [5], il est proposé un modèle qui permet de réduire le débit et le stockage en séparant le traitement des scènes selon l'événement. À cette fin [5], les auteurs classent les segments temporels de la vidéo de surveillance en deux types :

Segments temporels représentant une scène essentiellement statique (par exemple, seul le mouvement environnemental aléatoire est présent, arbres oscillants, drapeaux se déplaçant sur le vent, etc.) et segments temporels contenant une activité de mouvement non randomisée (par exemple, un véhicule se déplace sur une route). Les segments identifiés par le sous-fonds comme étant « non pertinentes du point de vue de la surveillance » sont codés en basse résolution, tandis que les segments pertinents sont codés en haute résolution. Cela réduit considérablement les exigences de stockage requises pour la vidéo de surveillance.

Partant de nos observations dans nos milieux de vie, nous avons constaté que nombreux utilisateurs et administrateur réseau envoient les images en ligne sans n'en savoir aucune stratégie sur l'optimisation du chemin de transmission dans les systèmes de vidéosurveillance.

L'extraction d'images clés et le résumé vidéo sont des approches courantes d'optimisation du volume de stockage dans les systèmes de vidéosurveillance. La structure de résumé est apprise par un algorithme d'apprentissage automatique, qui est utilisé plus tard pour classer uniquement les parties de la vidéo pouvant servir de représentation efficace de la vidéo originale. La détection d'événements intéressants dans les vidéos de surveillance est plus courante en raison des progrès de l'intelligence artificielle et des techniques d'apprentissage automatique. Les auteurs de [7,8,9] ont présenté aussi une technique de synthèse basée sur la classification hiérarchique. Dans cette technique, les plans présentant des caractéristiques similaires et étroitement liées dans le domaine temporel sont combinés. Les auteurs utilisent le descripteur visuel MPEG-7 pour choisir des index et générer un résumé. Le résumé obtenu contient les images clés et l'aperçu

de la vidéo d'origine, auxquels vous pouvez accéder de manière non linéaire.

Dans [6,10], les auteurs proposent l'extraction de cibles mobiles et l'algorithme de reconstruction vidéo rapide conformément au principe visuel. La stratégie de sélection de l'échantillon de l'algorithme VIBE est améliorée afin de réduire les erreurs de classification des pixels. Les fonctions d'imagerie infrarouge sont fusionnées pour supprimer l'artefact. Un mécanisme de croissance régionale est établi pour extraire et stocker des cibles en mouvement et des régions de fond pures. Selon les caractéristiques de la vidéosurveillance, il est le premier à établir le mécanisme de mappage cible, fond et vidéo pour proposer l'algorithme de reconstruction vidéo rapide. L'expérience montre que l'algorithme peut extraire complètement la cible en mouvement, établir l'arrière-plan pur dans diverses conditions complexes et réduire considérablement l'espace de stockage de la vidéo de surveillance.

3. Modèle de prédiction

Dans cette partie, nous proposons un algorithme de prédiction de chemin par apprentissage d'agents avec une application à la vidéosurveillance de la circulation routière. Grâce à un système d'apprentissage distribué, les caméras de surveillance auront la capacité de prédire la probabilité de passage par un chemin d'un véhicule dans ce réseau.

3.1. Algorithme de prédiction de chemin par apprentissage

Nous considérons un graphe (G) qui correspond à un réseau de circulation. Nous avons plusieurs nœuds (N_i) et plusieurs arêtes possibles (A_i).

Le but est de prédire le chemin [11,12,13], $V_a = A_1, A_2, \dots, A_n$ ou $V_n = N_1, N_2, \dots, N_n$ qu'une unité mobile peut prendre. Nous supposons qu'au niveau de chaque nœud du réseau, nous avons une caméra permettant de détecter l'unité mobile et de prédire son parcours.

