Analyse et enrichissement de flux compressés : application à la vidéo surveillance

Marc Leny

Thèse de doctorat de Télécom & Management SudParis
dans le cadre de l'école doctorale S&I en co-accréditation avec
l'Université d'Evry-Val d'Essonne

Spécialité :
Mathématiques et informatique

Par
M. Marc LENY

Thèse présentée pour l'obtention du diplôme de Docteur
de Télécom & Management SudParis

Analyse et enrichissement de flux compressés : application à la vidéo surveillance

Soutenue le 17/12/2010 devant le jury composé de :

Monsieur	**Malik MALLEM**	**Président**
Monsieur	**Touradj EBRAHIMI**	**Rapporteur**
Monsieur	**Philippe FUCHS**	**Rapporteur**
Monsieur	**Frédéric DUFAUX**	**Examinateur**
Monsieur	**Denis MARRAUD**	**Examinateur**
Madame	**Françoise PRETEUX**	**Directrice de thèse**
Monsieur	**Didier NICHOLSON**	**Responsable industriel**

Thèse n°2010TELE0031

« L'homme et sa sécurité doivent constituer la première préoccupation de toute aventure technologique. »

Albert Einstein

Remerciements

Puisque l'exercice de la thèse m'offre l'opportunité d'exprimer ma reconnaissance vis à vis des nombreuses personnes liées de près ou de loin à l'aboutissement de ces travaux, je tiens à citer et remercier ici :

- Madame Françoise Prêteux, pour m'avoir accueilli fin 2005 alors que je cherchais en vain un poste dans l'imagerie numérique. Merci pour m'avoir suggéré un Master 2 Recherche puis un Doctorat en milieu industriel en m'aiguillant vers Didier. Merci pour avoir accepté d'encadrer cette thèse en convention CIFRE, dont la finalisation bien que tardive n'aura jamais été aussi proche !
- Monsieur Didier Nicholson, pour m'avoir d'abord encadré en stage de DEA, déjà sur l'analyse dans le domaine compressé, puis pour l'opportunité de la thèse CIFRE dans le laboratoire MMP de Thales que j'appréciai déjà grandement. Une première thèse en tant qu'encadrant industriel soulève des défis inédits, mais ce manuscrit prouve que nous l'aurons menée à termes.
- Messieurs Touradj Ebrahimi et Philippe Fuchs, mes rapporteurs, pour avoir acceptés d'étudier en détail cette prose,
- Messieurs Malik Mallem Frédéric Dufaux et Denis Marraud, membres de mon jury,
- Cédric, Erwann, puis Séverin, mes collègues de bureau. Même si depuis le départ de Cédric nous avons enfin le droit de mettre un fond de musique, l'ambiance aura toujours été excellente, les sorties vélos fatigantes, et les soirées off distrayantes.
- L'ensemble des membres passés et actuels du laboratoire MMP de THALES : Bruno, Christophe, Emmanuel, François, Bertrand, Yannick, Rachid, Jérôme, Cyril, Benjamin, Gwenaël, Charles et Antonin.
- Les stagiaires et doctorants que j'ai eu l'occasion d'encadrer ou de côtoyer, en particulier Marc, Sébastien et Clément : pensez à boucler vos thèses avant les trois ans, c'est un conseil avisé !
- L'équipe du Département ARTEMIS de TELECOM SudParis, qui m'a toujours accueilli avec le sourire et des réponses ou des stagiaires potentiels lorsque les occasions se sont présentées,
- Les autres membres de THALES avec qui j'ai pu travailler, monter des projets, discuter, défendre nos brevets, etc., dont Benjamin, Anne-Marie, Christophe, Eric, Fabienne,
- Les partenaires des projets Caretaker et Infom@gic, puis Vanaheim, Skymedia, et tous les autres à venir,
- Enfin je remercie mes amis : Julien mon ami de 20 ans, Etienne, Isa & Ophélie, mes colocs le temps de cette thèse, Jérémie pour nous avoir remercié dans sa thèse, la bande de juristes et associés, les rescapés du groupe de l'INT et DCU,
- Last but not least, je remercie surtout toute ma famille, qui n'aura pas eu peur de me voir finaliser officiellement mes études peu avant mes trente ans. Merci à mes parents pour avoir su me pousser et m'encourager tout au long de mes études jusqu'à ce doctorat.

Table des matières

Table des figures

Liste des tableaux

Chapitre 0

Introduction

La sécurité et le bien-être des personnes sont aujourd'hui au cœur des politiques sociétales de tous les pays. Toutefois, selon les dirigeants et les richesses des nations, les réponses apportées varient grandement, mais la vidéosurveillance reste la solution la plus communément mise en place. Les caméras se multiplient, comme autant d'agents électroniques visant à faire reculer délinquance et criminalité. Londres est reconnue pour ses centaines de milliers de caméras, qui ont par exemple permis de déterminer le mode opératoire des terroristes suite aux attentats dans le métro londonien en 2005. Mexico est aujourd'hui en train de s'équiper, principalement *via* un programme d'aménagement de la ville, de milliers de caméras interconnectées. Les appels d'offre se multiplient dans le Monde sur ce modèle, que ce soit à Singapour ou à Moscou.
Toutefois les limitations des systèmes actuels posent des défis scientifiques et technologiques, tel qu'analyser en temps réel par informatique les nombreux flux récoltés sans pour autant investir dans un grand nombre de serveurs. Une première réponse consiste à diminuer la résolution en sortie de chaque caméra pour traiter des dizaines de vidéos sur une plateforme informatique. La conséquence est de rendre l'identification d'individus sur des plans larges impossible. Par ailleurs, il y a d'ores et déjà beaucoup trop de caméras par rapport au nombre d'opérateurs les visualisant. Pour être efficace, une personne en charge de la sécurité ne devrait pas avoir plus de 4 écrans devant elle, pour une durée maximale de 2 heures. De nombreuses infrastructures dans les grandes villes proposent aujourd'hui un opérateur pour une centaine de caméras, qu'il « observe » pendant 8 heures. Pour revenir aux attentats de Londres, si les bons résultats de reconstitution ont été annoncés, le déroulement de l'opération est moins connu : 2500 agents de Scotland Yard ont été mobilisés pendant plus d'une semaine pour visualiser une à une toutes les séquences vidéos recueillies auprès de la ville, des boutiques, des banques … Un travail titanesque a été entrepris pour aboutir aux résultats espérés. Et il s'agit encore d'enquête *a posteriori*. Il n'y a pas encore de solution temps réel capable d'interpréter une situation complexe.
Le décalage est d'autant plus grand que la population est influencée par les médias, films et séries télévisées. *Ennemi d'état* de Tony Scott montrait déjà en 1998 Will Smith qui tentait d'échapper à la surveillance des caméras, où l'on pouvait à partir d'une caméra dans une boutique tourner autour d'une personne pour déterminer ce que son sac contenait. Plus récemment *Les Experts* ou autres *NCIS* montrent des scènes dans lesquelles les personnages zooment sur des vidéos, améliorent la qualité, re-zooment, et identifient au final un visage présent dans un reflet sur une zone initiale de 20 pixels au carré. Des progrès sont faits tous les jours, mais on est loin du sentiment globalement partagé du « *Big Brother is watching you !* ». Certes les réseaux se déploient, le nombre de caméras augmente, mais les outils et les moyens d'analyse peinent à suivre. Au final, *Big Brother* ne nous observe pas comme nous pouvons l'imaginer (pour le moment).

Dans ce contexte, les travaux de recherche font progresser les chaînes de traitement automatique de la vidéo à tous les niveaux, depuis les algorithmes de segmentation jusqu'aux outils de classification et de levées d'alarme sur la détection d'événements anormaux. Même si le remplacement des agents de sécurité par des ordinateurs n'est pas à l'ordre du jour, il est important de faciliter leur tâche, en identifiant par exemple les caméras parmi un réseau qui présentent une activité de celles sans personne dans le champ de vision, ou encore celles où un événement particulier se produit de celles repérant une activité habituelle. A défaut de pouvoir entièrement automatiser la détection d'événements anormaux ou souhaitables d'éviter, tels qu'un bouchon sur une autoroute, un braquage de banque, ou une entrée dans une zone à accès restreint, l'objectif est dans un premier temps de permettre une meilleure exploitation des infrastructures déjà en place.

Quelques solutions d'automatisation existent déjà : surveillance d'accès proposée par ACIC [ACIC, 2010], lecture automatique de plaques d'immatriculation (parking de l'aéroport Roissy – Charles de Gaulle équipé par Thales), solutions plus grand public avec caméras intelligentes de Logitech ou AXIS. Le problème des outils actuels est qu'ils nécessitent une grande puissance de calcul pour aboutir à des résultats haut niveau, impliquant souvent la présence d'un ordinateur pour traiter un unique flux vidéo sans atteindre le temps réel (6 images par seconde sont par exemple exploitées dans les infrastructures du métro de Turin).

Les travaux de la présente thèse se sont initialement concentrés sur des problématiques de bas niveau liées à vidéosurveillance, avec la segmentation des objets mobiles. En effet, on distingue généralement en surveillance le premier plan de l'arrière plan selon le critère suivant : l'arrière plan est constitué de ce qui ne bouge pas, et n'est pas utile à l'application considérée. Ainsi l'environnement (route, bâtiment, mur, banc, etc.) n'est généralement pas important pour la sécurité. Le premier plan en revanche sera constitué des piétons, véhicules, animaux, robots, etc. se déplaçant dans la scène. Selon les applications, il peut y avoir des échanges entre ces deux plans. Une voiture qui se gare pour plusieurs heures dans une rue n'a plus besoin d'être identifiée comme premier plan une fois qu'elle est immobile. Nous nous sommes donc concentrés sur la distinction entre ce fond et les objets qui s'y déplacent. Prenant en compte l'expertise des deux unités d'accueil, le département ARTEMIS de TELECOM SudParis et le laboratoire MultiMedia Processing – MMP – de Thales, l'idée initiale était d'étudier la faisabilité de cette segmentation en s'appuyant uniquement sur les données contenues dans un flux vidéo compressé.

Cette thèse comporte quatre chapitres, décrivant les travaux menés depuis la caractérisation de la singularité des données du domaine compressé jusqu'à l'exploitation des résultats de la méthode d'analyse proposée.
L'état de l'art détaillé au chapitre 1 s'articule autour de deux thématiques nécessaires à la compréhension des outils développés. Dans un premier temps une synthèse sur la compression vidéo est proposée, reprenant les principes de suppression de la redondance spatiale et temporelle. L'analyse dans le domaine compressé étant dépendante des standards de compression, les points communs et les différences entre les normes MPEG-1, MPEG-2, MPEG-4 Part 2 (Visual) et Part 10 (H.264 / AVC) sont également rappelés. Les travaux présentés font partie d'une rupture technologique avec la majorité des investigations menées aujourd'hui, qui s'appuient sur les informations à l'échelle du pixel pour construire les chaînes de traitements et d'analyse d'image ou vidéo. Toutefois l'analyse dans le domaine compressé a débuté par l'extraction de contours à partir des coefficients JPEG il y a presque 25 ans, et nous rappellerons dans ce chapitre les différentes approches et les résultats qui peuvent être attendus des méthodes trouvées dans la bibliographie.
A partir de ces références, nous avons construit notre outil de segmentation dans le domaine compressé, qui est décrit dans le chapitre 2. Nous nous sommes attachés à fournir une méthode unifiée d'analyse, constituant notre première contribution. Les algorithmes mis au point ont en effet été validés sur des flux MPEG-2, MPEG-4 Part 2 et Part 10 (AVC). Le premier maillon de notre chaîne est donc un parseur de flux dédié au standard, qui extrait les informations compressées et les met en forme dans une structure commune aux différents standards. La suite des traitements est unifiée : une reconstruction des séquences en basse résolution est effectuée, et les vecteurs de mouvement sont filtrés et enrichis afin de fournir des coefficients transformés et des vecteurs à chaque bloc de la séquence. Deux segmentations sont en réalité menées en parallèle, puis un module de décision coopérative fusionne ces deux résultats intermédiaires selon les performances de chaque segmentation et le type d'image traitée pour aboutir à la segmentation finale.

Le cadre fortement industriel de la convention CIFRE dans laquelle s'inscrivent ces travaux a favorisé la validation des outils d'analyse sur un important volume de données. Bien que nous nous soyons restreints au contexte de la vidéosurveillance, plusieurs corpus liés aux projets collaboratifs ou internes à Thales ont été exploités.
Le chapitre 3 se charge dans un premier temps de décrire ces données qui représentent plusieurs centaines de vidéos, en intérieur ou en extérieur, de jour ou de nuit, avec des véhicules, des piétons ou des animaux, dans le visible ou l'infrarouge. Nous exposons ensuite les résultats obtenus lors des tests de segmentation sur ces vidéos et discutons aussi bien les performances positives que négatives. Une première application autonome est proposée, s'appuyant sur notre contribution au sein du projet Infom@gic, projet structurant du Pôle de Compétitivité Cap Digital. Il s'agit d'un outil d'aide à l'investigation permettant de rechercher un véhicule selon un témoignage. L'analyse dans le domaine compressé présente en effet comme principal avantage de traiter d'importants volumes de vidéos dans des temps réduits (plusieurs centaines d'images par seconde). Si une personne indique avoir vu une voiture verte sortant d'un parking entre 10h et 10h20 causer l'accident que tentent de résoudre des forces de l'ordre, la collecte des données des caméras avoisinantes et leur analyse rapide permet d'accélérer l'enquête et de gagner du temps lors des premières heures de recherche, cruciales pour l'identification d'une personne en fuite ou ayant disparue.
Comme ce premier démonstrateur le montre, nous avons considéré l'analyse dans le domaine compressé sans lui attribuer comme unique finalité la segmentation des objets mobiles. Les coûts réduits en temps de calcul et en mémoire peuvent et doivent être exploités pour proposer de nouveaux services. Nous avons ainsi imaginé un ensemble d'outils utilisant directement l'analyse dans le domaine compressé, qui ont fait l'objet d'une famille de brevets.
Le chapitre 4 expose ces applications qui couvrent l'accélération des traitements au niveau du pixel, l'adaptation de débit ou la protection aux erreurs d'un flux à transmettre sur réseau, la cryptographie visuelle ciblée sur les objets mobiles ou encore des outils de tatouage et/ou stéganographie s'adaptant au contenu pour ne pas dégrader la qualité des régions d'intérêt.
La conclusion générale, après avoir synthétisé l'ensemble des travaux exposés, ouvre sur les perspectives de recherches et d'industrialisation futures, puisque Thales continue l'exploitation des résultats présentés et poursuit le développement de services s'appuyant sur l'analyse dans le domaine compressé.

Chapitre 1

Etat de l'art sur l'analyse de flux vidéo compressés pour la vidéosurveillance

Résumé du chapitre

Que ce soit pour des considérations liées aux débits sur les réseaux ou les capacités de stockage limitées, un système de vidéosurveillance s'appuie sur de la vidéo compressée. Différents standards ont été et sont employés dans ce secteur. La première partie de ce chapitre est consacré à la compression de l'image et de la vidéo, et aux spécificités des flux dans le secteur de la surveillance.
De nombreux travaux de recherche et aujourd'hui de solutions commerciales proposent des outils d'analyse automatique de vidéo, permettant par exemple de détecter une intrusion dans une pièce. Les algorithmes mis en jeu sont généralement complexes et requièrent une grande capacité de calcul. En réponse à cette problématique, l'analyse dans le domaine compressé essaie d'exploiter au mieux les informations des flux sans décodage pour permettre le plus souvent une détection d'activité voire une segmentation des objets mobiles. La seconde partie de cet état de l'art est consacrée à ces méthodes, présentant leurs forces et faiblesses respectives.

Sommaire du chapitre

1.1 Introduction

La vidéosurveillance est un des enjeux majeurs de la dernière et des prochaines décennies en termes de réponse aux problématiques de sécurité. A niveau national la volonté de passer de 20.000 à 60.000 caméras entre 2009 et 2011 a été annoncé par le Ministre de l'Intérieur Brice Hortefeux le 12/11/09 lors d'une conférence de presse, principalement en réponse à la délinquance. La ville de Paris prévoit elle aussi un triplement du nombre de caméras entre 2009 et 2012 ; les régies de transports telles que la RATP ou la SNCF collaborent sur des projets de recherche coopératifs français ou européens (BOSS, Caretaker) pour évaluer les différentes solutions existantes et préparer l'avenir ; les forces de l'ordre ont de plus en plus recours à l'enregistrement de leurs interventions ; les forces armées sécurisent des périmètres tels que des camps de base temporaires par des capteurs vidéo déposés, ou proposent des solutions de surveillance automatisée de frontière ; etc.
Un volume croissant de flux de vidéosurveillance est donc généré. Il faut à la fois être capable de le transmettre, de le stocker et/ou de le traiter en temps réel selon les applications visées. L'analogique a progressivement laissé la place au numérique, représentant moins de dix pourcents des installations fin 2009, principalement des équipements de particuliers ou de petits commerces. Dès lors, les limitations dues aux divers réseaux et matériels de stockage, pour des flux numériques, ont été contournés par l'utilisation massive de la compression vidéo. D'abord Motion JPEG et MPEG-1, puis MPEG-2, MPEG-4 Part 2 et enfin MPEG-4 Part 10 (AVC et SVC). Certaines études et installations se sont appuyées sur du Motion JPEG et Motion JPEG 2000, mais restent marginales, bien que présentant certains atouts. D'autres enfin utilisent des formats propriétaires développés par des sociétés spécialisées en surveillance, mais font de plus en plus figure d'exceptions avec l'adoption massive des standards MPEG.
Concernant l'analyse des flux vidéos, la grande majorité des algorithmes existants travaille directement sur les pixels du flux vidéo, que ce soit directement en sortie de capteur ou après une décompression d'un flux ayant été compressé pour la transmission ou le stockage. La tendance observée est l'application de traitement d'image, après décompression, au sein d'unités dédiées raccordées au réseau, tout comme les enregistreurs. Plusieurs flux vidéo peuvent ainsi être traités, au prix d'une consommation de ressources importante. Peu de travaux de recherche se sont intéressés à traiter l'information telle qu'elle circule sur les réseaux ou est enregistrée. L'analyse vidéo dans le domaine compressé doit se contenter de peu de données, et les algorithmes de l'état de l'art présentent peu de variabilité. Rares sont les outils du domaine compressé qui ont abouti à un applicatif autonome, et dans tous les cas une solution universelle n'a pas percé, si bien que chaque méthode, forte de ses avantages et inconvénients, présente des verrous persistant qui feront choisir l'une plutôt que l'autre en fonction de la tâche à réaliser.

1.2 Les flux vidéo compressés

Les différents outils d'analyse développés s'appuient sur des flux vidéo compressés. Ceci implique une étude préalable des différents standards de compression utilisés dans le cadre de la vidéosurveillance après celle des principes sur lesquels chacun s'appuie. L'objet n'étant toutefois pas ici d'être exhaustif sur la compression vidéo, seuls les éléments nécessaires à la compréhension des différents algorithmes ainsi que les spécificités des profils utilisés en vidéosurveillance seront décrits. Pour un complément d'information sur la compression vidéo,

les références [Watkinson, 2004 ; Pereira et Ebrahimi, 2002 ; Flierl et Girod , 2004 ; Wiegand et al., 2003] sont autant de sources potentielles de détails, en complément bien entendu des normes elles-mêmes.