Nous supposons que nous avons une succession de nœuds (caméras) qui va correspondre à un certain

nombre d'arêtes (A_i). Nous pouvons considérer le réseau comme un ensemble d'arêtes :

$$G_a = \{A_i\} \text{ ou comme un ensemble de nœud}$$

$$G_n = \{N_i\}.$$

Si nous cherchons à prédire un chemin, c'est soit considéré que nous avons une succession de nœuds ou une succession d'arêtes. Un véhicule qui passe à travers ce circuit va prendre un chemin (il va passer d'une arête à une autre).

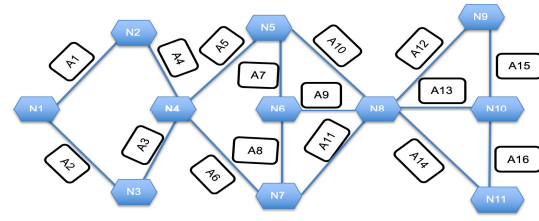


Figure 1 : Réseau de surveillance vidéo avec nombre d'arêtes et de segments

$$G_a = \{A_i\} \text{ ou } G'_n = \{N_i\} \text{ [Eq. 1]}$$

$$G_a = \{A_i\} \text{ ou } G'_n = \{N_i\} \text{ [Eq. 2]}$$

Théoriquement, il est censé prendre un chemin optimal V_{op} .

Posons V_{op} = chemin optimal

$$G_a = \{A_1, A_2, \dots, A_j\} / j \in \{1, 2, \dots, n\}$$

$$G'_n = \{N_1, N_2, \dots, N_j\} / j \in \{1, 2, \dots, n\} \text{ [Eq. 3]}$$

Vecteur d'arc $A_i, \dots, A_j \subset G$

$$L = |A_i| \implies \text{longueur de } A_i$$

$$V_{op} \in \{V_k\} / \{Coût(V_{op}) \leq Coût(V_k)\} \text{ [Eq. 4]}$$

Nous supposons que nous savons quel est le point de départ (A_1 ou N_1) et la destination (A_n ou N_n) et certainement, le conducteur va analyser le réseau en termes de coût (les approches les plus classiques). Les conducteurs cherchent généralement à optimiser ce chemin en termes de coût (le chemin avec le coût

minimal (V_{op}). Cependant, la circulation est assez particulière, car les coûts peuvent changer au fur et à mesure.

Nous considérons que chaque arête peut être caractérisée par un coût. Les solutions classiques de prédiction d'après la bibliographie sont les algorithmes de meilleur chemin. Cependant, quand nous descendons dans la circulation, nous avons deux problèmes : les coûts ne sont pas fixes (peuvent changer en fonction du temps ou l'existence de certaines arêtes ou nœuds est tout le temps changeant ; un feu rouge qui bloque momentanément le passage sur un chemin, etc.). Donc, nous pouvons avoir des coûts variables et même les distances d'un chemin en fonction du temps.

$$Coût(V) = \sum_i T_{A_i} / A_i \in V \text{ [Eq. 5]}$$

Dans notre cas, la supposition forte est que les véhicules connaissent leur chemin et leur destination. Comme nous cherchons à avoir un système de suivi, nous devons pouvoir prédire où va le mobile pour pouvoir le détecter avant son arrivée. C'est pourquoi

nous avons un système de caméras $G_{(cm)} = \{C_m\}$ [Eq. 6]

À chaque nœud, nous avons une caméra qui doit pouvoir constater que le véhicule y est arrivé et doit pouvoir prédire où est ce qu'il doit théoriquement aller. Notre système doit pouvoir recevoir en entrée un véhicule et nous prédire le chemin. Dès que nous détectons un objet, nous devons pouvoir en fonction de ces caractéristiques prévoir là où il va aller et quel est le chemin qu'il va suivre.

N_i {Arc Sortant : Nbr. Véhicule}

La stratégie adoptée est l'utilisation d'un processus d'apprentissage : pour chaque véhicule qui arrive à un nœud (N_i) (caméra) en passant par une arête (A_j), nous mémorisons le prochain arc emprunté pour mettre à jour la probabilité de passage par l'arc en fonction d'une origine. Selon la position du nœud, exemple un rond-point, nous pouvons avoir une multitude d'arêtes qui découle de ce nœud. Dans ce cas, il y'a une probabilité sur chaque arête. Nous pouvons apprendre cette probabilité (la probabilité qu'il passe par chaque arête qui découle de ce nœud).

$$Prob_{1(A_i, N_j, t)} = \frac{\sum_i NbrVehicule / A_j, t}{\sum_j NbrVehicule / A_j, t} \quad [Eq. 7]$$

$$Prob_{N_i(A_j, t)} = \frac{\sum_i Card_{N_i A_j(t)}}{\sum_{k=0}^{n_{nbreArcsN_i}} Card_{N_k A_i(t)}} \quad [Eq. 8]$$

Soit N_1 : le nœud

$$Prob_{2(V_i, N_j, t)} = \prod Prob_{1i} / k \in [Arc_1, \dots, Arc_j] \quad [Eq. 9]$$

Nous avons un modèle prédictif du chemin en lui-même qui se base sur un processus d'apprentissage. Nous allons avoir un processus d'apprentissage qui doit nous permettre de mémoriser pour chaque circuit le chemin de destination. Pour chaque véhicule qui vient à un nœud, nous devons pouvoir enregistrer les chemins qu'il a déjà empruntés, enregistrer le prochain nœud qu'il va prendre, observer les chemins qu'il va suivre et les enregistrer sur un vecteur de déplacement. À l'échelle du réseau routier, la caméra qui se trouve à un nœud ne peut nous renseigner que sur la probabilité du nœud suivant en se basant sur l'historique, mais n'a pas une vision globale du système. Dans ce cas, nous avons une base de mémoire (carte mentale partagée) où toutes les caméras enregistrent l'historique de passage des véhicules au cours du processus d'apprentissage qui va être alimenté par chacune des caméras au fur et à mesure.

3.2. Prédiction sans la connaissance des nœuds précédents

Nous pouvons apprendre cette probabilité distribuée en observant avant le comportement d'un nombre significatif de véhicules qui viennent avant le véhicule que nous voulons effectuer la mesure. Pour les n véhicules qui vont le précéder, nous affectons une probabilité à chaque arête. Nous avons dans ce cas une

prédiction basée sur un apprentissage de la distribution des passages précédents c'est-à-dire sur l'historique des passages précédents. Nous avons une prédiction qui se limite à une arête par rapport à une durée fixe. Dans notre cas, nous avons choisi une durée de 24 heures dans un premier temps en espérant améliorer l'algorithme pour des phases d'apprentissage plus longues (semaine, mois, année...).

Prochain arc / Noeud N_i :

$$N_c \{N_i\} / Prob_{N_i}(N, t) = \text{Max}(Prob_{N_i}((N_j, t)) \text{ et } Prob_{N_i}(N, t) \neq 0$$

[Eq. 10]

$$N_c \{N_2, N_3, N_4\} = Prob_{N_1}(A, t) = \begin{cases} Prob_{N_1}(N_2, t) = 2/5 \\ Prob_{N_1}(N_3, t) = 3/5 \\ Prob_{N_1}(N_4, t) = 0/5 \end{cases}$$

Représentation de la donnée (base de connaissances au niveau des caméras ou carte mentale partagée) :
Tab' [Noeud, TimeStamp, Noeud_Destination] = Matrice globale

Tableau 1.: Représentation de la donnée (base de connaissances au niveau des caméras)