1.2.1 Vue d'ensemble de la compression vidéo

Deux principes constituent les fondements de la compression vidéo : l'exploitation de la redondance spatiale et celle de la redondance temporelle. La Figure 1 propose un schéma global de la compression vidéo, dont les principaux éléments seront exposés au cours de ce chapitre. L'ensemble des descriptions suivantes détaillent le codage de l'information liée à la luminance (représentation en niveaux de gris de l'image), qui sera concrètement l'information utilisée par la plupart des algorithmes présentés en 1.3. Le codage de l'information de chrominance (permettant de reconstruire la couleur de chaque image) n'étant pas directement lié aux présents travaux, il ne sera pas exposé ici, mais est disponible dans beaucoup des articles cités.

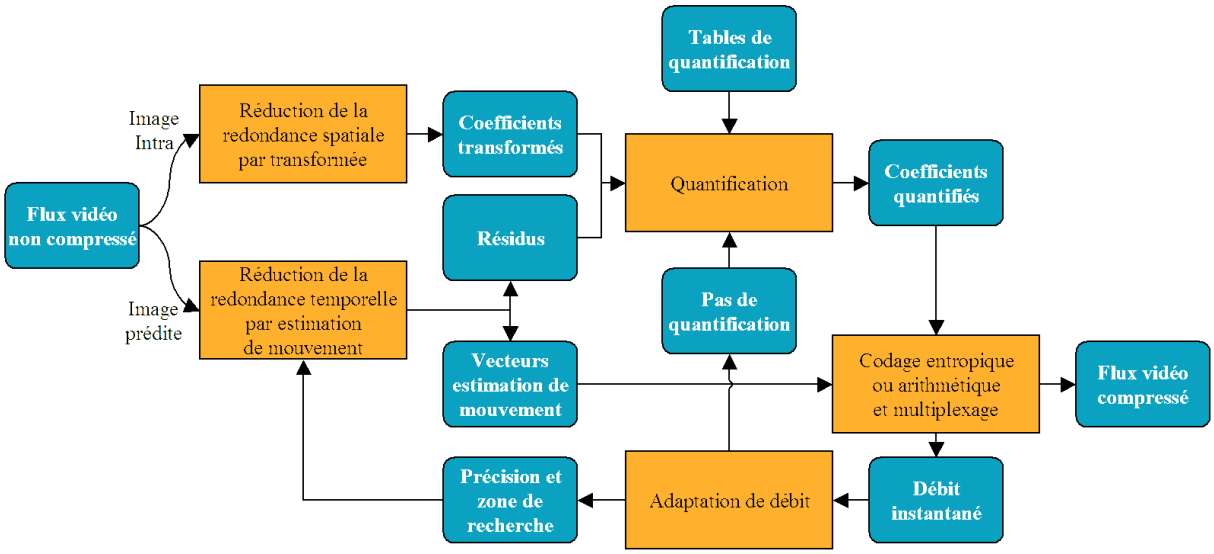


Figure 1 - Schéma global de la compression vidéo

1.2.1.1 La redondance spatiale

L'utilisation de la redondance spatiale correspond à de la compression d'images fixes, tout comme le feraient les standards JPEG ou JPEG2000 par exemple, qui constituent aujourd'hui les principaux formats d'échange d'images (appareils photo numériques, imagerie médicale ou satellite, etc.). Deux approches majeures sont aujourd'hui utilisées par les standards liés à la vidéosurveillance : la division de l'image en blocs de pixels ou une approche globale sur l'ensemble de l'image. La deuxième solution sera abordée dans la section consacrée à Motion JPEG 2000 (1.2.6), donc seule la première sera pour le moment détaillée.

Une image est découpée en blocs, de 4x4 ou 8x8 pixels selon le standard utilisé. Une transformée est appliquée aux coefficients (les valeurs des pixels) de ce bloc, permettant de passer du domaine spatial au domaine fréquentiel. Les différentes transformées et leurs particularités seront reprises lors de la description des différents standards ; les détails suivants permettant d'illustrer s'appuient sur la Transformée en Cosinus Discrète (ou DCT pour *Discrete Cosine Transform*). Dans ce cas, la transformée utilisée par les standards

MPEG et ITU s'appuie sur la matrice 8x8 de la DCT type II orthogonalisée (standard IEEE-1180-1990), produit de deux vecteurs X et X^T donnés par :

$$X_0 = \frac{1}{\sqrt{2}} \sum_{n=0}^{N-1} x_n \quad \text{et} \quad X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

où X_k est la $k^{ième}$ composante du vecteur X, x_n est la $n^{ième}$ valeur du signal source x, comprenant N valeurs.

Cette transformation a pour principal intérêt de décorréler les coefficients et mène, pour les images naturelles, à un regroupement de l'énergie autour des coefficients transformés de basse fréquence. Ainsi la plupart des coefficients transformés de haute fréquence sont nuls ou très faibles, comme illustré Figure 2. La DCT est la meilleure approximation de la transformée de Karhunen-Loève (KLT) dans le cas d'une image naturelle de taille 8x8 pixels [Lee, 2004].

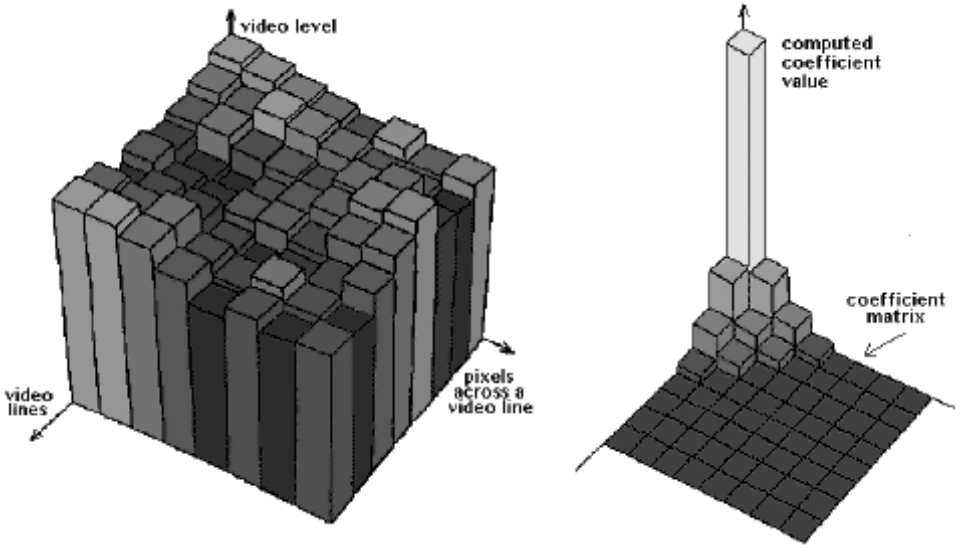


Figure 2 – Concentration de l'énergie par transformation DCT

(source [Lee, 2004])

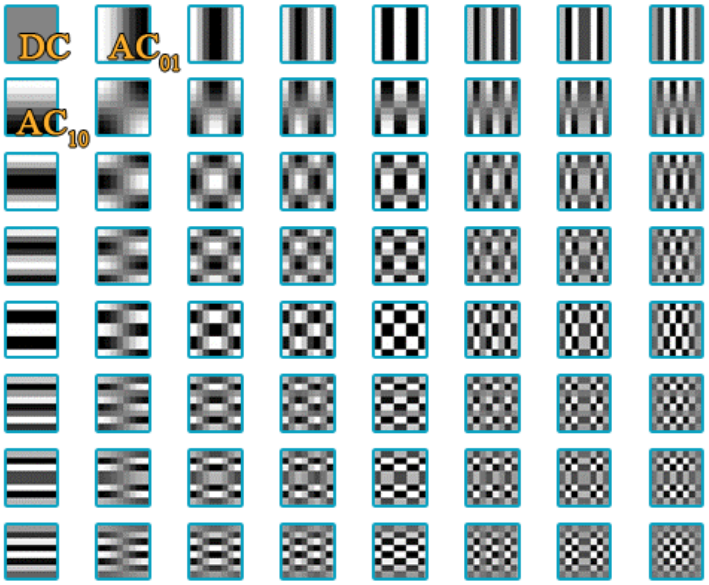


Figure 3 - Coefficients DCT : Correspondance domaine spatial / fréquentiel

Alors que le domaine spatial, sur le graphe de gauche Figure 2, présente des valeurs conséquentes pour chacun des 64 pixels du bloc, le passage dans le domaine fréquentiel, sur le graphe de droite, a permis de grouper le contenu de l'information sur une dizaine de coefficients DCT.
La Figure 3 montre, pour chaque coefficient d'un bloc 8x8, l'image correspondant dans le domaine spatial. Ainsi le premier coefficient, appelé coefficient DC, correspond à une valeur uniforme sur l'ensemble du bloc (moyenne de la valeur des pixels), puis la fréquence augmente horizontalement et verticalement lorsque l'on se déplace dans cette matrice, jusqu'au coefficient $AC_{77,}$ de fréquence maximale dans les deux directions, représentatif d'un damier.
Selon les standards, cette transformée est accompagnée d'une quantification des coefficients : Pour augmenter la compression, la réponse du système visuel humain est prise en compte. Ce dernier est moins sensible à certains types d'information, qui peuvent donc être atténués voire supprimés sans pertes de qualité ressentie, ce que l'on appelle le codage psychovisuel. C'est entre autre le cas des hautes fréquences spatiales, correspondant à de petits détails ou des zones fortement texturées sur l'ensemble d'une scène perçue, qui peuvent dans les cas extrêmes être assimilées comme des à-plats : A une certaine distance, un tableau impressionniste ne laissera pas paraître les nombreuses pointes de couleurs qui le composent. La sensibilité à la fréquence spatiale, liée à l'acuité visuelle, est ainsi prise en compte lors de la compression. Ainsi, sur l'ensemble des coefficients transformés, on pourra s'autoriser une approximation plus ou moins forte selon l'emplacement dans le bloc. Le coefficient DC sera peu modifié, ainsi que ceux qui le jouxtent, et plus l'on se déplace vers le coin inférieur droit du bloc transformé, moins les coefficients seront traités précisément, entraînant une moins bonne approximation.

Concrètement, chaque coefficient sera divisé par un pas de quantification donné par une matrice de quantification W comme celle présentée Figure 4 et par la formule suivante :

$$QAC_{i,j} = \frac{16 * AC_{i,j}}{Quant * W_{i,j}}$$

où $QAC_{i,j}$ est le coefficient quantifié, $AC_{i,j}$ le coefficient transformé avant quantification, *Quant* est le pas de quantification, (paramètre du codeur, variant de 1 à 31 pour MPEG-2) et $W_{i,j}$ le coefficient de la matrice de quantification Figure 4.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	34	38	46	56	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

Figure 4 – Une matrice type de quantification MPEG-2 (W)

Les premiers coefficients AC seront ici divisé par 16, jusque 83 pour le dernier coefficient, et stockés ainsi dans le fichier ou flux vidéo par ordre de fréquence croissante (avec différents

ordonnancement possibles). Lors du décodage, la même matrice de quantification est réutilisée pour re-multiplier chaque coefficient. L'approximation tient alors aux erreurs générées par la division d'entiers avec arrondi à l'inférieur.
Le coefficient DC est traité différemment, selon le nombre de bits de précision qui lui sont dédiés (8 à 11 en MPEG-2). Il est ainsi initialement calculé sur 11 bits, puis divisé, au maximum par 8, pour s'adapter à la précision souhaitée. Il ne subit par ailleurs pas l'impact du pas de quantification *Quant* utilisé pour les coefficients AC.
Ainsi, les coefficients correspondant aux plus hautes fréquences ont encore davantage de probabilité de se retrouver à zéro. Ceci sera utilisé lors du codage du bloc dans son ensemble : les valeurs ne sont pas lues ligne par ligne mais dans un ordre de fréquence croissante (balayage en zig-zag ou *alternate scan* préférable pour les images entrelacées), ce qui aura pour effet de grouper ces coefficients nuls en fin de code. Au lieu de les écrire un à un, ils sont décrits comme un groupe de zéros en donnant leur nombre. Cet artifice permet de gagner encore en compression. Les étapes suivantes de codage entropique, arithmétique, etc. n'ayant pas d'impact sur les présents travaux ne seront pas décrits.
Une image est donc ainsi codée, permettant d'achever une première compression via l'atténuation sinon la suppression de la redondance spatiale, et peut être reconstruite directement à partir du flux. Mais la principale source de redondance au sein d'une séquence vidéo provient de la troisième dimension introduite par le temps.

1.2.1.2 La redondance temporelle

Au sein d'une séquence vidéo, deux images successives sont généralement très proches l'une de l'autre (en dehors des plans de coupe par exemple). Le deuxième principe permettant de compresser fortement la taille d'un flux vidéo numérique utilise cette similitude et vise à ne coder que ce qui est nouveau d'une image à l'autre. Les images codées comme décrit dans le paragraphe précédent (1.2.1.1) sont dites images Intra, ou I, puisqu'elles peuvent être décodées entièrement intrinsèquement, sans éléments supplémentaires. Des images de ce type sont insérées régulièrement dans le flux, permettant de retrouver une référence en cas d'erreur lors de la transmission réseau ou de la lecture du fichier. Entre ces Intra, des images sont reconstruites à partir de ces images références.
Au niveau du codeur vidéo, l'image à reconstruire est découpée en blocs, comme précédemment. Sur la figure ci-dessous, l'image de gauche représente une image référence, et celle de droite celle que l'on souhaite reconstruire. Les blocs jaunes sont inchangés, on peut les recopier directement depuis l'image source. En revanche, les blocs bleus sont bien présents dans l'image de gauche, mais à une position différente. Enfin, les blocs rouges n'étaient pas présents auparavant et devront être entièrement codés dans le flux.

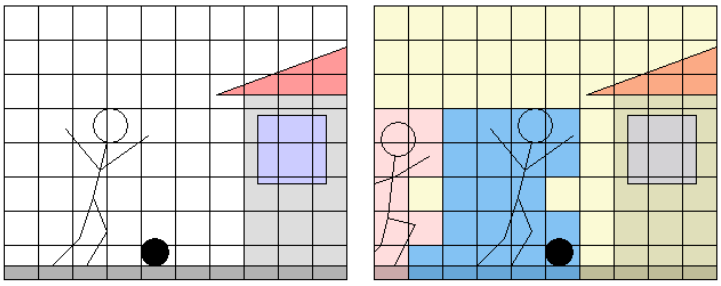


Figure 5 - Différents types de prédiction de blocs

Pour chacun des blocs à prédire (jaunes et bleus), l'information qui sera stockée est un vecteur, dit vecteur d'estimation de mouvement (ou *motion vector* en anglais, abrégé MV). Concrètement, pour chacun de ces blocs, le bloc source de l'image de référence qui correspond le mieux à celui de l'image courante va être cherché, selon un critère de minimisation d'erreur. Différents algorithmes de recherche sont utilisés selon les codeurs et leurs paramètres, incluant la *full search*, pour laquelle chaque bloc candidat sera testé, la *three steps search*, *log search* ou *diamond search*, qui ne testent que quelques candidats dans une zone réduite autour de la position d'origine [Asefi et Dabbagh, 2006 ; He et Liou., 1997 ; Hoi-Ming et al., 2005 ; Li et al., 1994 ; Ma et Hosur, 2000 ; Meng et Li, 2009 ; Tourapis et al., 2000]. Ces approches sont beaucoup moins coûteuses que la recherche exhaustive en temps de calculs (13 candidats à vérifier contre 121 pour une fenêtre de 11x11 pixels autour du centre du bloc) mais peuvent ne pas être optimales selon le contenu de l'image.
Pour chacun des blocs candidats, une erreur est calculée. Encore une fois différents critères coexistent : *MAD*, *SAD*, *MSE*, *SSE*… Tous calculent une valeur reflétant le taux d'erreur entre le bloc courant et chaque bloc candidat. Le candidat avec le plus faible taux sera retenu comme bloc source. Un vecteur d'estimation de mouvement est alors codé, déterminant le déplacement (x,y) de sa position d'origine dans l'image source vers celle du bloc courant, comme illustré ci-dessous.

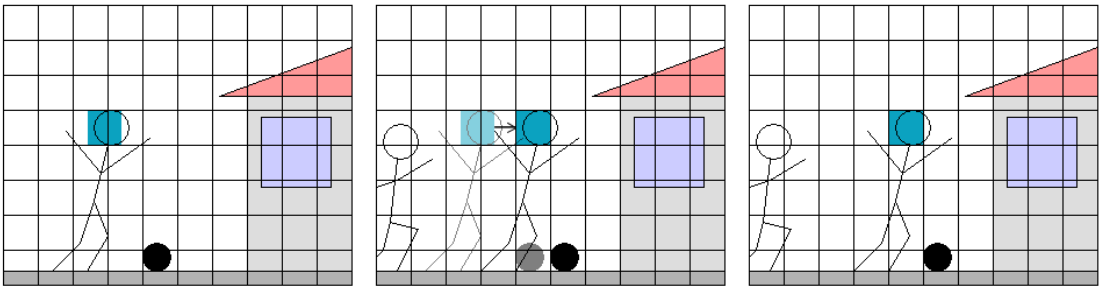


Image référence | Images superposées | Image courante

Figure 6 - Détermination d'un vecteur d'estimation de mouvement

Pour chaque bloc, une source est ainsi déterminée, permettant une première reconstruction de l'image. Toutefois, le choix du bloc source étant déterminé par minimisation d'une erreur, il se peut que cette dernière soit non nulle, et qu'une simple approximation du bloc soit en fait reconstruite. Dans ce cas, une erreur résiduelle, ou résidu, est calculée par soustraction entre le bloc réel et le bloc reconstruit, comme illustré Figure 7. Ce résidu est généralement faible, et sera codé par transformée comme détaillé dans la partie 1.2.1.1. Toutefois, si l'erreur minimale est au delà d'un certain seuil, paramètre du codeur, l'utilisation d'un bloc Intra est choisi : ce bloc, bien qu'appartenant à une image prédite, sera codé comme ceux des images Intra. C'est le cas sur l'illustration Figure 5 des blocs rouges, qui correspondent à une information nouvelle dans l'image (une autre personne entre dans le champ).

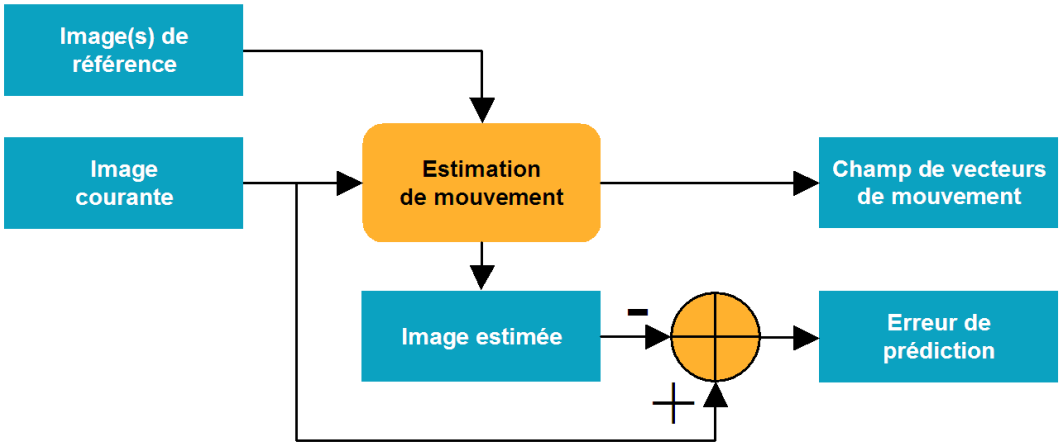


Figure 7 - Estimation de mouvement et calcul de résidus

On peut ainsi reconstruire une image à partir d'une image de référence déjà décodée. Mais selon les standards, plusieurs références sont parfois possibles, situées avant ou après dans la séquence vidéo. Dans ce cas, les images reconstruites à partir d'images de référence passées sont dites prédites, notées P, et celles reconstruites à partir d'images passées et/ou à venir sont dites bidirectionnelles, ou B. Des sous séquences décodables indépendamment, également appelées GoP pour *Group of Pictures*, sont créées. Elles contiennent une image Intra I, suivies d'images prédites P. Les images I et P servent d'images de références pour les images P qui les suivent. Enfin, un codage bidirectionnel des images B est utilisé pour les images entres ces I et P. Chaque bloc composant l'image peut alors être reconstruit à partir d'une image I ou P précédent ou suivant la B courante, voire des deux avec une interpolation entre les deux blocs sources. Selon les paramètres du codeur, le nombre de P et B par GoP est variable. Couramment, un GoP contient 12 images, comme suit : I B B P B B P B B P B B. La première image, I, est codée complètement. Ensuite on code la 4ème image, P, à partir de la I. Alors les images B 2 et 3 sont codées. On recommence alors avec la 7ème image P, prédite à partir de la 4ème, puis les B entre les deux, etc. On obtient au final une différence d'ordre entre la transmission ou le stockage du flux et son affichage, comme le précise la Figure 8. Ce phénomène introduit potentiellement une latence au codage/décodage du fait de la différence d'ordonnancement, qui demande la mise en mémoire tampon de quelques images le temps de coder/décoder les images intermédiaires.
D'autres niveaux de décomposition du flux sont également possibles selon les standards et les besoins de l'utilisateur (Figure 9). Ainsi le *slice*, ou portion d'image, représente une sous partie de l'image (souvent un tiers ou la moitié). Il permet par exemple en cas de problème de transmission d'une image Intra de pouvoir malgré tout décoder une partie de celle-ci, et donc de reconstruire par la suite une partie du GoP suivant. Le macrobloc quant à lui représente un ensemble de blocs, couvrant une surface de 16x16 pixels. Il est utilisé à des fins différentes selon les standards et sera donc détaillé lors de la revue de chacune des approches.

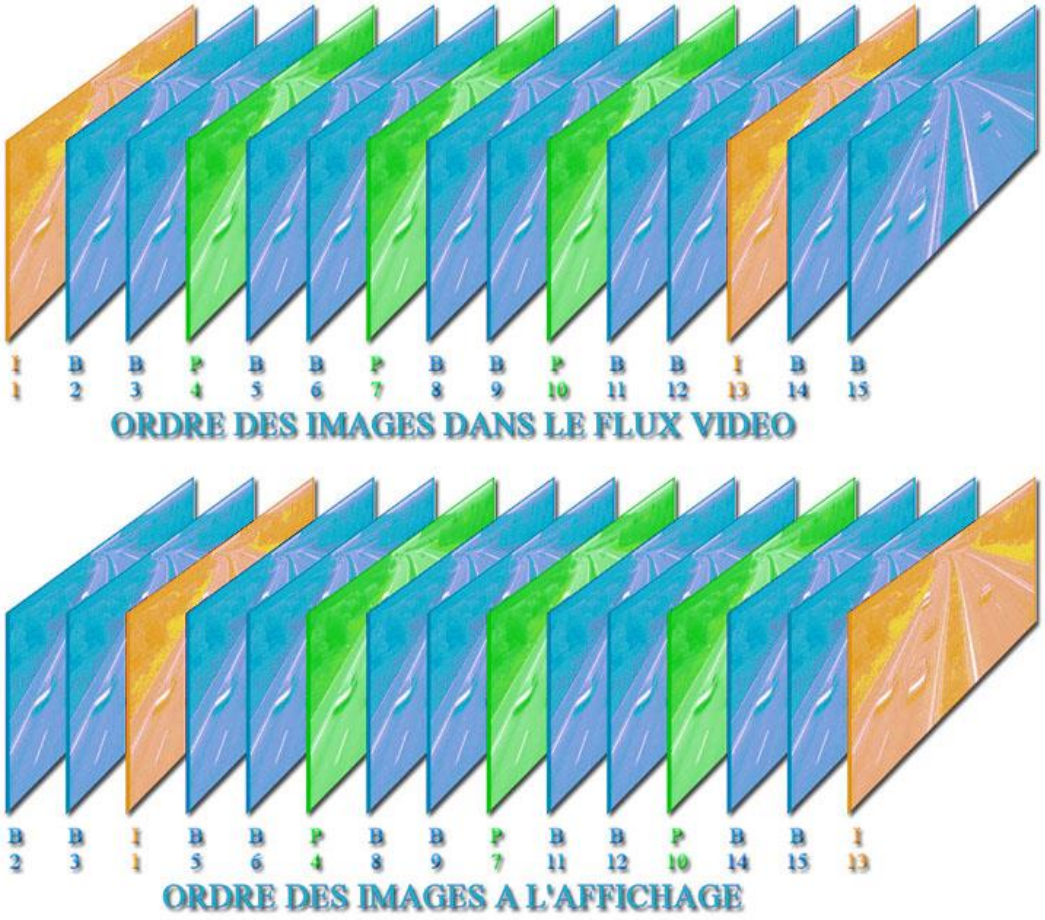


Figure 8 - Ordonnancement des images, dans le flux vidéo et à l'affichage

Remarque : dans cette figure, les B 2 et 3 ont besoin de l'image référence du GoP précédent pour être décodées.

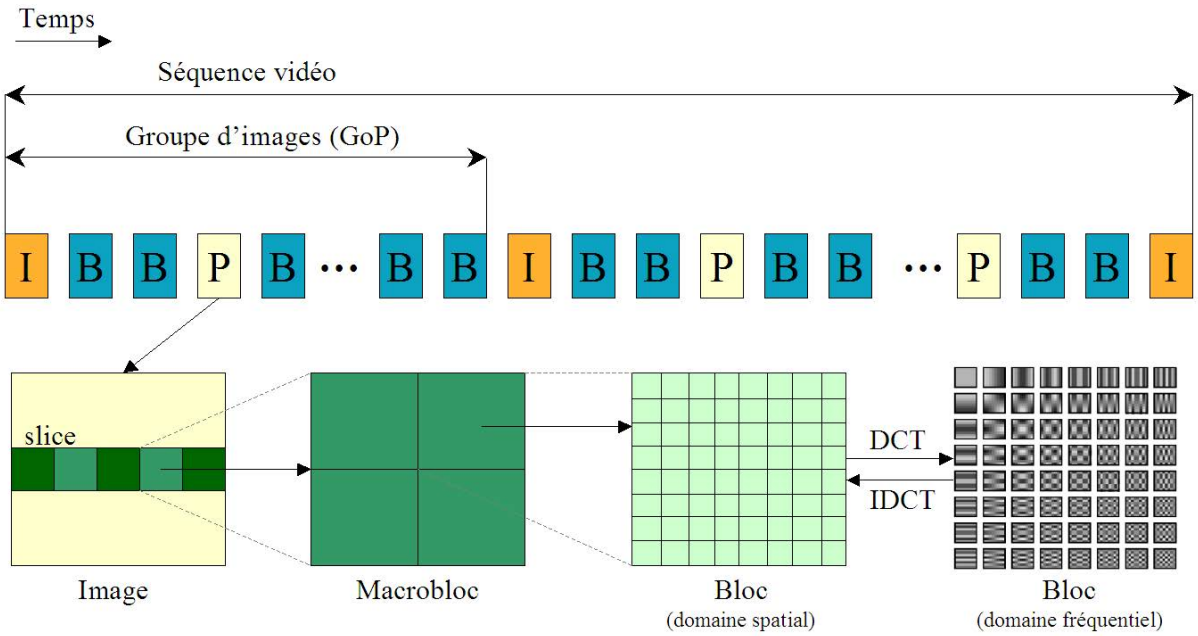


Figure 9 - Décomposition d'une séquence vidéo : du GoP au bloc

1.2.2 MPEG-2

MPEG-1 est né en 1988 d'un groupe d'experts de l'ISO/CEI, le *Motion Picture Expert Group*, ou MPEG, visant à définir un standard de codage pour les contenus cinématographiques numériques. Bien que jetant les bases qui seront reprises par les futurs codecs lors de sa publication en 1993, il ne sera pas détaillé ici puisque n'étant plus représentatif des solutions de vidéosurveillance.
MPEG-2, conjoint avec le standard ITU H.262 pour la partie 2, date de 1994, même s'il fut publié en 1995, et visait à standardiser la compression vidéo et audio, ainsi que le transport des flux générés. La partie vidéo (MPEG-2 Part 2) s'appuie sur la DCT, utilisée en codage avec perte pour les utilisations autour de la surveillance. La compensation de mouvement est également employée afin de réduire la redondance temporelle, à partir d'image prédite P et bidirectionnelles B. Les images I et P peuvent être utilisées comme image de référence pour l'estimation de mouvement, mais pas les B. L'estimation se fait par ailleurs pour chaque macrobloc, ce qui signifie qu'un vecteur de mouvement est déterminé pour quatre blocs et non individuellement. En réalité le dernier profile permet une prédiction pour deux blocs et non quatre (16x8 ou 8x16) mais n'est presque jamais utilisé d'une manière générale, encore moins en vidéosurveillance.
En cas d'erreur résiduelle trop importante sur un bloc, tout le macrobloc est codé en Intra. Les vecteurs d'estimation de mouvement sont quant à eux au demi-pixel près, par interpolation de chaque pixel du bloc prédit à partir des quatre pixels sources recouverts.
Contrairement à MPEG-1 les tailles possibles de vidéo ne sont plus fixes ou multiples de 16, mais restent limitées à une résolution de 1920x1152 (*High Level*). Le support de séquences entrelacées est également nouveau. Le codage des coefficients s'appuie sur un code RLE (*Run-Length encoding*) permettant de compresser les zéros de chaque bloc en haute fréquence, puis un code de Huffman permet de réduire la suite de coefficients générée.

Différents profils sont prédéfinis dans la norme, dont principalement deux utilisés en vidéosurveillance (SP et MP). Divers niveaux sont également présents, mais pour des questions de débits et taille d'image, la surveillance ne fait appel qu'aux LL et ML.

Tableau 1 – Profils MPEG-2

<table>
<tr><th></th><th>Profil</th><th>Types d'image</th><th>Format chromatique</th><th>Ratios de pixel</th><th>Mode scalable</th></tr>
<tr><td>SP</td><td>Simple</td><td>I,P</td><td rowspan="3">4 :2 :0</td><td rowspan="5">Carré,
4 :3 ou 16 :9</td><td>Aucun</td></tr>
<tr><td>MP</td><td>Main</td><td rowspan="3">I,P,B</td><td>Aucun</td></tr>
<tr><td>SNR</td><td>SNR scalable</td><td>Scalabilité en SNR</td></tr>
<tr><td>Spatial</td><td>Spacially scalable</td><td>4:2:2 ou 4:2:0</td><td>Scalabilité en SNR ou spatiale</td></tr>
<tr><td>HP</td><td>High</td><td></td><td>Scalabilité en SNR ou spatiale</td></tr>
<tr><td>4:2:2P</td><td>Pro</td><td>I,P</td><td>4:2:2</td><td>Aucun</td></tr>
</table>

Tableau 2 - Niveaux MPEG-2

	Niveau	Images par seconde	Nombre max. de colonnes	Nombres max de lignes	Débit max (MP) en Mb/s
LL	Low Level	23.976, 24, 25, 29.97, 30	352	288	4
ML	Main Level		720	576	15
H-14	High 1440	23.976, 24, 25, 29.97, 30, 50, 59.94, 60	1440	1152	60
HL	High Level		1920	1152	80

1.2.3 MPEG-4 Part 2

Apparu en 1998, MPEG-4 est également connu en tant que standard ISO/IEC 14496. Vingt-cinq parties le composent à ce jour, depuis les descriptions systèmes (Part 1) jusqu'à la représentation symbolique de musique (Part 23) en passant par un environnement d'animation 3D (AFX Part 16), et bien sûr la compression audio et vidéo. Pour plus d'informations sans entrer dans les détails de chaque partie, [WG11, 2002a] décrit globalement le standard, alors que [WG11, 2002b] propose une description globale du standard par des industriels, comprenant ses forces et faiblesses aussi bien technologiques que commerciales.

MPEG-4 Part 2 (ISO/IEC 14496-2) est la première proposition de compression vidéo présente dans le standard, souvent simplement appelée MPEG-4 lorsque le contexte suffit à déterminer que l'on parle de compression vidéo. Il s'appuie une fois de plus sur une transformée DCT, et une estimation de mouvement, cette fois-ci avec une précision au quart de pixel. Un vecteur est attribué par défaut à chaque macrobloc, mais il est également possible d'en déterminer un par bloc. Les vecteurs sont codés en différentiels, c'est à dire que pour connaître la valeur d'un vecteur au décodage, on utilise ceux précédemment décodés dans l'image (au dessus, au dessus à droite, à gauche) pour estimer la valeur courante (par valeur médiane) puis on ajoute le résidu codé dans le flux.

La partie 2 propose au total environ 21 profils et 4 niveaux [Pereira et Nunes, 2001]. Toutefois, seuls deux sont couramment utilisés, surtout en ce qui concerne la vidéosurveillance : les *simple profile* (SP) et *advance simple profile* (ASP, très proche du profil *Advance Real Time Simple* prévu pour une utilisation sur plate-forme embarquée). Le premier est une restriction conséquente des options proposées par MPEG-4 Part 2, le rendant peu coûteux en termes de temps de calculs mais avec des ratios de compression peu élevés, principalement du fait de la restriction à des images progressives et l'absence de trames B. Il est implanté dans de nombreux appareils portables comme les téléphones portables, les "lecteurs MP4" ou autre appareil embarqué à faible puissance. Les nombreux profils et niveaux permettent d'adapter le codage selon le type d'applications et de contenus, avec par exemple une scalabilité en texture très efficace dans le cas d'animations ou de séquences de synthèse.

Une compensation de mouvement global, ou GMC, est introduite, permettant d'obtenir une meilleure compression lors de travellings de caméra par l'utilisation d'un nombre réduit de vecteurs d'estimation de mouvement pour l'ensemble de l'image (1 à 3 selon les modes).

Au moment de la standardisation, MPEG-4 Part 2 apportait un gain d'environ 60% par rapport à MPEG-2. Mais celui-ci étant énormément utilisé, principalement à travers le DVD, de nombreuses recherches ont continué et continuent encore sur le sujet, et ont permis de diviser encore par deux le débit en MPEG-2 à qualité équivalente (entre 1998 et 2001, source [Koenen, 2003, planches 39-40]). Ceci explique qu'avec un gain final de l'ordre de 30% sur MPEG-2, le multimédia grand public aura attendu son successeur avant de renouveler tout le

matériel audio-visuel. En revanche, MPEG-2 proposait un rapport qualité/débit un peu trop faible pour de nombreuses applications de type vidéosurveillance. Les gains apportés à son arrivée par MPEG-4 Part 2 ont donc permis d'atteindre des ratios jugés meilleurs et de nombreuses sociétés proposant des solutions de sécurité ont développé des produits à partir de la nouvelle norme. Par ailleurs, l'adaptation à la diffusion au réseau (RFC 3016, RFC 3640 et MPEG-4 Part 8) a été fortement simplifié par l'arrivée de MPEG-4. Tout cela explique l'importance beaucoup plus significative du standard dans ce type de secteur.

Tableau 3- Profils MPEG-4 Part 2 utilisés en vidéosurveillance

	Profil	Type d'image	Précision des MV	GMC	Entrelacement	Quantification type MPEG	Résolutions typiques	Débits max. (kb/s)
SP	Simple	I,P	1 pixel	Non	Non	Non	QCIF, CIF, SD	768
ASP	Advance Simple	I,P,B	¼ pixel	Oui	Oui	Oui	CIF, SD	4000
MP	Main	I,P,B	¼ pixel	Oui	Oui	Oui	CIF, SD, 1920x1088	38400

1.2.4 MPEG-4 Part 10 / H.264 AVC

Le manque de gain entre MPEG-2 et MPEG-4 Part 2, couplé aux travaux lancés de son coté par l'ITU à travers H.26L, a motivé l'adjonction d'une nouvelle partie à la norme, une fois de plus consacrée à la compression vidéo. Nouvelle collaboration (JVT) entre l'ITU et MPEG, elle porte les noms de ITU-T H.264 ou ISO MPEG-4 part 10 / AVC, et est régulièrement désignée par H.264 ou AVC (par opposition à SVC, décrit en 1.2.5). La première version date de mai 2003, et les évolutions se sont stabilisées depuis décembre 2004 même si des améliorations sortent de temps à autre via des profils et extensions.

Outre l'objectif d'améliorer le ratio qualité / compression, H.264 vise à fournir une solution pour des supports (acquisition, transmission et diffusion) et des applications variés (streaming vidéo sur Internet ou téléphones portables, home cinéma, surveillance, etc.). Il comprend 16 profils et différents niveaux par profil (14 pour le main par exemple).

Deux principales ruptures se dégagent par rapport aux précédents standards : le changement de transformée et d'estimation de mouvement. La DCT a laissé la place à la transformée entière (*Integer Transform*) sur des blocs 4x4 ou 8x8 pixels. Bien que proche structurellement de la DCT, elle permet des calculs sur entiers sans approximations, accélérant les temps de traitement et diminuant les pertes liées aux arrondis (les résidus sont plus pertinents). Le basculement entre les transformées 4x4 et 8x8 peut être automatique selon les codeurs, permettant de s'adapter au mieux au contenu.

La prédiction se fait toujours au quart de pixel, avec un codage différentiel des vecteurs, selon le même algorithme que pour MPEG-4 Part 2. Elle est maintenant multi références. Ceci signifie que jusqu'à 16 images précédemment décodées peuvent être utilisées pour reconstruire l'image courante. Concrètement, les études montrent des gains important pour 2 références (4 sur les images B), mais peu d'améliorations au delà compte tenu du temps de calcul [Wiegand et al., 2008]. Ces références peuvent être à diverses positions dans le GoP, et non plus simplement juste avant ou après, permettant des optimisations comme illustré Figure

10. Les portions 8x8 de l'image utilisent toutefois les mêmes références pour éviter de perdre en référencement de blocs sources ce qui est gagné par le découpage plus fin de l'image. Enfin, ces références multiples peuvent être pondérées pour une compression optimale même en cas de fondu par exemple.