Time Stamp (heure)	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}
[8,9[$T_{N_1:5}$ $T_{N_1:4}$	$T_{N_4:5}$	$T_{N_4:4}$	$T_{N_2:7}$ $T_{N_7:2}$	$T_{N_1:1}$ $T_{N_6:6}$	$T_{N_6:2}$	$T_{N_6:0}$ $T_{N_6:2}$	$T_{N_6:0}$ $T_{N_{10}:7}$ $T_{N_{11}:2}$	$T_{N_{10}:0}$		$T_{N_{11}:2}$
[9,10[$T_{N_1:10}$ $T_{N_1:4}$	$T_{N_1:10}$	$T_{N_4:4}$	$T_{N_2:10}$ $T_{N_7:4}$	$T_{N_1:1}$ $T_{N_6:6}$	$T_{N_6:2}$	$T_{N_6:0}$ $T_{N_6:2}$	$T_{N_6:0}$ $T_{N_{10}:7}$ $T_{N_{11}:2}$	$T_{N_{10}:0}$		$T_{N_{11}:2}$
[10H 11H [$T_{N_2:1}$ 0 $T_{N_1:1}$ 0	$T_{N_4:5}$	$T_{N_4:4}$	$T_{N_2:7}$ $T_{N_7:2}$	$T_{N_1:0}$ $T_{N_6:10}$		$T_{N_6:0}$ $T_{N_6:4}$	$T_{N_6:0}$ $T_{N_{10}:7}$ $T_{N_{11}:2}$	$T_{N_{10}:0}$		$T_{N_{11}:2}$
[11H 12H [$T_{N_2:1}$ 5 $T_{N_1:1}$ 0	$T_{N_4:5}$	$T_{N_4:4}$	$T_{N_2:7}$ $T_{N_7:2}$	$T_{N_1:1}$ $T_{N_6:6}$	$T_{N_6:2}$	$T_{N_6:0}$ $T_{N_6:2}$	$T_{N_6:0}$ $T_{N_{10}:7}$ $T_{N_{11}:2}$	$T_{N_{10}:0}$		$T_{N_{11}:2}$
[12H 13H [$T_{N_2:1}$ 5 $T_{N_1:1}$ 5	$T_{N_4:5}$	$T_{N_4:4}$	$T_{N_2:7}$ $T_{N_7:2}$	$T_{N_1:1}$ $T_{N_6:6}$	$T_{N_6:2}$	$T_{N_6:0}$ $T_{N_6:2}$	$T_{N_6:0}$ $T_{N_{10}:7}$ $T_{N_{11}:2}$	$T_{N_{10}:0}$		$T_{N_{11}:2}$

3.3. Prédiction avec connaissance des nœuds (chemins) précédents

Dans notre cas, notre prédiction va au-delà de la prochaine arête pour viser potentiellement tout le chemin du circuit. D'autres paramètres viennent s'ajouter à notre étude. En prenant en compte l'historique des chemins prises par un véhicule, nous pouvons déterminer la prochaine arête qu'il doit suivre toujours en utilisant le même processus d'apprentissage. Cette prédiction va se faire sur la base de l'historique des nœuds précédents c'est-à-dire en mettant en rapport la prédiction basée sur la probabilité sur chaque nœud et sur l'historique d'avant (le chemin emprunté par l'unité mobile avant son arrivée à un

nœud N_j); nous espérons non seulement faire la prédiction sur la prochaine arête, mais sur tout le chemin qu'il va prendre.

3.4. Mise en œuvre de l'apprentissage sur les systèmes distribués

Comme nous l'avons vu, les réseaux de neurones profonds peuvent être utilisés pour proposer de nombreux services à différents utilisateurs. Idéalement, l'apprentissage de ces modèles se fait à l'aide d'un ensemble de données provenant directement de ces mêmes utilisateurs. En pratique, cela demande de réunir toutes ces données sur une machine ou un serveur afin d'y entraîner un réseau de neurones profond.

Présentation des modèles de données

Pour modéliser notre base de connaissances, nous allons faire une représentation des données de traitement au niveau du système.