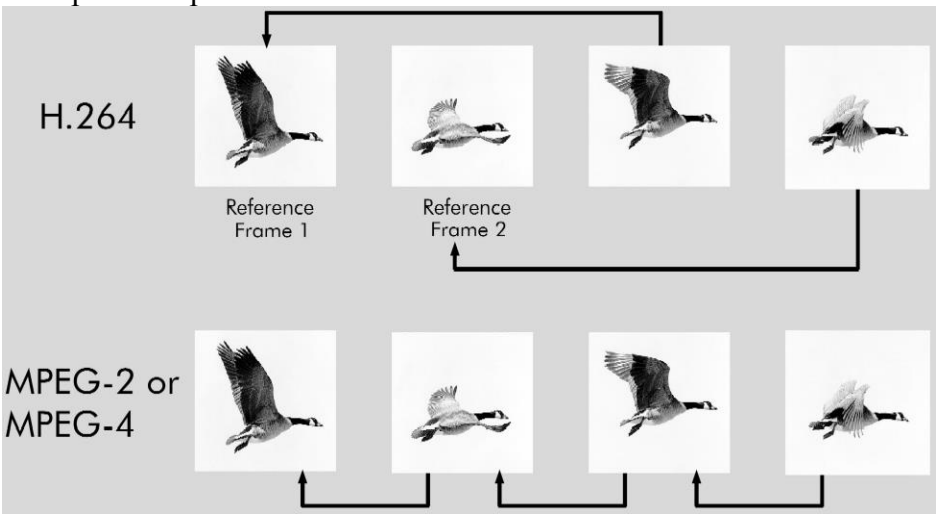


Figure 10 – Prédiction multi références en H.264/AVC

(source [Mehaoua 2006])

Par ailleurs, l'estimation de mouvement a lieu sur des blocs de tailles variables, 4x4, 4x8, 8x4, 8x8, 8x16, 16x8 et 16x16, s'adaptant une fois de plus au contenu. Chaque macrobloc contient donc potentiellement plusieurs vecteurs d'estimation de mouvement, jusqu'à 32 dans le cas d'une image B.

La prédiction intra est ajoutée. Elle permet de coder un bloc non plus à partir d'une image référence passée ou future, mais à partir des bords des blocs voisins. La dernière ligne et/ou colonne des blocs du dessus/de gauche sont utilisées comme première approximation avant calcul du résidu. Cette stratégie est utilisable sur tout type d'image (I, P, B) et accroît une fois de plus la compression.

La reconstruction des images peut s'accompagner d'un filtre de boucle (*in-loop deblocking filter*). Celui-ci permet d'atténuer les artefacts de transition d'un bloc à l'autre souvent reprochés aux précédents standards MPEG.

Les modes scalables de MPEG-2 et MPEG-4 (en qualité et spatial) ont disparu. Non utilisés dans les profils SP et MP de MPEG-2 et SP et ASP de MPEG-4 Part 2 en vidéosurveillance, ils sont remplacés par une scalabilité temporelle, accessible grâce à la numérotation des images. Selon le codage du flux, on pourra visionner la même séquence à 50, 25, 12 ou 6 images par seconde par exemple.

Le codage entropique de type VLC n'est conservé que pour des éléments syntaxiques (codage Exponential-Golomb), et est ailleurs remplacé par un codage adaptatif selon le contexte CABAC ou CAVLC.

De nouvelles structures de partitionnement de l'image sont ajoutées, notamment l'ordonnancement flexible des macroblocs ou FMO et l'ordonnancement arbitraire des *slices* ou ASO qui permettent de grouper les macroblocs selon les besoins du codeur. Elles ne sont cependant présentes que dans les profils de base (BP) et étendu (*Extended Profile*, non utilisé en vidéosurveillance).

Tableau 4 - Profils H.264 utilisés en vidéosurveillance

	Profil	Types d'image	FMO	ASO	Redundant slices	Support de l'entrelacé	CABAC	Transformée adaptative 4x4 / 8x8
BP	Baseline	I,P	Oui	Oui	Oui	Non	Non	Non
MP	Main	I,P,B	Non	Non	Non	Oui	Oui	Non

1.2.5 MPEG-4 Part 10 / H.264 SVC

Scalable Video Coding, ou SVC, est une extension de H.264 AVC, apparu en tant qu'annexe G en novembre 2007. Il s'appuie sur les différentes notions de scalabilités introduites dans les standards précédents, et se justifie par la multiplication des plateformes de visualisation et le changement d'habitude des utilisateurs finaux. Une même source peut être visualisée sur un téléphone portable ou une télévision Full HD, sur un assistant personnel (PDA, *Personnal Digital Assistant*) lors de la patrouille d'un agent ou un moniteur dans un central de sécurité. Un unique flux pourra donc s'adapter selon les capacités du réseau, celles des puces de décodage, ou encore à la résolution de l'écran utilisé.
Quatre scalabilités sont gérées par SVC :
- la scalabilité spatiale, grâce à laquelle un même flux peut être visualisé à des résolutions différentes. La couche basse présentera ainsi soit une version basse résolution de la séquence, soit une sous fenêtre pleine résolution (les deux joueurs autour du ballon lors d'un match de handball par exemple), soit une solution intermédiaire (un plan plus serré qu'initialement avec les acteurs lors d'un dialogue, à une résolution intermédiaire par exemple). Dans tous les cas, cette couche basse servira de première estimation de la séquence des couches hautes, ou couche de raffinement (Figure 11), évitant de transmettre plusieurs flux comprenant chacun des représentations complètes de la scène.

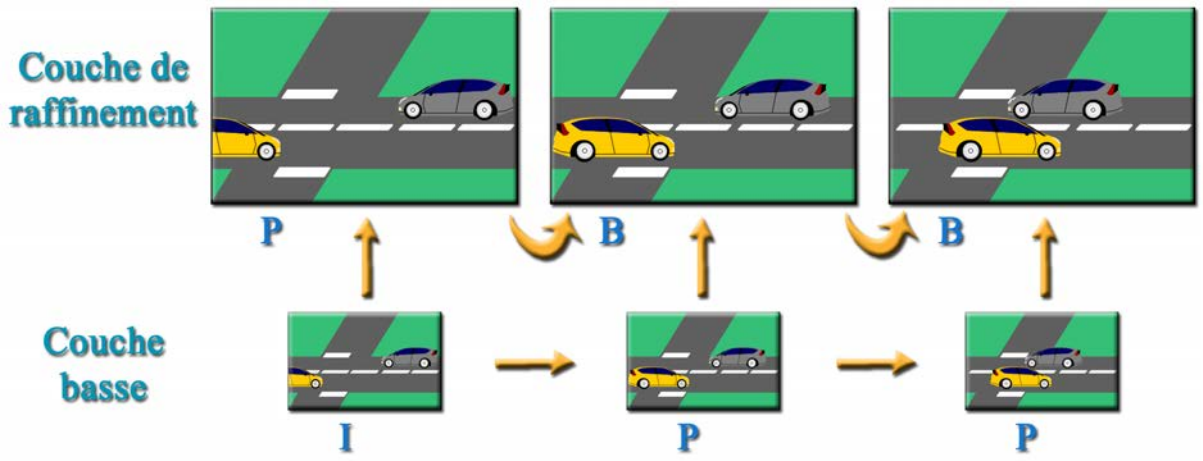


Figure 11 – Scalabilité spatiale avec SVC

- la scalabilité temporelle, permettant d'adapter le nombre d'images par seconde, pour la visualisation du flux selon le moyen de visualisation ou pour l'optimisation au contenu de la scène. Ainsi de par ses capacités de calculs réduites, un appareil portable pourra afficher 12 images par seconde quant un moniteur en reproduira 60 ; pour une scène de vidéosurveillance sans activité (parking vide, etc.) 1 image par seconde peut suffire, alors qu'en cas d'affluence

(le même parking à la sortie d'un match de foot), il faudra accélérer le rafraîchissement à 25 ou 30 images par secondes pour distinguer plus de détails. Une séquence avec un faible taux de rafraîchissement constitue la couche basse, à laquelle vient s'ajouter des images intermédiaires supplémentaires codées en couche(s) de raffinement (Figure 12).

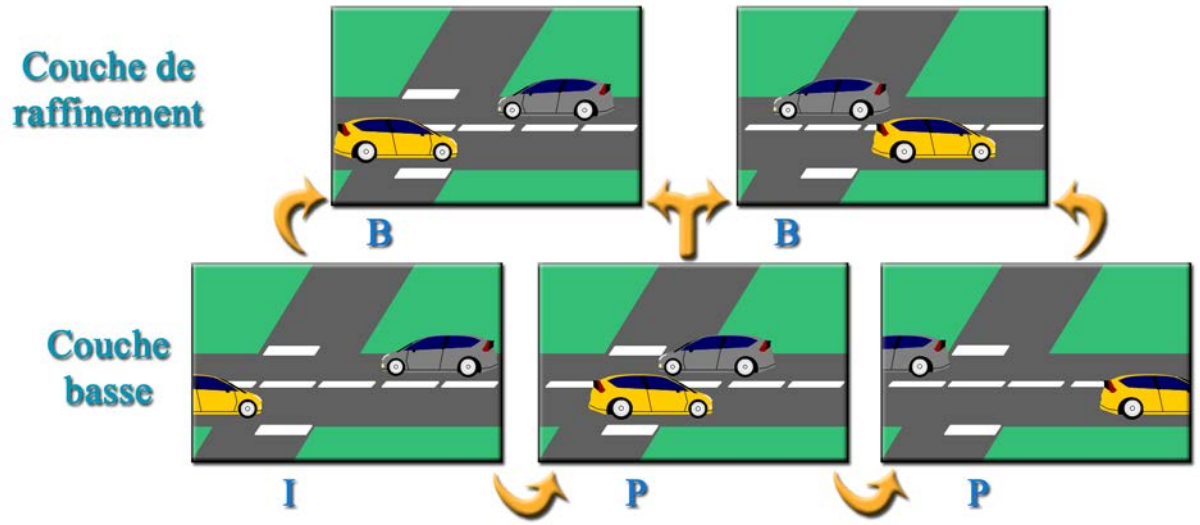


Figure 12 – Scalabilité temporelle avec SVC

- la scalabilité en qualité, scalabilité SNR ou encore scalabilité en fidélité, qui permet en utilisant la même résolution d'augmenter la qualité perçue au travers des détails. Ceci peut par exemple être atteint en utilisant des pas de quantification différents d'une couche à l'autre, les pas les plus faibles étant réservés aux couches les plus élevées (Figure 13).

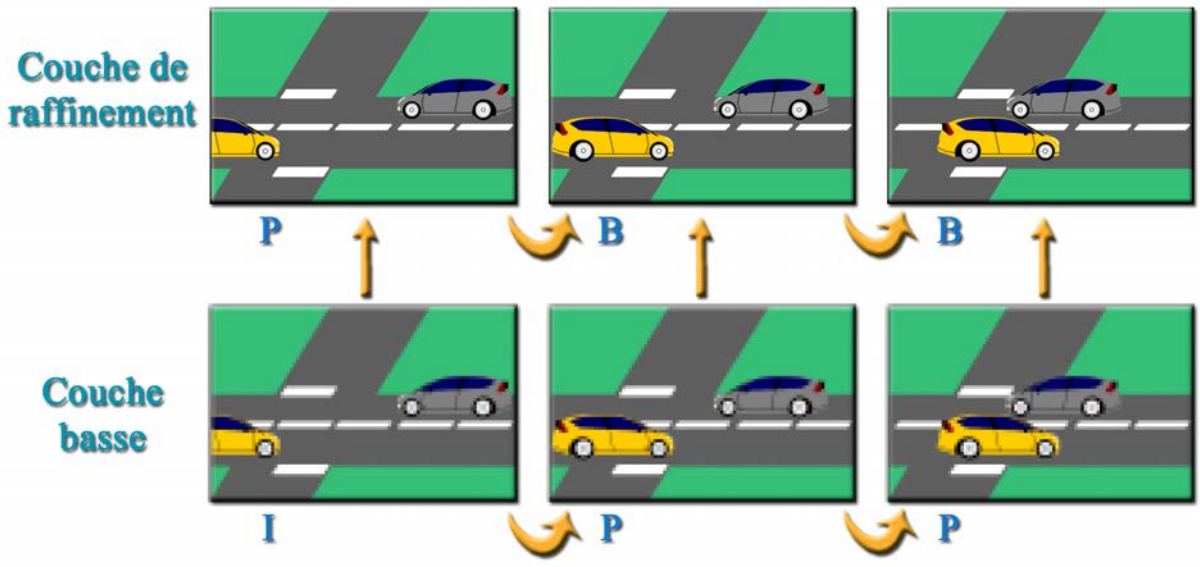


Figure 13 – Scalabilité en qualité avec SVC

- une combinaison de ces trois scalabilités, avec une ou plusieurs couches de raffinement, permettant de s'adapter à un maximum de cas de figures.

Bien qu'étant encore en cours développement, au travers de nouveaux profils notamment, SVC fait l'objet d'un nombre croissant de publications. Pour un complément d'information,

[Schwarz et al., 2007 ; Ebrahimi et Horne, 2000] en proposent une introduction assez détaillée. En tant qu'extension de H.264 AVC, les profils de SVC ont été ajoutés à ceux déjà présents dans AVC. Sont ainsi apparus le scalable baseline profile, le high profile et le high intra profile (Tableau 5). Dans le cas du SHIP, les images IDR, ou *Instantaneous Decoder Refresh*, sont les seules types codés dans le flux vidéo. Ce sont des images Intra décodables intégralement indépendamment les unes des autres (même les paramètres de décodage communs d'une image à l'autre sont redondants), et correspondent à une réponse face à Motion JPEG 2000 (cf. 1.2.6) dans le monde multimédia professionnel (TV, Cinéma, etc.) car offrant une qualité optimale au prix de ratios de compression moins importants.

Tableau 5 - Profils H.264 SVC

	Profil	Types d'image	Scalabilité spatiale	Scalabilité temporelle	Scalabilité en qualité	Support de l'entrelacé	CABAC	Autre
SBP	Scalable Baseline	I,P,B	Oui, avec un ratio entre 1.5 et 2 dans chaque direction	Oui	Oui	Non	Oui	Couche basse compatible AVC
SHP	Scalable High	I,P,B	Oui	Oui	Oui	Oui	Oui	Couche basse compatible AVC
SHIP	Scalable High Intra	IDR	Oui	Oui	Oui	Oui	Oui	Couche basse compatible AVC avec IDR

1.2.6 Motion JPEG 2000

Motion JPEG 2000 (ISO/IEC 15444-3, IUT T.802), ou Motion J2K voire MJ2K, précise l'utilisation de la norme de compression d'images fixes JPEG 2000 pour des séquences vidéo (ainsi que du format de fichier final). Tout comme pour SVC SHIP détaillé ci-dessus, les images sont codées individuellement les unes des autres, sans prendre en compte la redondance temporelle.
JPEG 2000 est un standard de compression d'images fixes (ISO/IEC 15444-1 et ISO/IEC 15444-2) basé sur la transformée en ondelettes discrètes (*Discrete Wavelet Transform* ou DWT), créé par le comité JPEG en 2000. Cette transformée permet une décomposition multi résolutions de l'image, tout en permettant d'atteindre des facteurs de compression importants grâce à son fort pouvoir de décorrélation et sa capacité à compacter l'énergie sur les premiers coefficients transformés. Ces coefficients, une fois quantifiés, sont groupés en sous bandes (Figure 14). Elles permettent un accès multiple à l'information, autorisant un décodage à des résolutions différentes selon les besoins de l'utilisateur. [Christopoulos et al., 2000] propose un premier contact avec JPEG 2000.

Figure 14 - Exemple de décomposition dyadique en sous bandes de l'image test Barbara

(Source [Christopoulos et al., 2000])

La troisième partie de la norme comporte le format de fichier décrivant le stockage d'une vidéo codée en MJ2K ainsi que la description des profils et les fichiers de conformités. Ceux-ci incluent des séquences de vidéosurveillance utilisant les régions d'intérêts fournies par UCL (Université Catholique de Louvain) dans le cadre du projet WCAM

Parmi les atouts de Motion JPEG 2000, pour la vidéosurveillance entre autre, figurent la représentation multi résolutions, le codage des régions d'intérêts (RoI), ainsi que la résilience aux erreurs. Le codage des RoI est rendu possible par les multiples accès au flux, qui autorisent des codages différents, en termes de compression avec pertes par exemple, de différentes zones spatiales de l'image définie lors du codage. Dans l'absolu, la limitation en résolution est celle de JPEG 2000, soit 4 milliards de pixels de côté (2^{32}-1). Toutefois les usages en vidéosurveillance, principalement pour des raisons de complexités et de capacités de calcul, se contentent de résolution SD (720x576 voire 720x288 avec une source entrelacée – une trame sur deux est enregistrée).

JPEG 2000 est aujourd'hui le format de compression retenu par l'industrie du cinéma à travers les standards SMPTE, ainsi que par le monde de l'imagerie médicale [Pearson et Gill, 2005], dans lequel la vidéo se développe surtout depuis l'apparition de l'imagerie à résonance magnétique fonctionnelle (IRMf) qui fournit une visualisation dynamique des tissus. Enfin, certaines solutions propriétaires de vidéosurveillance, comme celle d'Avigilon, s'appuient également sur le standard.

1.2.7 Synthèse sur les formats utilisés en vidéosurveillance

D'une manière générale, le monde de la vidéosurveillance adopte les standards successifs, avec un délai de 4 à 5 ans avant les mises sur le marché imputable au niveau de maturité des solutions chez les industriels. Ainsi H.264 (AVC) a commencé à être proposé significativement en 2008, avec un vrai éventail de solutions commerciales en 2009/2010, alors que SVC n'est aujourd'hui présent que pour de rares démonstrateurs [Aventura].

Les implantations évoluent à mesure que la maîtrise des standards augmente. A sa sortie, AVC était supposé nécessiter jusqu'à soixante fois plus de ressources que MPEG-4 Part 2 [Goldman 2005]. Or il est communément admis aujourd'hui qu'un facteur 10 à 20 est réaliste. Dans tous les cas, la complexité croissante prend également en compte les évolutions matérielles, qui permettent de porter aujourd'hui sur une unique puce (DaVinci - DM6446 de Texas Instrument par exemple) un algorithme de compression AVC. De plus, ces chiffres de complexité sont avancés à qualité équivalente de la séquence décodée, ce qui s'accompagne d'un débit réduit entre AVC et MPEG-4 en faveur du dernier standard en date.