- **Représentation de la donnée (connaissance au niveau de chaque caméra) locale :**

Nous posons :

- TimeStamp (à quel moment le nœud a été rafraîchi pour la dernière fois)
- Nœud : Nœud où se trouve le mobile
- Nœud_Destination : Nœud suivant que l'unité mobile va prendre
- NbreVehicule = Nombre de fois qu'un véhicule est passé par un nœud.
- Nous posons le TimeStamp comme un slot de temps sur un intervalle de 1h

Tab [TimeStamp, Nœud_Destination] = NbreVehicule
Tableau 2 : Représentation de la donnée locale (base de connaissances au niveau de chaque caméra)

- Tab [TimeStamp, Chemin_Precedent, Nœud_Destination] = NbrVehicule

$$N_5 : V = [8h9h, \langle N_1, N_3, N_4, \rangle, N_6] = 3$$

$$= [8h9h, \langle N_1, N_2, N_4, \rangle, N_6] = 1$$

$$= [8h9h, \langle N_1, N_3, N_4, \rangle, N_7,] = 0 \quad [\text{Eq. 11}]$$

- **Représentation de la donnée (connaissance au niveau des caméras) ou carte mentale partagée :**

- Tab [Nœud] [TimeStamp, Chemin_Precedent, Nœud_Destination] = NbrVehicule

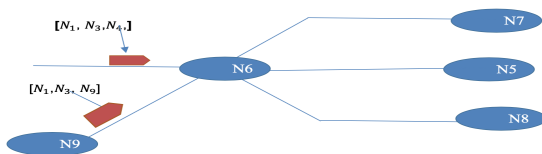


Figure 2 : Prédiction sans la connaissance des nœuds précédents

❖ Algorithme de prédiction de chemin

Nous avons besoin d'un modèle de traitement de l'information pour prédire le chemin et la destination d'une unité mobile dans une architecture de vidéosurveillance.

Algorithme Prediction_de_parcours

{Cet algorithme prédit le chemin et la destination d'un mobile dans un réseau de vidéosurveillance}

Début

• Prédiction au niveau de nœud N_i

Prochain Nœud $N_i(\text{Vehicule}) =$

$$N \subset \{N_j\} / \text{Prob}_{N_i}(N, t, Vh) = \text{Max} (\text{Prob}_{N_i}(N_j, t, Vh))$$

Et $\text{Prob}_{N_i}(N, t, Vh) \neq 0$

Avec Vehicule \rightarrow NumeroVehicule

Vh \rightarrow Vecteur Historique = $\langle \text{Nœud}_{\text{Depart}}, \dots, \text{Nœud}_{\text{Destination}} \rangle$

Nous avons alors :

Prochain Nœud $N_i(\text{Vehicule}) \rightarrow \text{Nœud}_{\text{Next}}$

Vehicule.vh \leftarrow Vehicule.vh + N_i

• Prédiction de parcours

Prochain Nœud $N_i(\text{Vehicule}) \rightarrow \text{Nœud}_{\text{Next}}$

VehiculeTest \leftarrow Vehicule ;

VehiculeTest.Vh \leftarrow VehiculeTest.Vh + N_i ;

For $i \leftarrow 0$ avec i de 0 à n

Prochain Nœud $N_{\text{Next}}(\text{VehiculeTest})$

$\rightarrow \text{Nœud}_{\text{Next}}$

VehiculeTest.Vh \leftarrow VehiculeTest.Vh + $\text{Nœud}_{\text{Next}}$

$\text{Nœud}_{\text{Next}} \leftarrow \text{Nœud}_{\text{Next}}$

Fin

Dans [11,12], nous avons proposé des modèles mathématiques qui permettent d'évaluer le temps de traitement des données et il en résulte que l'architecture distribuée offre les meilleures performances. Dans [13], Nous avons montré par rapport à notre proposition, l'optimisation de la consommation énergétique dans les réseaux de vidéosurveillance.

Nous allons maintenant appliquer notre modèle de prédiction présenté ci-dessus dans une architecture de vidéosurveillance pour optimiser le volume de stockage des données dans une telle architecture.