Les débits considérés en vidéosurveillance sont généralement beaucoup plus faibles que ce que les profils et niveaux sélectionnés autorisent (Tableau 7). Ainsi pour un flux AVC en résolution standard (720x576), les objectifs sont entre 1,5 et 3Mb/s. Il faut toutefois considérer l'application finale, et éviter de ne trop compresser au risque de perdre trop en qualité et ainsi manquer de détails pour l'analyse future des flux. En France, l'Arrêté du 3 août 2007 portant définition des normes techniques des systèmes de vidéosurveillance [JO, 2007a ; JO, 2007b] préconise ainsi au minimum les débits présentés dans le Tableau 6 pour des séquences à 12 images par seconde. Le domaine de la vidéosurveillance se voit également sujet à des normes pour les métadonnées (*Surveillance Application Format*) ou pour l'interopérabilité des flux provenant de différentes sources (agences de sécurités, municipalités, etc.) [ISO/TC 223/AH 3, 2009].

Tableau 6 - Débits recommandés par décret pour la vidéosurveillance (source [JO, 2007b])

TYPE DE MECANISME DE COMPRESSION	DEBIT THEORIQUE MOYEN pour disposer d'images au format 4CIF à 12 images par seconde
JPEG	5 Mbits/s
JPEG 2000	3 Mbits/s
MPEG 2	2 Mbits/s
MPEG 4	1 Mbits/s
MPEG 4 (H 264)	0,5 Mbits/s

Aujourd'hui SVC sort des laboratoires académiques pour ceux des industriels comme Thales. Toutefois il doit encore faire ses preuves, car beaucoup préfèrent transmettre deux flux AVC aux résolutions adaptées plutôt qu'un unique flux SVC; qui peut s'avérer requérir plus de débit que les deux flux. La multiplication de plateformes de visualisation hétérogènes (moniteur de sécurité, PDA, téléphone portable, radio tactique, etc.) pourrait s'avérer être un allié pour asseoir le statut de cette extension. Toutefois en l'état, une implantation de SVC comprenant de multiples couches de raffinement avec les trois types de scalabilités représente une complexité telle qu'elle n'est pas réaliste pour une solution temps réel, et n'aurait pas d'application directe.

Dans ce panel de solution pour la vidéosurveillance, Motion JPEG 2000 promettait initialement des qualités supérieures à AVC, ce qui pouvait être pertinent dans le cas d'infrastructures critiques où il peut être intéressant de noter de nombreux détails (personne jetant un mégot en forêt, transmission discrète d'objets dans un aéroport, etc.). Toutefois, des études [Ouaret et al., 2006 ; Topiwala et al., 2006 ; Boxin et al., 2008] ont montré que JPEG 2000 et AVC, même à des résolutions supérieures à celles actuellement utilisées en vidéosurveillance aboutissent à des résultats comparables en termes de qualité pour des débits équivalents, ce en n'utilisant que des images Intra pour H.264. Toutefois, la tendance reste en faveur de JPEG 2000 pour les hautes résolutions, comme pour les solutions proposées par Avigilon.

L'avenir de la compression pour la vidéosurveillance semble s'orienter vers des solutions de moins en moins coûteuses en termes de capteurs (coûts matériel, consommation électrique, etc.). L'arrivée de codeurs distribués, tels que DVC - *Distributed Video Coding* [Weerakkody et al., 2007] – permettra de déporter la plupart des calculs vers les outils de visualisation des flux générés. Si cette solution n'est pas nécessairement adaptée, par exemple, à la démocratisation de la vidéo sur téléphone portable, elle est particulièrement pertinente dans le cas de la sécurité ou les sources vidéo sont souvent limitées en puissance et autonomie, alors que les serveurs sont souvent des ordinateurs avec la puissance nécessaire.

L'un des points forts de chacun des standards de compression vidéo est avant tout le nombre de profils et niveaux disponibles, même s'ils peuvent générer beaucoup de confusion, notamment chez des groupes se lançant pour la première fois dans ces normes. Chacune des utilisations possibles trouve une configuration optimale. Si l'information est représentée via différents types d'images (I, P, B), pour des problèmes de complexité et de capacité mémoires restreintes sur les appareils embarqués tels que les caméras de surveillance, les codeurs utilisés reposent principalement sur des images I et P, avec des horizons temporels de référencement de 1 (seule l'image précédente sert de référence), ce même en H.264.
L'estimation de mouvement permet de généralement prendre en compte le déplacement d'objet dans le champ de vision. Certains cas restent problématiques : d'une part les objets de trop petite taille (moins d'un bloc ou macrobloc) peuvent être codés par des résidus, si la portion de fond immédiatement derrière prévaut au niveau de l'impact sur la compression. D'autre part, si les normes autorisent des amplitudes de vecteurs très conséquentes (-2048 à +2047,75 pixels en horizontal pour H.264), les limitations viennent souvent des fenêtres de recherches pour l'estimation de mouvement, liées au codeur. Dès lors, si un bloc se déplace trop vite d'une image à l'autre, il sera le plus souvent codé comme un bloc Intra, même sur une image prédite.
D'une manière générale, plus les normes proposent une compression efficace, plus le flux vidéo compressé résultant s'éloigne de l'information source (mouvement réel d'objet, etc.). Ainsi la taille 4x4 des blocs de H.264 mène souvent à une estimation de mouvement sur de simples portions d'objet et non sur l'objet lui-même. En décorrélant de plus en plus l'information source et le résultat de la compression, on augmente le pouvoir de compactage des différents codeurs entropiques. Ces derniers sont maintenant adaptatifs et changent de tables dynamiquement selon le contenu à coder. Le temps dévolu à la récupération et la mise en forme des informations avant décompression par les parseurs est de plus en plus important, de même que pour effectuer les transformées inverses et les compensations de mouvement.
De nouveaux outils sont disponibles, tels que le FMO ou les MBA Map, qui permettent de structurer le flux selon l'importance des différentes zones de l'image, autorisant ainsi par exemple une modélisation des RoI que seul JPEG 2000 fournissait auparavant.

Le Tableau 7 page 40 propose un récapitulatif des différences entre chacun des standards précédemment présentés. MV y signifie "vecteur estimation de mouvement", VS "vidéosurveillance", (VS) précise les paramètres atteignables avec les profils généralement utilisés en vidéosurveillance. Picture AFF et MB AFF correspondent à des modes de basculement automatiques d'images progressives vers entrelacées et inversement (selon les contenus par exemple lors d'une transmission télévisuelle).

Ainsi, que ce soit pour des problématiques de débits sur les réseaux ou de stockage, les flux de vidéosurveillance sont compressés. Dans l'optique d'automatiser l'analyse de ces nombreuses sources vidéo, beaucoup d'algorithmes ont vu le jour, principalement par des études au niveau pixellique. Mais une tendance s'est également dégagée, utilisant l'information de ces flux sans les décompresser au préalable.

Tableau 7 - Récapitulatif des formats de compression pour la vidéosurveillance

	MPEG-2	**MPEG-4 Part 2**	**MPEG-4 Part 10 AVC**	**MPEG-4 Part 10 SVC**	**Motion J2K**
Publication	1995	1998	2003	2007	2000
Types d'images (VS)	I, P, B (I,P)	I, P, B	I, P, B	I, P, B	I
Taille de bloc	8x8	8x8	4x4 ou 8x8	4x4 ou 8x8	N/A
Taille des blocs par MV	16x16	16x16, 16x8, 8x16, 8x8	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4	N/A
Précision max. des MV	½ pixel	¼ pixel	¼ pixel	¼ pixel	N/A
Codage des MV	Indépendants les uns des autres	Différentiel	Différentiel	Différentiel ou prédit par scalabilité	N/A
Transformée (VS)	DCT	DCT	Entière (4x4)	Entière (4x4)	Ondelettes
Prédiction Intra image	Aucune, sauf la prédiction possibles des cœfficients DC	Aucune, sauf la prédiction possibles des cœfficients DC	- 4x4 ou 16x16 spatial - I_PCM	- 4x4 ou 16x16 spatial - I_PCM	N/A
Types de codage d'images	- Image - Trame - Picture AFF	- Image - Trame - Picture AFF	- Image - Trame - Picture AFF - MB AFF	- Image - Trame - Picture AFF - MB AFF	Image
Filtres de boucle	Non	Non	Oui	Oui	N/A
RoI	Non	Non	Possible (par Slices Group, MBA Map …)	Possible (diverses approches)	Oui
Résolution max. (VS)	1920x1152 (720x576)	4096x2304 (1024x768)	4096x2304 (1440x1080)	4096x2304 (1600x1200)	4Gp x 4Gp (1600x1200)
Prédiction des P	Référence simple	Référence simple	Références multiples	Références multiples	N/A
Prédiction des B	1 à 2 références	1 à 2 références	Références multiples (pondérables)	Références multiples (pondérables)	N/A
Débit max. (VS) Mb/s	80 (15)	38 (15)	240 (15)	240 (20)	(20)
Scalabilité (VS)	SNR ou spatiale (aucune)	Temporelle, qualité ou spatiale (aucune)	Temporelle	Temporelle, qualité ou spatiale	Qualité et spatiale
Codage entropique	VLC	VLC	CAVLC et CABAC	CAVLC et CABAC	Adaptative Binary Arithmetic encoder

1.3 Méthodes d'analyse pour les flux vidéo compressés

Pour faire face au volume croissant de données générées par la multiplication des solutions de vidéosurveillance déployées, de nombreux travaux ce sont intéressés aux outils d'analyse automatique de vidéo. La plupart des approches s'appuient sur les données au niveau du pixel, depuis la détection d'activité jusqu'aux résultats plus haut niveau caractérisant des comportements. Toutefois, quelques travaux utilisent directement l'information contenue dans les flux vidéo compressés pour obtenir des rapports d'analyse avec différents niveaux de complexité.

1.3.1 L'analyse en vidéosurveillance

Définir et proposer des solutions de vidéosurveillance intelligente, de par l'analyse automatique du contenu vidéo, a suscité et suscite encore de nombreux travaux. L'objectif de cette section est d'exposer différentes approches couramment utilisées en vidéosurveillance, afin de présenter ensuite en comparaison les solutions existantes s'appuyant sur des outils dans le domaine compressé. Le domaine de la surveillance implique bien souvent la nécessité d'utiliser des algorithmes relativement rapides (capables au moins de suivre le temps réel de préférence, même si ce n'est pas toujours le cas), excluant de nombreuses méthodes ayant fait leur preuves dans des cas complexes mais trop gourmandes en ressources. La Figure 15 présente un exemple de chaîne de traitement pour la vidéosurveillance permettant de lever des alarmes selon des événements préalablement appris, et permet d'illustrer les différents éléments qui vont être présentés par la suite.

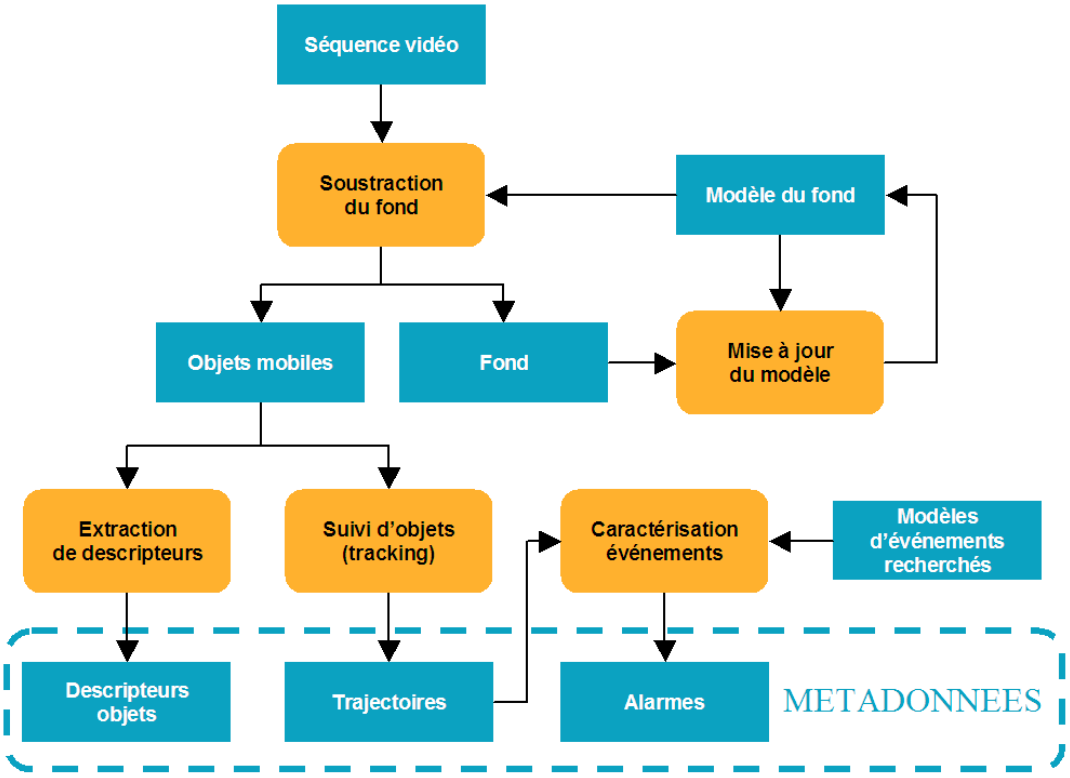


Figure 15 - Exemple de chaîne de traitement pour la vidéosurveillance

L'application considérée comme la plus simple en surveillance est la détection d'activité, sans précision supplémentaire. Ce système le plus basique peut être directement intégré à la caméra comme Multitel le propose depuis une dizaine d'années, maintenant enrichi par d'autres informations de plus haut niveau [Desurmont et al., 2005]. Une seule sortie est offerte

sous la forme d'un booléen : « il y a actuellement une quelconque activité dans le champ de vision » (1) ou « rien ne se passe » (0).
Différentes méthodes permettent d'arriver à ce résultat bas niveau. L'une d'elles consiste à effectuer un apprentissage du fond, au cours duquel une image représentative du contenu perçue en période d'inactivité est enregistrée lors d'une phase préalable. L'implantation la plus simple se contentera de calculer une valeur moyenne pour chaque pixel. Pour déterminer les zones d'activité dans l'image courante, une soustraction est alors effectuée entre cet instantané et l'image moyenne servant de référence. Si cette différence est en dessous d'un seuil, paramètre du système, le pixel est actif. Un post filtrage par morphologie mathématique par exemple permet de supprimer les pixels isolés et de déterminer si des zones de tailles suffisantes ont été détectées. Si tel est le cas, le booléen résultant indiquera la présence d'une activité.
Un premier niveau de raffinement consiste à modéliser le fond par une valeur moyenne complétée par un écart type. Le seuillage se fait alors par rapport à cet écart type avec une marge opérationnelle faible permettant d'éviter les fausses alarmes (algorithme de gauche Figure 16).
Ces outils, qui ont initialement été développés et pensé pour des applications sur ordinateurs fixes [Haritaoglu et al, 1998 ; Wren et al., 1997 ; Toyama et al., 1999], peuvent maintenant être embarqués directement à l'intérieur des capteurs CMOS de caméras ou d'appareils photos [Verdant et al., 2007]. Des produits à la frontière de la robotique, la domotique et la surveillance arrivent également sur le marché depuis quelques années. Ainsi Rovio de Wowee [Rovio, 2008] ou Spykee de Meccano [Spykee, 2007] sont des robots équipés de caméra (webcam), qui peuvent être programmés pour faire des rondes et ayant une détection d'activité intégrée. Un instantané peut alors automatiquement être envoyé par mail au propriétaire.

L'utilisation d'outils de classification ont par la suite permis d'apporter des résultats plus robustes, notamment face aux problèmes liés aux environnements non maîtrisés. La modélisation par moyenne et écart type peut fonctionner en éclairage constant, avec une caméra parfaitement fixe, comme dans un couloir de métro ou un parking souterrain. Mais dès lors que la séquence à analyser s'étale sur plusieurs périodes de la journée voir de l'année, ou encore que la caméra se retrouve en haut d'un mât subissant les aléas du vent, les outils les plus simples ne permettent plus d'obtenir de détection cohérente. Pour enrichir ces deux valeurs, des modélisations plus complexes pour chaque pixel sont maintenant utilisées. Les mélanges de gaussiennes ou GMM (pour *Gaussian Mixture Model*, algorithme de droite Figure 16) permettent de prendre en compte un environnement variable, pouvant intégrer différentes valeurs pour chaque pixel, contenues dans les gaussiennes déterminées lors de l'apprentissage [Huang et al., 2009 ; Haque et al., 2008 ; Sofka, 2008 ; Zhang et Chen, 2007 ; Jodoin et al., 2006]. Plus récemment, les machines à vecteurs de support, parfois appelées séparateurs à vastes marges pour reprendre l'acronyme anglais SVM (*Support Vector Machine*) [Sheng et al., 2009 ; Fan et al., 2008 ; Santiago-Mozos et al., 2003] ont fait une percée dans le monde du traitement vidéo. Malgré un coût en terme de temps de calcul très conséquent, les SVM permettent d'apporter des solutions tantôt comparables tantôt complémentaires aux GMM. Ces deux approches permettent de modéliser un environnement non parfaitement fixe, pouvant par exemple prendre en compte correctement les pixels au niveau d'un arbre bercé par le vent, où le vert des feuilles et le bleu du ciel s'entremêlent, ou encore des séquences à l'illumination changeante, par modélisation avec une fenêtre temporelle très longue (sur plusieurs jours) ou par utilisation d'une mise à jour avec un facteur d'oubli autorisant des modifications lentes telles que la tombée de la nuit sans générer une segmentation de toute l'image. Des situations qui avec les premiers outils soit déclenchaient systématiquement des alarmes, soit généraient des non détections par un écart type trop grand,

telles qu'un feu tricolore, une publicité déroulante, une zone éclairée par un projecteur tournant ou un escalator peuvent être efficacement modélisées et interprétées.

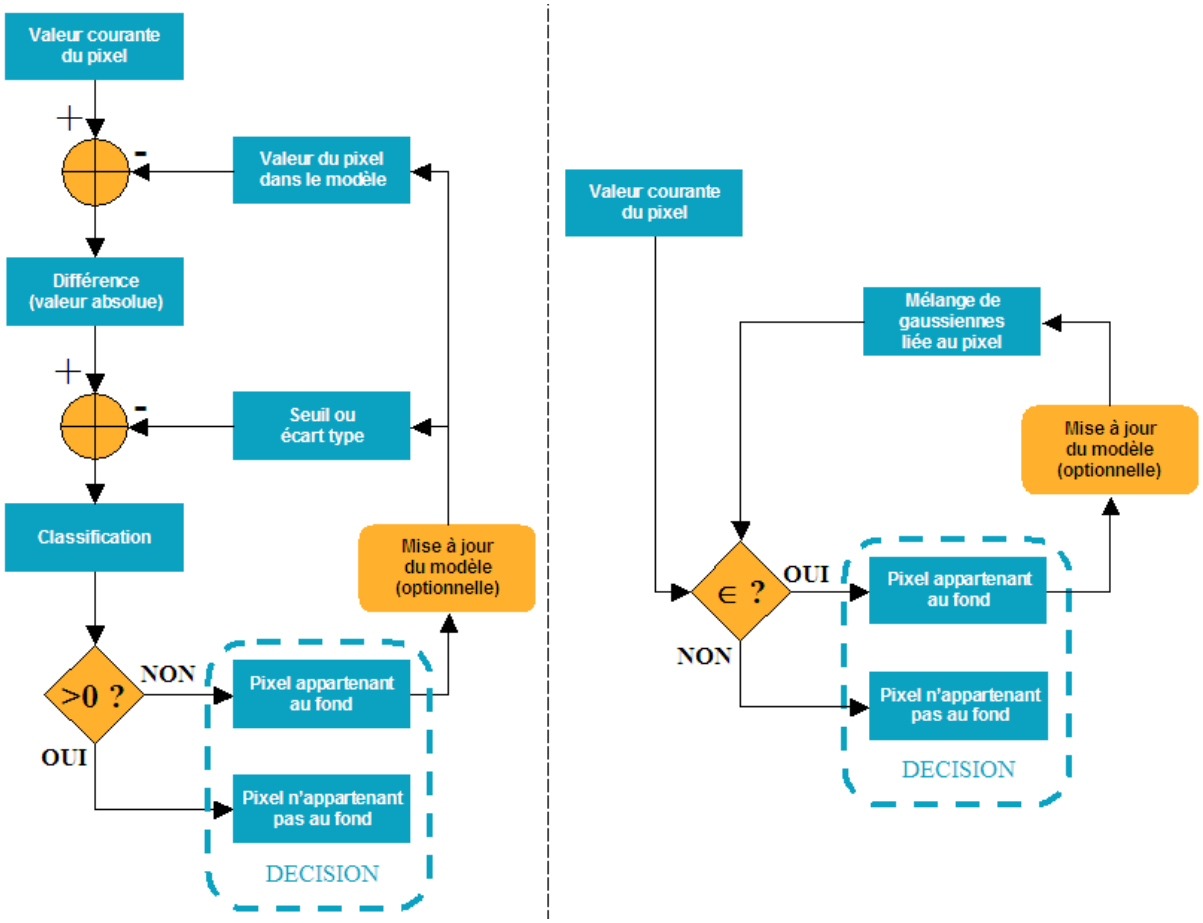


Figure 16 – Méthode de soustraction du fond

(à gauche, par moyenne/écart type ; à droite, par mélange de gaussiennes)

Un autre complément apporté par la suite a permis de prendre en compte les caméras mobiles type PTZ (*Pan, Tilt, Zoom*) [Bevilacqua et Azzari, 2006 ; Bo et al., 2006]. Du fait de leur possible mouvement, modéliser un fond fixe n'est plus suffisant. Si différentes possibilités permettent de prendre en compte ce problème, la plus couramment employée consiste à considérer une représentation de l'ensemble de la scène couverte par la caméra PTZ, dont on ne voit alors plus qu'une zone restreinte à chaque instant. La fonction de zoom n'est pour sa part généralement pas gérée de la même façon. Puisqu'elle est manipulée par une commande extérieure (opérateur humain ou premier élément de détection en mode grand angle), la modélisation du fond se retrouve débrayée lorsqu'un zoom est effectué. Les zones mobiles et/ou d'intérêt ont alors été identifiées avant le changement de focale. La mise à jour du modèle du fond demande d'identifier efficacement la zone couverte en instantané par rapport au champ élargi. Pour cela, un suivi de points saillants ou points d'intérêt peut être utilisé, permettant un recalage précis (jusqu'à une précision subpixellique). Ces points saillant peuvent être sélectionnés manuellement lors de la calibration de la caméra ou automatiquement, par l'utilisation par exemple de points SIFT [Battiato et al., 2007]. Leur suivi est par la suite réalisé par des méthodes similaires à celles de suivi d'objets (abordées page suivante). La prise en compte des caméras PTZ peut également être mise en place via une autre approche, couplant une caméra fixe grand angle à cette caméra mobile [Zhou et al., 2003]. Dans ce cas, l'identification de la zone d'activité est faite sur la caméra fixe, et une

calibration préalable du couple de capteurs permet de déterminer, par rapport aux coordonnées dans le plan image de la caméra fixe, la position à donner à la caméra PTZ pour se focaliser sur l'action en cours.

Figure 17 - Champ de vision par rapport à la couverture d'une caméra PTZ

Ces différentes méthodes traitent le flux vidéo au niveau de chaque pixel, ce qui peut être très difficilement atteignable voir impossible à mettre en œuvre sur des plateformes aux ressources limitées. Pour cela, des parades ont été développées, depuis le sous échantillonnage spatial [Kompatsiaris et al., 2000] permettant une approche multidimensionnelle de la segmentation, ou un sous échantillonnage temporel ne prenant en compte qu'une image par seconde par exemple [Kompatsiaris et al., 2000 ; Alexiadis et Sergiadis, 2008]. Si ces artifices s'avèrent pertinents pour des levées d'information bas niveau comme dans le cas présent, dès lors que l'on souhaite effectuer un suivi d'objet précis ou caractériser un comportement en détails, il sera nécessaire de traiter l'information sur plusieurs niveau [Rong et al., 2002], le dernier étant généralement au même niveau de précision que les méthodes sur tous les pixels de toutes les images.

Que ce soit par une méthode simple ou plus évoluée, la détection d'activité permet en sortie de fournir une réponse binaire liée à la présence d'objets mobiles dans le champ de la caméra, sans aucun a priori sur les objets en question. Toutefois, l'une des étapes intermédiaires est constituée d'un masque binaire suite au seuillage de la différence entre le modèle et l'image courante. Si ce masque comporte des zones d'une surface représentative d'un objet, l'alarme est déclenchée. Mais ce masque peut avoir d'autres utilités, comme fournir un moyen de détection individuelle d'objet pour extraire des informations plus haut niveau.
Les masques de segmentation générés lors de la détection d'activité fournissent des instantanés successifs des zones d'activités au sein d'une séquence vidéo. Lorsque ceux-ci sont précis, il est possible d'obtenir un ensemble de pixels connexes, ou blob, segmenté pour chaque objet, sur chaque image. Les méthodes de suivi d'objets visent alors à reconstruire des trajectoires par objets à partir de ces représentations image par image. Sans entrer dans le détail de l'ensemble des outils de suivi, nous citerons les travaux de Stauffer et Grimson [Stauffer et Grimson, 1998] qui couplés aux outils d'appariement de Munkres (algorithme hongrois ou algorithme de Kuhn-Munkres) [Frank, 2004] permettent d'obtenir un suivi d'objet tout en conservant des temps de calcul autorisant le temps réel sur des plates formes informatiques puissantes. Les autres méthodes classiquement employées, parmi lesquelles le filtre de Kalmann [Kim, 2007] ou le filtrage particulaire [Arulampalam et al., 2002] sont aujourd'hui les étendards, se révèlent trop complexes pour des traitements massifs de vidéo au-delà du temps réel.
Une fois les trajectoires de chaque objet identifiées, différents algorithmes proposent une analyse haut niveau (sur le plan sémantique) d'une scène de vidéosurveillance. Par exemple, les SVM (*Support Vector* Machines, ou Machines à Support de Vecteurs) peuvent être utilisés pour déterminer des trajectoires anormales [Piciarelli et Foresti, 2007]. Après une étape d'apprentissage supervisé, le SVM permet de scinder en deux classes (de par son caractère binaire) les trajectoires des différents objets : ceux qui correspondent aux déplacements

attendus, et les autres, quels qu'ils soient. Une alarme peut ainsi être levée en cas de comportement anormal. Les études menées dans le cadre de ces travaux sur l'analyse dans le domaine compressé n'exploitant pas ce type de caractérisation haut niveau pour le moment, nous ne détaillerons pas davantage ces outils qui sont pour la plupart aujourd'hui encore en train de mûrir et de progresser dans de nombreux laboratoires universitaires, tels que l'équipe PULSAR de l'INRIA Sophia Antipolis ou celle de l'IDIAP avec qui nous avons pu travailler.

1.3.2 L'analyse dans le domaine compressé : Pourquoi ?

Initialement le traitement d'images et vidéo s'effectuait au niveau pixellique. Mais avec la démocratisation des formats de compression, et surtout du fait du volume de données croissant, différentes équipes se sont intéressés à l'utilisation directe des informations contenus dans les fichiers compressés. L'intérêt suscité a principalement été rendu possible par la possibilité de s'abstenir des transformées inverses. Par exemple, une IDCT complète requiert 4096 multiplications et 4032 additions. Des optimisations ont vu le jour, en stockant des résultats intermédiaires, mais nécessitent malgré tout 1024 multiplications et 896 additions. L'une des motivations étaient donc de s'affranchir de ces calculs.

Les premiers travaux correspondent aux années 1995-2000 sur JPEG & MPEG-1. Les résultats étaient alors souvent bas niveau, sans réelle application immédiate intéressante, mais ont permis de démontrer un potentiel dans cette démarche alors marginale, avec des résultats tels que l'extraction de carte de contour en JPEG ou la détection de personnes sur des vidéo soigneusement sélectionnées pour le peu de bruit qu'elle contenait.

Les informations extraites du domaine compressé sont en effet singulières, et ne peuvent être traitées directement que dans des cas simples. La plupart du temps, un prétraitement est requis. De fait, après ces premiers résultats ne permettant pas réellement de construire des algorithmes complexes, seuls quelques articles sont parus entre 2000 et 2005, avec de la segmentation en environnement maîtrisé d'objets, de grande taille en pixel. Beaucoup plus de publications traitent de la détection de plans de coupe pour le résumé automatique de journaux ou DVD, correspondant à des évaluations ou compétitions comme TRECVID ou PETS [TRECVID ; PETS]. Il s'agit alors de détecter des changements de caméra brutaux ou fondus, la présence de texte à l'écran, etc. Mais les outils proposés sont spécifiques à l'exercice et ne peuvent être réutilisés pour la problématique soulevée par la vidéosurveillance.

En revanche, depuis 2005, un fort regain d'intérêt transparaît dans le nombre d'articles liés à l'analyse dans le domaine compressé, entre autre de par le volume de vidéo qui explose (sites Internet communautaires ou de partage type YouTube, vidéosurveillance, multiplication des chaînes télévisuelles, etc.). Retrouver l'information devient critique, et la nécessité d'indexer sans ressources matérielles surdimensionnées se fait sentir. Le retour aux données compressées se fait avec de nouveau outils et de nouveau standards, permettant de renouveler les résultats obtenus précédemment sur les anciennes normes, mais également d'aller plus loin dans la démarche et dans la richesse des solutions.

Chaque approche a sa petite particularité qui rend difficile une classification. Mais 3 cas sont clairement identifiables : les solutions s'appuyant uniquement sur les coefficients transformés dans le domaine fréquentiel, celles utilisant seulement les vecteurs d'estimation de mouvement, et enfin les méthodes hybrides, combinant généralement coefficients transformés et vecteurs, bien que certaines variantes existent. Ce sont ces trois considérations qui seront détaillées par la suite. Nous présenterons ici les articles fondateurs de ces approches, qui ont généralement constitué une rupture technologique, ainsi que des publications parmi les plus récentes qui permettent d'avoir un aperçu du potentiel actuel de l'analyse dans le domaine compressé.

1.3.3 L'utilisation des coefficients transformés dans le domaine fréquentiel

Dès les débuts de la compression, des études ont eu lieu sur les diverses possibilités offertes par un traitement direct de ces données nouvelles. Les premiers travaux se sont intéressés aux caractéristiques qui pouvait être extraites à partir des coefficients dans le domaine fréquentiel issus de la DCT (voir 1.2.1.1). Celle-ci, bien qu'utilisée par MPEG-1, 2 et 4, étant issue de JPEG, ce sont d'images fixes que proviennent les premiers résultats.
Il est possible d'utiliser ces données pour obtenir certaines approximations, comme l'emplacement de contours. Ainsi, les coefficients AC des blocs dans le domaine compressé, permettent de reconstruire une carte de gradient à faible résolution [Shen et Sethi 96]. Ces travaux montrent entre autre que les coefficients AC_{01} et AC_{10} sont directement proportionnels aux gradients vertical et horizontal du contenu du bloc considéré. La technique considérée permet par ailleurs d'estimer l'orientation du contour dans le bloc (vertical, horizontal, diagonal), le décalage du contour par rapport au centre du bloc, ainsi que l'amplitude de la variation de luminance au niveau du contour. La Figure 18 illustre ces propriétés sur l'image « Lena ». L'image de gauche correspond à une carte de contours obtenue par un filtrage de Sobel sur l'image d'origine. Celle du milieu est un sous échantillonnage d'un facteur 8 de la première. Ce ratio vient de la taille des blocs utilisés pour la DCT, pour permettre une comparaison directe. Enfin, l'image de droite est obtenue en moyennant les coefficients AC_{01} et AC_{10} pour chaque bloc.

Figure 18 - Carte de gradient et approximation de contours dans le domaine compressé

Cette approche permet entre autre de déterminer le contenu d'un bloc compressé par la présence ou non d'un gradient vertical ou horizontal au travers principalement des coefficients AC10 et AC01. Ceci sera utilisé par les approches hybrides pour évaluer la pertinence des vecteurs de mouvements (voir 1.3.5).

Différentes méthodes permettent la recherche d'images à partir des coefficients DCT. Ainsi, en sélectionnant un échantillon de fenêtres aléatoirement entre l'image requête et l'image cible, [Shneier et Mottaleb, 1996] propose d'appairer ces fenêtres, également aléatoirement. Sur chaque fenêtre, la moyenne de chaque coefficient DCT est calculée, générant un vecteur de dimension 64. Les vecteurs des deux images sont comparés et chaque composante se voit attribué un bit (0 ou 1) selon la similarité (fixé par seuil). Un mot de 64 bit est ainsi créé pour chaque fenêtre de comparaison, et le score de similarité final est calculé à partir de l'ensemble des mots obtenus sur les images. Les résultats proposés par cette méthode sont relativement bas niveau d'un point de vue sémantique, comme la plupart des travaux publiés dans les

débuts de l'analyse dans le domaine compressé. Ainsi l'algorithme est capable de retrouver des photos d''identité parmi une collection d'images. Testé sur des images Intra extraites d'une séquence vidéo, il est capable de retrouver les plans provenant de la même caméra (fond identique). Deux limites sont toutefois présentes : si les coefficients DC sont représentatifs d'une moyenne sur un bloc et sont donc effectivement comparable sur une zone même si les blocs sont décalés de quelques pixels d'une image à l'autre, les coefficients AC sont eux beaucoup plus sensible au positionnement du bloc. Dans le cas extrême, si un groupe de 8 colonnes de pixels blancs succèdent à 8 colonnes de pixels noirs, les coefficients DC seront moyennés à un gris moyen, et les coefficients AC, nuls dans les deux cas puisque les 8 colonnes forment un bloc entier, auront une moyenne nulle également. Si les blocs sur l'image cible sont décalés de 4 pixels, le bloc du milieu aura 4 colonnes noires et 4 colonnes blanches. Le coefficient DC sera bien celui d'un gris moyen, en revanche le coefficient AC_{01}, correspondant au passage abrupt de noir vers blanc, ne sera pas nul mais aura au contraire la valeur maximale (127). Il n'est donc pas toujours cohérent de comparer l'ensemble des coefficients DCT pour déterminer la similarité entre deux images. Par ailleurs, la méthode de vectorisation de la mesure de similarité puis les multiples comparaisons mises en jeu sont coûteuses et ne permettent pas d'atteindre des temps de calculs réduits.
En exploitant des histogrammes calculés à partir des coefficients DCT, [Lay et Guan, 1999] propose une méthode alternative d'appariement d'images. Les algorithmes placent les images correspondant aux requêtes dans les cinq premiers rangs des résultats. Cela montre toutefois que l'approche, bien que parfois présentant un taux de rappel consistant sur ces cinq premiers rangs, ne permet pas de trancher directement sans validation humaine.

Les premières utilisations des informations du domaine compressé dans la vidéo se sont focalisées sur le découpage en scène d'une séquence : [Zhang et al., 1995] utilise les coefficients DCT des images Intra consécutives pour déterminer les plans de coupe dans une vidéo MPEG. Pour chaque bloc b, la somme normalisée des valeurs absolues des différences (SAD) des coefficients DCT de l'image source f_m et l'image cible f_n est calculée :

$$D(f_m, f_n, b) = \frac{1}{64} \sum_{k=1}^{64} \frac{\left| c(f_m, b, k) - c(f_n, b, k) \right|}{\max(c(f_m, b, k), c(f_n, b, k))}$$

où $c(f_m,b,k)$ est le $k^{ième}$ coefficient du bloc b de l'image f_m. Si cette différence est au delà d'un seuil, le bloc est considéré comme modifié. Si le nombre de blocs modifiées entre les deux images est plus grand qu'un certain seuil, un changement de scène est identifié entre ces deux images. De nombreuses compétitions comme TRECVID ont proposé des challenges liés à la segmentation en plan de coupe pour lesquels l'analyse dans le domaine compressé fournit des résultats probant, depuis les coupes franches jusqu'aux fondus d'un plan à l'autre.

Dans le domaine de la segmentation vidéo, [Wang et al., 2008] décrit trois méthode de modélisation du fond équivalentes au domaine pixellique s'appuyant sur les coefficients DCT. Ainsi les moyennes avec facteur d'oubli, filtrage médian (avec une fenêtre temporelle par pixel) et mélange de gaussienne sont adaptées pour traiter des coefficients du domaine fréquentiel. Les ratios en terme de faux négatifs et faux positifs sont comparables à ceux des méthodes classiques, avec des temps de calcul réduit jusqu'à un facteur cinq pour le mélange de gaussiennes (deux ailleurs).
La méthode est fiable en terme de segmentation, les outils décrits permettant de repasser au niveau du pixel sans rester à l'échelle du bloc. Parmi les inconvénients, seules les images intra peuvent être traitées, laissant des périodes classiquement autour de la seconde en vidéosurveillance sans segmentation. Selon le cas, certains objets peuvent passer totalement

inaperçus (mouvement rapide, passage très proche de la caméra, etc.), ou d'une manière plus générale cet espacement peut s'avérer pénalisant pour les étapes de suivi et levée d'alarme susceptibles de suivre. Le gain en temps de calcul n'est donc mesurable que si l'on considère une image analysée par GoP, ce qui est aujourd'hui très peu quels que soient les outils considérés.

1.3.4 L'utilisation de l'estimation de mouvement

En s'appuyant uniquement sur les vecteurs d'estimation de mouvement, il est possible de détecter un mouvement, et même de segmenter voire suivre les objets mobiles dans une séquence vidéo.
Ainsi, [Yokoyama et al., 2009] présente un algorithme de détection d'objets mobiles dans le cadre d'une caméra fixe pour des flux MPEG-4 Part 2. Les vecteurs non nuls permettent d'identifier les zones en mouvement par segmentation. Lorsqu'un objet s'arrête, la zone continue d'être étiquetée comme objet, ce qui permettra le cas échéant de reprendre la piste.
Globalement, cette méthode est très simple, et permet d'analyser un grand nombre d'image à la seconde (240 images 640x480 par seconde sur un processeur double cœur à 2.4 GHz, 2 Go de RAM). Toutefois, ces premiers résultats ne sont valables que pour le codeur et les séquences utilisées, qui génèrent un signal peu voire pas bruité. Par ailleurs, le suivi n'est pas possible sur les images Intra, et les pistes sont donc perdues à chaque nouveau groupe d'images. Les résultats en sortie sont des masques binaires de segmentation image par image, couplé à un appariement des blobs successifs (suivi partiel) lorsque cela est possible. Aucune information de taille, position ou couleur n'est extraite du flux.