4. Évaluation des besoins en matière de stockage

Dans cette partie, nous allons appliquer notre modèle de prédiction dans une architecture de vidéosurveillance afin d'optimiser le volume de stockage des données. Pour atteindre nos objectifs, nous allons évaluer notre architecture afin d'appliquer notre modèle de prédiction de parcours. Pour l'optimisation du stockage de données, nous prenons comme paramètres :

- Durée de fonctionnement caméra = D_{fc}
- Durée de fonctionnement du système = D_{fs}

- Nombre image capturée haute résolution avec algo= Ni_{haa}
- Nombre image capturée basse résolution avec algo= Ni_{baa}
- Nombre image capturée haute résolution sans algo= Ni_{hsa}
- Nombre image capturée basse résolution sans algo= Ni_{bsa}
- Capacité de stockage caméra entrée du réseau (Go) = C_{cn1}

Nous posons :

T_{fc} = Fréquence d'image capturée par seconde pour une caméra

$T_{mc} = 24H$

T_c = Taux de compression

Nous avons :

$D_{fc} = T_d + T_{ac} + T_t + T_{at}$ [Eq. 12]

$Ni_{haa} = D_{fc} \times T_{fc}$ /Eq. 13]

$D_{fs} = D_{fc} \times Nbr_{ct} \times T_{fc}$ [Eq. 14]

$C_{cn1} = T_{im} \times T_{fc} \times T_{mc}$ [Eq. 15]

□ Avec algorithme de prédiction

Sans compression

$C_{asc} = T_{im} \times Ni_{haa} \times D_{fs} \times N_{ct} \times Nbr_T$

+ C_{cn1} [Eq. 16]

Avec compression

$C_{aac} = C_{asc} \times T_c$ [Eq. 17]

□ Sans algorithme de prédiction

Sans compression

$C_{ssc} = T_{im} \times T_{fc} \times T_{mc} \times Nbr_{ct} \times Nbr_T$

[Eq. 18]

Avec compression

$C_{sac} = C_{ssc} \times T_c$ [Eq. 19]

Tableau 2. - Données expérimentales sur l'optimisation du volume de stockages dans les systèmes de vidéosurveillance

Durée de fonctionnement caméra = D_{fc}	45,02	40,02	36,02	32,74	30,02	27,71	25,73	22,52	20,02	18,02
Durée de fonctionnement du système = D_{fs}	360,18	640,35	864,52	1047,97	1200,86	1330,26	1441,2	1441,36	1441,52	1585,84
Nombre image capturée haute résolution avec Algo= Ni_{haa}	675	600	540	491	450	415	386	337	300	270
Nombre image capturée basse résolution avec Algo= Ni_{baa}	360	320	288	261	240	221	205	180	160	144
Nombre image capturée haute résolution Sans algo= Ni_{hsa}	1296000	1296000	1296000	1296000	1296000	1296000	1296000	1296000	1296000	1296000
Nombre image capturée basse résolution sans algo = Ni_{bsa}	691200	691200	691200	691200	691200	691200	691200	691200	691200	691200

Capacité de stockage caméra entrée du réseau (Go) = C_{cn1}	1,16	1,16	1,16	1,16	1,16	1,16	1,16	1,16	1,1664	1,16
Taille image Haute Résolution 640x480 = T_{im}	900	900	900	900	900	900	900	900	900	900

5. Résultats et discussions

Tableau 3. - Résultats de l'optimisation du volume de stockage

Capacité de stockage avec algorithme de prédiction et données sans compression (Go) = C_{asc}	2,91	6,70	11,25	15,99	20,63	25,05	29,20	29,21	29,21	35,11
Capacité de stockage avec algorithme de prédiction et données avec compression (Go) = C_{aac}	0,87	2,01	3,37	4,79	6,19	7,51	8,76	8,76	8,767	10,53
Capacité de stockage sans algorithme de prédiction et données sans compression (Go) = C_{ssc}	9,33	18,66	27,99	37,32	46,65	55,98	65,31	74,64	83,98	102,64
Capacité de stockage sans algorithme de prédiction et données avec compression (Go) = C_{sac}	3,079	6,15	9,23	12,31	15,39	18,47	21,55	24,63	27,71	33,87