Dans [Babu et Ramakrishnan, 2002 ; Babu et al., 2004], les vecteurs d'estimation de mouvement des images P et B sont tout d'abord normés selon la direction et la distance de leur image de référence. Ils sont alors accumulés sur plusieurs images consécutives, permettant d'obtenir plus qu'un vecteur par macrobloc. Lors de cette étape, seuls les vecteurs correspondant à des blocs ayant une erreur résiduelle sous un seuil paramétrable sont pris en compte, afin d'éviter de potentielles données non liées à un déplacement réel. En cas de rejet ainsi que pour les blocs Intra, une interpolation sur les vecteurs voisins est faite. L'ensemble des vecteurs nuls est alors attribué au fond, et le couplage d'un groupement par K-centroïdes (*K-means clustering*) avec un algorithme de maximisation d'espérance (EM - *Expectation Maximisation*) segmente les vecteurs restant pour obtenir les masques des différents objets mobiles. Les contours sont améliorés par un processus itératif étudiant le voisinage de chaque bloc contenant une limite d'objet pour permettre un lissage. L'objectif est en fait ici de créer les différentes couches permises par MPEG-4 Part 2 correspondant aux différents objets dans une séquence. La segmentation permet au final la structuration du flux en plan prévus par la norme : les VOP (*Video Object Plan*).
Les limitations de cette implantation proviennent principalement des choix algorithmiques pour la segmentation et le raffinement des contours : l'utilisation d'un algorithme K-means avec un EM est très pénalisante en termes de temps de calcul. L'application finale ne tient pas le temps réel, et les contours restent approximatifs malgré la phase de raffinement, qui procure plus un lissage qu'une amélioration.

La compensation du mouvement de la caméra peut se faire à partir des vecteurs d'estimation de mouvement pour reconstruire un modèle affine à 6 paramètres de ce déplacement. Il s'agit d'un modèle 3D prenant en compte translation et rotation.
Dans [Ewerth et al., 2007], une première étape de classification par voisinage et similarité permet de déterminer différents groupes de vecteurs sur des flux MPEG-1, 2 et 4 (Part 2). A partir de la classe majoritaire, le mouvement de la caméra est estimé approximativement.

Cette information est utilisée dans une phase de raffinement permettant à la fois d'obtenir un modèle plus précis du mouvement de la caméra mais également de segmenter les objets mobiles, présentant alors un champ de vecteurs différents de celui de la caméra. Les zones de l'image qui comportent ces objets sont alors décompressées pour fournir une segmentation au pixel près par modèles de contours actifs. Le suivi d'objet est réalisé en prenant en compte les vecteurs d'estimation de mouvement (avant et arrière) pour projeter l'objet courant dans l'image précédente et trouver le blob correspondant le plus proche.
Cette approche, particulièrement pertinente pour prendre en compte une caméra mobile, présente l'intérêt de proposer une première segmentation uniquement dans le domaine compressé. Par contre la segmentation affinée par contours actifs est très coûteuse en temps de calcul et ralentit considérablement l'algorithme, ce dernier tournant à peu près en temps réel (selon le nombre d'objets présents dans le champ de la caméra et la résolution de la séquence). Par ailleurs, le filtrage des vecteurs se fait uniquement suite à la première classification, par élimination des extrêmes locaux isolés. Les objets de petite taille peuvent donc passer inaperçu, et de grande zone d'à-plat présentant des vecteurs similaires pourraient à l'inverse être segmentés comme objet mobile.

Dans [Kas et Nicolas, 2009], cette estimation du mouvement global (GME) permet de différencier les blocs présentant un vecteur différent (bruit ou objet mobile) de celui associé au mouvement de la caméra. Un filtrage spatio-temporel par morphologie mathématique puis filtre médian et passe-bas est employé pour réduire les fausses alarmes et non détections, avec un horizon temporel d'un GOP (8 images ici). Le suivi est similaire à celui de [Ewerth et al., 2007], avec un appariement par plus proche voisin. Les situations de séparation et fusion (*split and merge*) sont prise en compte par un étiquetage dynamique des objets avec filiation. La suite du système permet d'obtenir des trajectoires en 2D lissées par filtrage, ainsi qu'une évaluation de la trajectoire en 3D des objets en prenant en compte leurs variations de dimension.
La même équipe propose par ailleurs des outils dédiés à la surveillance de trafic routier [Kas et al., 2009]. Un bref apprentissage du fond par GMM est alors réalisé, à partir des images Intra qui sont totalement décodées (une image sur huit). Un second modèle permet de déterminer le mouvement "normal" de la circulation par un apprentissage et modélisation par GMM également sur les vecteurs d'estimation de mouvement. Les objets mobiles sont alors segmentés sur les images prédites (B uniquement, pas de P prises en compte dans l'algorithme), et affinés par soustraction du fond, débruitage et suppressions des ombres, le tout dans le domaine pixellique sur les images Intra décompressées. Pour le suivi des objets, la sous segmentation rencontrée par la solution proposée dans le domaine compressé ne permettant pas d'obtenir des résultats concluant, elle est couplée à un appariement des objets d'intra en intra par correspondance de description par points SIFT.
Tout comme la méthode précédente, celles-ci sont représentatives des derniers travaux en date sur l'analyse dans le domaine compressé, commençant par compenser les mouvements de caméra avant de segmenter les objets mobiles. L'une des particularités de ces travaux est qu'ils s'appuient sur H.264 SVC, via lequel deux couches sont codées pour une scalabilité spatiale (d'un facteur 2). Dans les deux cas, l'estimation du mouvement global se retrouve faussée si un objet de taille trop importante se trouve dans le champ de la caméra, comme une personne, un animal ou un véhicule passant juste devant l'objectif. Dans ce cas cet objet serait étiqueté comme fond, et la segmentation résultante serait erronée (le fond devenant un objet mobile, le mouvement des objets étant déterminé par rapport à l'objet principal et non plus par rapport au fond, etc.). Enfin, les performances globales atteignent tout juste le temps réel (23 à 26.5 images par seconde) pour des séquences en 480x272 (Intel Core2Duo à 2.16 GHZ, 1 Go de RAM).

1.3.5 Les approches hybrides

En s'appuyant à la fois sur les coefficients transformés et les vecteurs d'estimation de mouvement, il devient possible de conjuguer les avantages de l'ensemble des méthodes proposées dans les paragraphes 1.3.3 et 1.3.4. *A priori*, cela permet d'offrir une segmentation des objets sur chaque image et donc d'assurer un suivi temporel total. Toutefois, les approches hybrides peuvent souffrir de complexité augmentée et donc de temps de calcul trop long pour justifier ce type de traitements.
L'un des premiers articles parus sur le sujet, [Eng et Ma, 2000], s'intéresse à un flux MPEG-1 issue d'une caméra fixe. Un filtre médian spatial permet de réduire une partie du bruit présent sur les vecteurs d'estimation de mouvement, avant une segmentation par seuillage permettant d'isoler les objets mobiles. Un système de suivi s'appuyant sur des projections temporelles des masques obtenus sur les images précédant et suivant chaque image, couplé à un filtre de Kalman, extrait les trajectoires de chaque objet. Les images Intra sont utilisées via les coefficients DC de chaque bloc, à partir desquels une segmentation par classifieur est réalisée.
Comme dans de nombreux articles s'étant intéressé à l'analyse dans le domaine compressé à ses débuts, celui-ci propose des outils permettant d'aboutir à une segmentation et un suivi des objets mobiles au sein d'une (unique) séquence vidéo. L'utilité de ces résultats n'est pas étudiée, que ce soit au travers de descripteurs ou de levées d'alarmes. L'approche floue mise en place pour la segmentation d'objets sur les images Intra est par ailleurs coûteuse en temps de calcul, et l'algorithme final ne tient pas le temps réel.

De part son adoption massive pour le support DVD, le standard MPEG-2 a fait l'objet de nombreuses études. [Manerba et al., 2008] propose d'utiliser dans un premier temps les images P d'un flux MPEG-2 pour détecter les objets mobiles. Cela est réalisé au travers d'une estimation du mouvement de la caméra puis d'une segmentation des objets mobiles après cette stabilisation. Par ailleurs, les images I sont traitées à faible résolution : pour chaque bloc, les coefficients DC sont utilisés pour reconstruire une imagette (couleur) à partir de laquelle une carte de gradient est calculée pour segmenter les objets selon les régions homogènes en couleur. En interpolant les masques obtenus pour les images P juste avant et après une image I, une correspondance entre les deux segmentations est établie, permettant d'exporter des descripteurs comportant trajectoire, taille et couleur des objets. Un lissage temporel des trajectoires est effectué, permettant de combler une partie des non détections. Toutefois ceci implique un traitement a posteriori sur un groupe d'images minimum, introduisant une latence non négligeable (les GoP variant de 1/2 à 1 seconde classiquement en vidéosurveillance). De nombreuses fonctions permettent d'affiner les résultats, telles que l'utilisation de « tubes temporels » pour le lissage ou l'amélioration de la segmentation des objets par utilisation de surfaces quadriques (2D+t).
Une fois de plus l'estimation de mouvement de la caméra étant fait sur l'ensemble de vecteurs majoritaire, elle montre ses limites dans le cas d'un objet occupant plus de la moitié du champ de vision (testé jusque 30%). Par ailleurs, l'approche dans le domaine compressé permet d'obtenir une segmentation d'objet sans décompresser, pour gagner du temps de calcul. Les différentes briques technologiques post segmentation améliorent certes le suivi et les descripteurs, mais au prix d'un temps d'exécution réduit. En effet, la méthode globale n'atteint pas le temps réel de 25 images 720x576 par seconde.

Travaillant ici encore sur un flux MPEG-2, [Lan et al., 2006] propose une méthode pour laquelle un objet est initialement sélectionné par un opérateur humain sur une image de la séquence qui sert donc de point d'entrée à l'algorithme. Les vecteurs d'estimation de mouvement formant une classe homogène connexe autour de l'objet sélectionnés servent dans un premier temps à déterminer un contour de l'objet dans le domaine compressé, i.e. à la

résolution bloc. Les vecteurs sont alors utilisés en combinaison avec les résidus sur les images P pour effectuer un suivi de l'objet tout en mettant à jour une signature dans le domaine compressé calculée à partir des coefficients DC. D'un GOP à l'autre, l'objet est recherché sur l'image I faisant la transition à travers sa signature. Si aucune détection n'a lieu, un appel à l'opérateur est lancé (alarme). Trois mesures de confiance sont déployées pour chaque bloc de chaque image P : une résiduelle, une spatiale, et une liée à la texture. La première permet de confirmer que la compensation de mouvement n'est pas faussée par un résidu trop important, ce qui pourrait impliquer une grande variation entre le bloc prédit et le bloc compensé. La seconde favorise les blocs connexes de vecteurs de mouvement, considérant que les objets suivi sont rigides. Enfin la troisième permet d'éviter d'écarter des vecteurs qui pourraient sembler aléatoires dans des zones fortement texturées. En effet, ce type de région tend à générer une estimation de mouvement bruitée, beaucoup de candidats étant éligibles pour chaque bloc à prédire. Cette méthode doit au final permettre de suivre un objet malgré sa variabilité (rotation, changement d'illumination, etc.).
L'originalité de cette approche réside avant tout dans la combinaison des trois mesures de confiance mise en place pour suivre un objet. Le parti pris de ne suivre qu'un objet permet de réduire les fenêtres de recherches, à l'intérieur du voisinage des objets et d'accélérer ainsi les traitements : 1600 images P analysées par seconde sur un cœur à 1 GHz, pour des séquences MPEG-2 en 352x288 (résolution CIF). C'est la méthode la plus rapide trouvée dans la littérature, même si ceci est au prix de certaines restrictions (1 unique objet de grande taille dans des séquences de faible résolution). Le nombre de vecteurs traités est en fait de 22x18, soit 396, par image P. Enfin, une fois le suivi effectué, l'exploitation des données n'est pas abordée, et le recours à l'opérateur pour trancher chaque cas problématique peut vite devenir bloquant quant à l'adoption d'un tel procédé pour un système réel de vidéosurveillance.

Une approche hybride originale est exposée dans [Ji et Park, 1999] : l'objectif est d'intégrer la segmentation au codeur vidéo pour éventuellement proposer des quantifications différentes selon le contenu des différentes zones de l'image. La segmentation d'initialisation a lieu sur les blocs intra, tous disponibles au niveau du codeur, et utilise les discontinuités en termes de moyenne et variance de la couleur d'un bloc à l'autre. Cette segmentation s'appuie donc concrètement sur un critère de texture. La segmentation s'appuyant sur les vecteurs d'estimation de mouvement commence par regrouper ces derniers en régions homogènes, puis les blocs sont classés par types selon leur voisinage (bloc isolé, bloc frontière, bloc intérieur, etc.) leur attribuant ainsi un poids qui servira pour la segmentation des zones mobiles.
Travaillant sur des blocs 8x8 de DCT et des vecteurs d'estimation de mouvement directement au niveau du codeur, cette approche autorise l'utilisation de plus d'information en entrée (notamment tous les blocs en intra), ce qui permet de s'appuyer sur la texture plus que sur l'estimation de mouvement, chose rare pour de l'analyse dans le domaine compressé. Cet avantage pénalise par ailleurs l'approche qui se retrouve nécessairement localisée au niveau d'un module de compression, et qui est incapable de traiter des flux déjà compressés. Par ailleurs, le filtrage qui s'effectue sur les vecteurs a lieu sur des critères spatiaux sans prendre en compte la continuité dans le temps, ce qui pénalise la détection d'objets de petites tailles (moins de 4 blocs connexes d'après la méthode de pondération des blocs).

Certaines approches combinent également l'analyse dans le domaine compressé et décompressé. Ainsi [Hsieh et al., 2008] propose dans un premier temps de décoder totalement les images I et P d'un flux MPEG-1 ou 2 pour effectuer une soustraction du fond sur ces dernières. Un remplissage des masques de segmentation, la suppression des ombres et des opérations de morphologie mathématique (ouverture – fermeture) permet d'affiner ce résultat intermédiaire, avant de mettre à jour le modèle du fond pour chaque nouvelle image. Sur les

images P, les vecteurs d'estimation de mouvement sont classés itérativement par voisinage et similarité, différenciant ainsi les différents objets mobiles. Leur suivi temporel est réalisé par l'intermédiaire des superpositions de régions d'une image à l'autre. L'exemple applicatif propose de déterminer des comportements tels que la surveillance de zone, la disparition d'objets (par assimilation au fond) ou le franchissement de mur (escalade). Ces détections sont faites soit par la présence d'une activité dans une zone de l'image (ce qui nécessite la définition de ces zones par un opérateur au préalable), soit selon les vecteurs (franchir un mur implique des vecteurs verticaux, ce qui déclenche l'alarme).
Cette solution est originale puisqu'elle utilise d'abord des données décompressées avant de s'appuyer sur les vecteurs d'estimation de mouvement. La phase de décodage est donc obligatoire sur toutes les images I et B, ce qui diminue sensiblement l'intérêt des traitements dans le domaine compressé dans le but d'accélérer les calculs. L'algorithme proposé traite des vidéo MPEG-2 en 352x240 à 12 à 13,55 images par seconde (Pentium 4 3.4GHz). Comme les images B sont ignorées, le système tient presque une fois et demie le temps réel. Le suivi de proche en proche par régions superposées est efficace en termes de temps de calcul. Toutefois si des objets se déplacent rapidement et que le second prend potentiellement la place du premier entre deux images P (cas relativement courant par exemple pour de la surveillance d'autoroute), il ne fonctionnera pas.

1.3.6 Conclusion sur l'analyse dans le domaine compressé

Si la grande majorité des outils développés dans le cadre de l'analyse automatique de vidéo se sont d'abord focalisés sur des approches au niveau pixellique, de nombreux travaux publiés proposent aujourd'hui d'utiliser directement les informations contenues dans les flux vidéo compressés pour identifier les mouvements de caméras, les objets mobiles présents, voire lever des alarmes en cas de détection d'événements anormaux.
Les méthodes consistant à décompresser pour analyser ont comme inconvénient majeur des temps de calculs importants, avec des complexités proportionnelles au nombre de pixels par image. Par ailleurs, à défaut d'utiliser les données fournies par la compression vidéo, ces méthodes peuvent être négativement impactées par celles-ci, notamment par les artefacts qui peuvent faire varier faiblement les valeurs de chaque pixel aux frontières de blocs, augmentant l'écart type des modèles de fond, et donc diminuant les performances de soustraction du fond.
De son côté, l'analyse dans le domaine compressé laisse encore une assez grande part de liberté quant aux choix des données utilisées et des algorithmes mis en œuvre. Certains n'emploient que les coefficients du domaine fréquentiel pour détecter des objets par contour ou texture, ou par modélisation du fond. Le principal inconvénient de ces méthodes est leur renouvellement qui n'a lieu que sur les images intra, aboutissant à un taux de rafraîchissement de la segmentation autour de la seconde (selon les tailles de GoP choisies).
Les algorithmes fondés sur les vecteurs d'estimation de mouvement profitent du travail qui a été fait par le codeur vidéo pour l'appariement des blocs. Mais ces vecteurs sont bruités, et un filtrage conséquent et surtout adapté est nécessaire pour proposer une segmentation correcte. Par ailleurs, les images intra ne possèdent pour leur part aucun vecteur, et représente donc des points de discontinuité pour la segmentation et le suivi.
Les approches hybrides gomment la plupart des inconvénients liés à l'analyse dans le domaine compressé, assurant une segmentation rapide sur les images intra et prédites. Elles perdent toutefois généralement l'avantage principal du temps de calcul réduit.