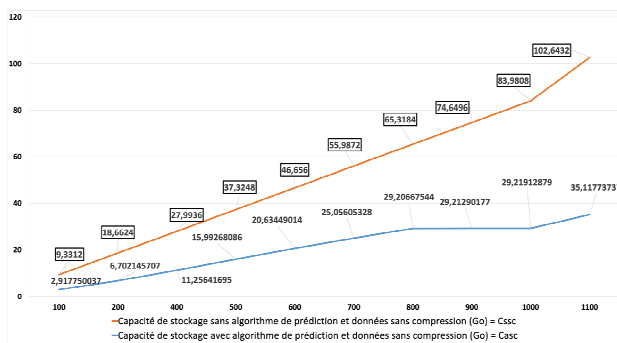


Figure 3 : Stockage de données de vidéosurveillance sans compression de données

La figure 3 représente la variation du stockage de données d'une architecture de vidéosurveillance en fonction du nombre de voitures qui passent sur un tronçon sans compression de données issues du trafic. Nous constatons pour une architecture de vidéosurveillance prédictive, une réduction du stockage

des données par rapport à une architecture standard où les caméras doivent fonctionner 24h/24. Ceci s'explique du fait que pour une architecture de vidéosurveillance prédictive, seule la caméra d'entrée est active 24h/24 alors que pour une architecture standard, toutes les caméras sont opérationnelles. Le principe de prédiction se base sur un modèle de vidéosurveillance qui se base sur l'événementiel. Les caméras du réseau sont opérationnelles que s'il y'a mouvement (détection d'un mobile) dans leur champ de vision. Leurs durées d'activité se résument au temps d'attente de l'arrivée du mobile détecté par la caméra précédente, jusqu'à l'activation de la prochaine caméra dans la direction du mobile. Donc, seules les informations capturées durant cette période seront stockées. Ce qui réduit considérablement le volume de stockage.

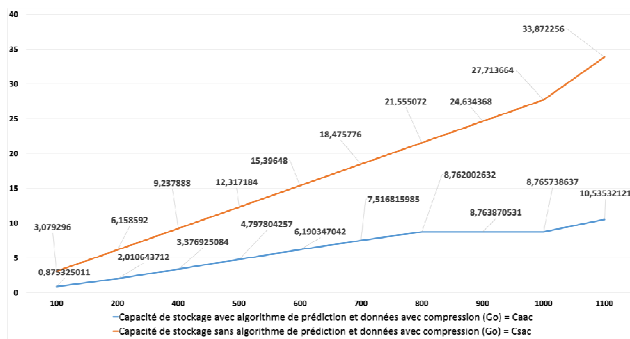


Figure 4 : Stockage de données de vidéosurveillance avec compression de données

La figure 4 représente la variation du stockage de données d'une architecture de vidéosurveillance en fonction du nombre de voitures qui passent sur un tronçon avec compression de données issues du trafic. Nous constatons la même variation pour un système de vidéosurveillance prédictif et non prédictif. Nous remarquons une diminution de volume beaucoup plus importante que dans la figure 4 Ceci s'explique du fait que nous avons appliqué l'algorithme de compression de données qui réduit considérablement le volume de stockage à moins de 33% de la normale. Le format de compression H.264 est de loin la technique de compression vidéo la plus efficace disponible aujourd'hui. Un encodeur H.264 peut réduire la taille d'un fichier vidéo numérique de plus de 80 % par rapport à la norme Motion JPEG et de 50 % par rapport à la norme MPEG-4, sans que la qualité d'image ne soit compromise.

6. Conclusion

Dans cette étude, nous avons examiné les problèmes de stockage de données dans les architectures de vidéosurveillance routière. En premier lieu, nous avons proposé un algorithme de prédiction du chemin par l'apprentissage. Ensuite, nous avons formulé le problème de stockage et appliquons notre modèle de prédiction de trajectoire pour optimiser le volume de stockage.

Ensuite, en tenant compte de l'exigence de la vidéosurveillance en temps réel et des caractéristiques des tâches de vidéo surveillance, nous avons appliqué notre algorithme de prédiction qui va désactiver par défaut toutes les caméras du réseau de vidéosurveillance à l'exception de celle qui est à l'entrée du réseau de vidéo surveillance. Une fois la prédiction réalisée, il va activer la caméra suivante (celle qui est censée recevoir le mobile dans son champ de vision). Le suivi et l'activation des caméras au niveau de la section se feront en parallèle avec la prédiction du nœud suivant (caméra). Cette méthode réduit considérablement le volume de données stocké.

Des expériences ont démontré l'efficacité de notre méthode en augmentant le nombre de sections des réseaux de vidéosurveillance ainsi que le trafic à chaque section. Il serait très important d'optimiser notre algorithme de prédiction afin d'avoir de meilleur résultat en prenant en compte d'autre facteur compte la

taille des voitures ou le type de mobile. Il sera mis en œuvre dans un proche avenir.

REFERENCES

- [1] Zhao C, Zhao J, Lin X, Wu C. Capacity of P2P on-demand streaming with simple, robust, and decentralized control. In: Proceedings of INFOCOM; 2013. p. 2697–705
- [2] Amani P, Bastani S, Landfeldt B. Towards optimal content replication and request routing in content delivery networks. In: Proceedings of IEEE international conference on communications; 2015. p. 5733–9.
- [3] Jia S, Xu C, Guan J, Zhang H, Muntean GM. A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks. *IEEE Trans Broadcast* 2014;60(2):370–84.
- [4] Y. Yu, D. Doermann, Model of object-based coding for surveillance video, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Process. (ICASSP), 2005, pp. 693–696.
- [5] A. Hakeem, K. Shafique, M. Shah, An object-based video coding framework for video sequences obtained from static cameras, in: Proceedings of the ACM International Conference on Multimedia, 2005..
- [6] A. Vetro, T. Haga, K. Sumi, H. Sun, Object-based coding for long-term archive of surveillance video, Technical Report TR-2003-98 MERL (2003).
- [7] Naeem Ramzan Toni Zgaljic EbroulIzquierdo School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom
- [8] J.-H. Lee, G.-G. Lee, and W.-Y. Kim, "Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 742-749, 2003.
- [9] Z. Rasheed and M. Shah, "Detection and representation of scenes in videos," *IEEE transactions on Multimedia*, vol. 7, pp. 1097-1105, 2005.
- [10] ShiQiu, Y. Tang, Y. Du, S. Yang, The infrared moving target extraction and fast video reconstruction algorithm, *Infrared Physics & Technology* (2018), doi: <https://doi.org/10.1016/j.infrared.2018.11.025>
- [11] Diop P.S., Mbacke A.B., Mendy G. (2017) Centralized and Distributed Architectures: Approximation of the Response Time in a Video Surveillance System of Road Traffic by Logarithm, Power and Linear Functions. In: Puliafito A., Bruneo D., Distefano S., Longo F. (eds) Ad-hoc, Mobile, and Wireless Networks. ADHOC-NOW 2017. Lecture Notes in Computer Science, vol 10517. Springer, Cham
- [12] Diop P.S., Mbacké A.B., Mendy G. (2016) Predictive Assessment of Response Time for Road Traffic Video Surveillance Systems: The Case of Centralized and Distributed Systems. In: Hsu CH., Wang S., Zhou A., Shawkat A. (eds) Internet of Vehicles – Technologies and Services. IOV 2016. Lecture Notes in Computer Science, vol 10036. Springer, Cham

[13] Diop, P.S., Mbacké, A.B., Mendy, G., Gaye, I., Bilong, J.R.N. (2019). Optimisation of Energy Consumption in Traffic Video Monitoring Systems Using a Learning-Based Path Prediction Algorithm. In: Palattella, M., Scanzio, S., Coleri Ergen, S. (eds) Ad-Hoc, Mobile, and Wireless Networks. ADHOC-NOW 2019. Lecture Notes in Computer Science(), vol 11803. Springer, Cham. https://doi.org/10.1007/978-3-030-31831-4_26