1.4 Discussion

Vidéosurveillance et compression sont étroitement liées. Les contraintes matérielles pour la transmission et le stockage obligent aujourd'hui les systèmes déployés à intégrer les derniers standards. MPEG-4 Part 2 est l'un des plus représentés sur les architectures en place (principalement *via* son usage pour les régies de trains et métros, que ce soit en France, en Italie ou ailleurs dans le monde). Motion JPEG et MPEG-2 restent fortement présents dans les structures plus anciennes. Dernièrement H.264 fait une percée, ayant atteint un niveau de maturité suffisant au niveau industriel pour faire partie des offres commerciales des différentes entreprises impliquées dans le secteur de la sécurité. Si de nombreux investissements sont pris en charge pour multiplier le nombre de caméras, les infrastructures plus discrètes aux yeux du grand public, telles que les serveurs de calcul, restent souvent en retrait, et les capacités de traitement ne peuvent suivre le volume de séquences généré.
Dans cette optique, l'analyse dans le domaine compressé, appliqué à la vidéosurveillance, semble idéale pour compenser une partie de ce décalage et fournir des outils de segmentation d'objets mobiles et de levées d'alarmes automatiques pour un moindre coût en temps de calcul et en mémoire. Toutefois, les différents algorithmes proposés par l'état de l'art ne permettent pas d'envisager une méthode unique répondant aux besoins des opérationnels. Ainsi, les principaux verrous persistants que nous nous sommes attachés à lever lors des présents travaux comprennent :

- La mise au point d'**une méthode unifiée**, qui soit capable de prendre en compte différents standards de compression. En effet, aucune publication ne traite de la diversité des flux vidéos. Non seulement une seule norme de compression est envisagée, mais concrètement il s'agit même d'un unique codeur, avec un unique jeu de paramètres. Hors pour un passage à l'échelle et vers une étape d'industrialisation de l'analyse dans le domaine compressé, il est indispensable de pouvoir prendre en compte la pluralité des flux recueillis. Cela passe donc par une étude des spécificités de chaque flux, puis par la détermination d'outils de traitements communs.
- L'assurance de **la cohérence temporelle de la segmentation**, en étant capable d'identifier les objets mobiles aussi bien sur les images Intra que sur les images prédites, et ainsi de fournir une extraction des zones d'intérêt pour chaque image du flux.
- **L'optimisation de l'ensemble de la chaîne de traitement**, depuis le parseur vidéo jusqu'aux outils de classification. Le principal point négatif de l'analyse dans le domaine compressé est la précision des contours, déterminée par la taille des blocs. *A priori* il n'y a pas forcément d'intérêt à considérer cette approche, sauf qu'en exploitant le travail fourni par le codeur vidéo, elle peut normalement fortement diminuer les temps de calculs. Cette caractéristique demande de considérer chaque maillon d'analyse et de tous les optimiser pour proposer l'analyse de plusieurs flux pleine résolution (720x576 pour la plupart des réseaux actuels), à 25 images par seconde, sur une même plateforme informatique.
- **La validation des outils à grande échelle**, dans un premier temps par la détermination des méthodes adaptées à la singularité des données compressées, puis par le test de l'ensemble des algorithmes sur des corpus de vidéosurveillance variés et représentatifs de nombreux cas d'usage.

Le chapitre suivant décrit ainsi la construction de notre approche, depuis l'étude statistique menée sur les vecteurs d'estimation de mouvement, jusqu'au développement d'une chaîne de segmentation optimisée et capable de prendre en compte les flux de la famille de standard MPEG.

Chapitre 2

Méthode générale d'analyse de flux vidéo compressés pour la vidéosurveillance

Résumé du chapitre

Dans ce chapitre, nous présentons dans un premier temps la singularité des données qui sont disponibles dans un flux vidéo compressé selon les standards MPEG-2, MPEG-4 Part 2 et Part 10/H.264, à partir desquelles nous avons construit des outils de détection d'objets mobiles.
Nous abordons ensuite la chaîne de traitement développée lors des présents travaux, qui comportent cinq modules communs aux différents standards. Différentes métriques ont été définies pour évaluer les résultats de ces unités d'analyse et ainsi valider notre approche. Fort du contexte industriel de cette thèse en convention CIFRE, une attention particulière a été apportée aux temps de traitement et donc à l'implantation du code, pour obtenir le meilleur compromis possible entre rapidité et précision de la segmentation.

Sommaire du chapitre

2.1 Introduction

Des différentes méthodes existantes, exposées dans le chapitre précédent, se dégagent plusieurs approches de l'analyse dans le domaine compressé. Selon qu'elles s'appuient sur un type ou l'autre de données, les algorithmes mis en place proposent des cheminements variés pour arriver à leurs résultats respectifs. D'une manière générale, un type de flux est proposé, avec son jeu d'options de compression, puis une méthode d'analyse est suggérée. Nous exposerons ici dans un premier temps les flux et options que nous avons souhaité prendre en compte, avec ce qui les rassemble et ce qui les différencie. Notre contribution est ici la mise en place d'une chaîne de traitement unifiée, qui a été rendue possible par l'étude des données brutes contenues dans ces flux, nous permettant de déterminer une réponse adaptée aux caractéristiques liées au domaine compressé. Une importante étape d'optimisation a ensuite rendu possible l'accélération de ces traitements de manière significative. Enfin, cette chaîne a été validée par la mise en place de tests et métriques permettant de valider les choix effectués.

2.2 Les données du domaine compressé

De par le contexte industriel de la thèse en convention CIFRE, une certaine exhaustivité quant aux flux traités a orienté une partie des travaux. Dans le cadre de la vidéosurveillance, les applications déployées ou en cours de déploiement à court ou moyen terme nous ont dirigé vers trois standards de compression vidéo.
Dans un premier temps, et pour se familiariser avec les particularités liées à la manipulation de données compressées, l'étude de faisabilité permettant une première validation des résultats attendus s'est effectuée sur des vidéo MPEG-2. La prise en main du logiciel de référence [MPEG-2 RS] est relativement directe et permet rapidement de traiter des flux MPEG-1 et 2 (le décodeur gérant les deux normes).
Dans un second temps, pour prendre en compte les déploiements de solutions de surveillance représentatifs, et préparer l'avenir, les travaux ont par la suite été portés sur MPEG-4 Part 2 et Part 10/H.264. Cette généralisation a également été motivée par les compétences conjointes du département ARTEMIS de TELECOM SudParis et du laboratoire MMP de THALES, travaillant tous deux sur ces deux standards. Pour MPEG-4 Part 2, la généralisation des outils d'analyse dans le domaine compressé a été rendue possible par le développement d'un parseur de flux dédié, permettant d'obtenir directement les informations nécessaires sans décompresser toute la vidéo. Pour H.264, la brique d'analyse a été intégrée dans le décodeur MMP déjà existant, AVC et SVC étant à ce moment les centres d'intérêt de l'équipe en ce qui concerne la compression vidéo.
Une attention particulière a donc été apportée sur la généricité des outils et leur adaptation aux différentes options possibles. Certaines d'entre elles se révèlent communes aux différents standards, alors que des spécificités requièrent une préparation particulière des données. Le Tableau 8 présente ces points, qui sont ensuite détaillés dans ce chapitre.
Pour la suite, nous appelons flux ou vidéo compressé(e) le flux réseau ou fichier contenant une séquence vidéo compressée. Le terme de « données compressées » renvoie aux les informations d'une vidéo compressée extraite par le parseur, et qui nous intéresse pour les traitements, à savoir les vecteurs d'estimation de mouvement, et les coefficients du domaine transformé (DCT ou transformée entière).

Tableau 8 – Les standards et options adressés

Standard	MPEG-2	MPEG-4 Part 2	MPEG-4 Part 10 / H.264 AVC
GoP de taille variable	✓	✓	✓
Entrelacement	✓	✓	× (non pris en compte par le parseur MMP)
Vecteur de mouvement	Maximum 2 par blocs, sur des images I ou P précédent et suivant	Maximum 2 par blocs, sur des images I ou P précédent et suivant	Références multiples dans le flux, prédiction intra
Blocs transformés	DCT 8x8	DCT 8x8	Transformée entière, 4x4 à 16x16
Codage des données lié au standard	Codage entropique	Codage entropique	Codage entropique ou arithmétique (CABAC, CAVLC)

2.2.1 Les concepts communs aux différents flux à traiter

2.2.1.1 Les structures de GoP

Depuis MPEG-2, les tailles de GoP sont susceptibles d'être variables. Cela permet par exemple lors d'une scène à très forte stationnarité de prolonger le groupe d'images et ainsi de retarder la prochaine image codée en intra, puisque c'est ce type de trame qui prend le plus de place dans le flux.
Les applications d'analyse de vidéo détaillées dans les articles de l'état de l'art font mention systématiquement de GoP de taille fixe. Ainsi il devient possible de prévoir le type de trame suivant et de gérer le stockage mémoire des informations simplement.
Pour permettre la continuité de l'analyse, il faut à la fois gérer l'image courante et sa ou ses référence(s), mais également les résultats d'analyse passés. Comme exposé dans le chapitre précédent, l'ordre des images diffère entre le flux vidéo et l'affichage. Mais avec des tailles de GoP variables, il n'est plus possible de prévoir la fréquence des trames Intra, ce qui demande de gérer une structure souple pour le suivi de l'historique d'analyse, comme l'illustrent les Figure 1 et Figure 20.

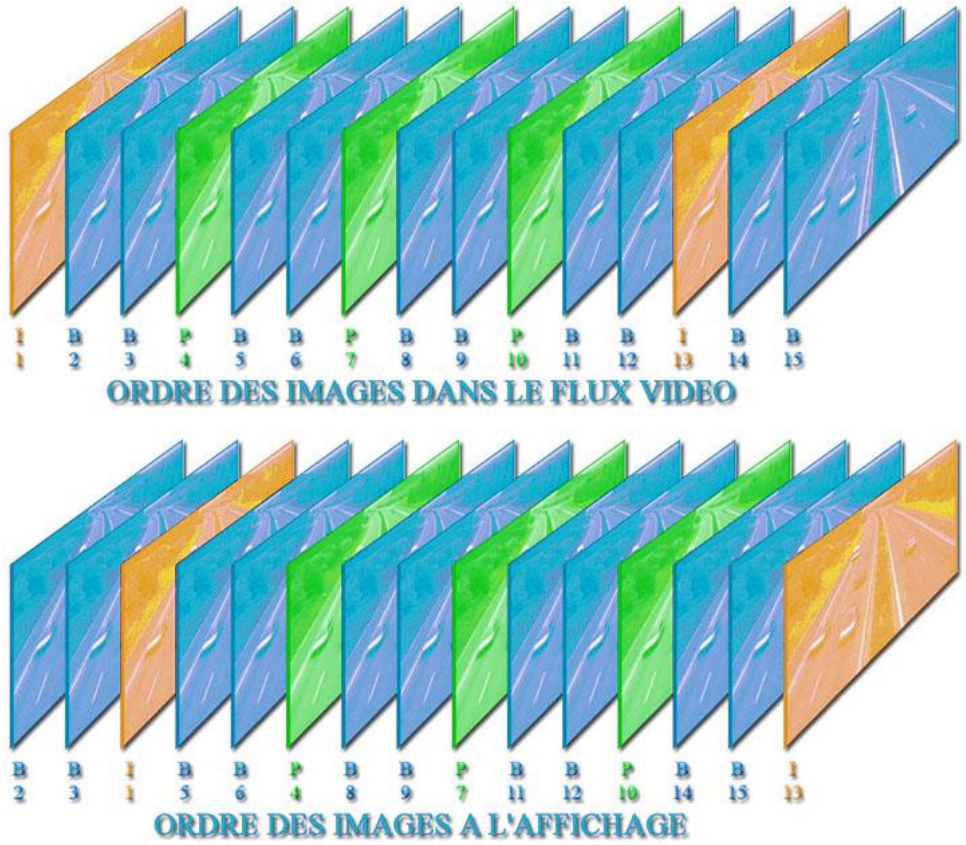


Figure 19 - Réordonnancement des GoP de taille fixe

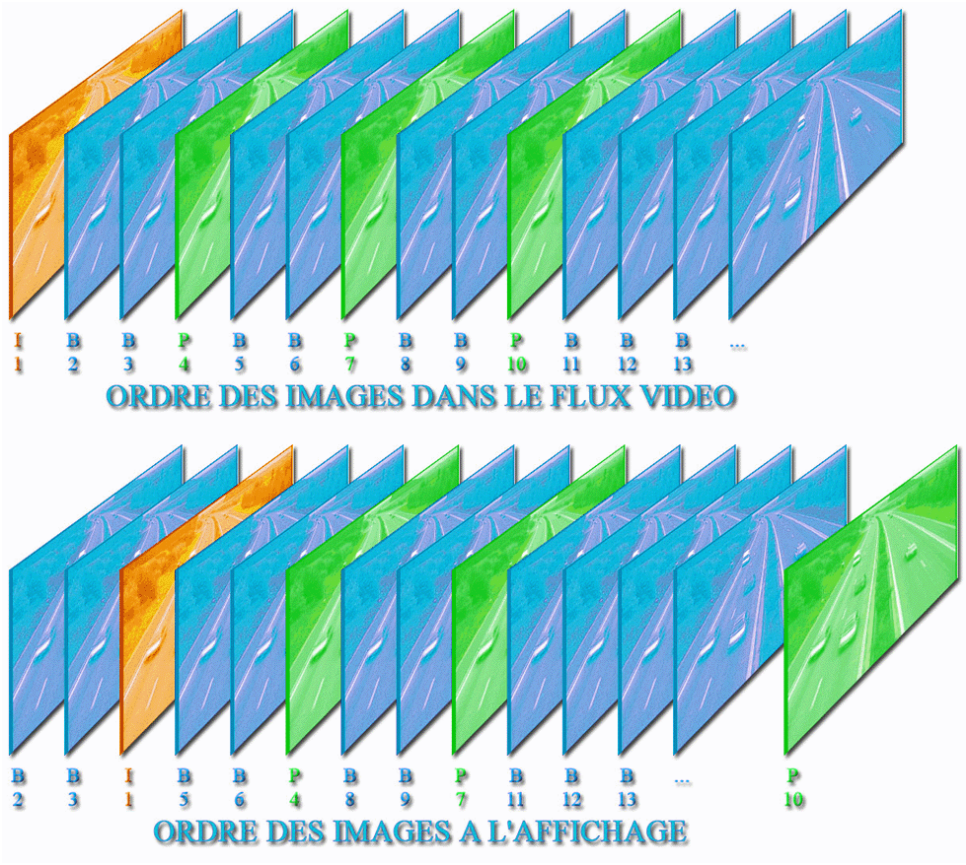


Figure 20 - Réordonnancement des GoP de taille variable

Le principal impact pour l'analyse dans le domaine compressé vient de la distance entre les trames de référence utilisées pour calculer les vecteurs d'estimation de mouvement. En effet, pour pouvoir obtenir des données comparables d'une image à l'autre, il est nécessaire de normaliser ces vecteurs. L'objectif est ici de définir une même échelle pour les données. Concrètement dans ce cas, l'échelle est la distance qui sépare l'image courante de l'image de référence vers laquelle pointe le vecteur.

2.2.1.2 L'entrelacement

Les systèmes de vidéosurveillance mis en place utilisent le plus souvent des caméras qui génèrent des flux entrelacés. C'est en effet le cas de presque tous les capteurs analogiques, et même de certaines solutions entièrement numériques. Si le codage reste à la discrétion du système et de la compression adoptée, une solution devant prendre en compte des mouvements rapides s'appuie historiquement sur l'entrelacement, puisque ce dernier propose artificiellement une fréquence d'images doublée. Même si les nouvelles générations de capteurs directement en progressifs, dont les coûts diminuent fortement, proposent des cadences d'acquisition supérieure, la plus grande partie du parc déployé reste analogique et entrelacé.
L'utilisation d'une image entrelacée en progressif génère des défauts caractéristiques, comme la présence d'objets 'fantômes' (Figure 21), phénomène connu sous le nom de '*ghosting*'.

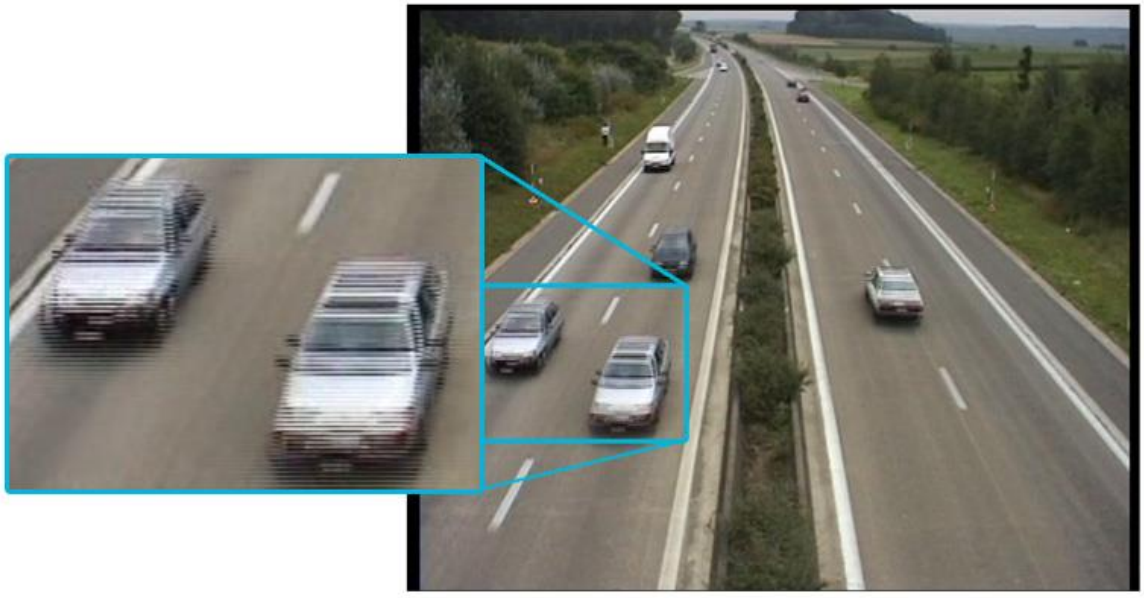


Figure 21 - Phénomène de *ghosting*
(l'entrelacement superpose deux images du même objet, dédoublant ses contours et affectant la précision sur sa position).

D'une façon générale, pour un système vidéo sur lequel l'application ici développée pourrait être utilisée, au moment de l'analyse il n'y a aucune maîtrise quant au format de l'acquisition et quant à la présence ou non d'un entrelacement.
Comme évoqué dans le chapitre précédent, l'entrelacement n'est pas pris en compte par l'état de l'art. On retrouve très régulièrement en fin de publications une simple évocation de ce format, qui selon les auteurs ne constitue qu'une petite variante et nécessite peu de travail

d'adaptation. La réalité implique des différences non négligeables et ne pas gérer l'entrelacement soulève des problèmes d'interopérabilité et la non-généricité des algorithmes.
Dans un premier temps, pour compléter les tailles variables, la structure des GoP est encore différente dans ce cas. Chaque image comporte deux champ, '*top field*' et '*bottom field*', qui peuvent à la discrétion du codeur être l'un ou l'autre en premier (même si les valeurs par défaut – que l'on retrouve dans la quasi-intégralité des codages – met le '*top field*' en premier).
Avec MPEG-1, 2 et 4 Part 2, pour les trames I, le second champ est calculé à partir du premier, et est par conséquent codé P. Pour les trames P, le premier champ est codé à partir du premier champ de la I ou P précédente, et le second à partir du premier une fois de plus. Enfin pour les B, les deux champs sont calculés à partir des premiers champs des I et/ou P les encadrant (Figure 22).
Paradoxalement, avec H.264 le problème est plus simple, puisque comme dans le cas des références multiples, l'identification de l'image à utiliser en référence est stockée dans le flux, et la formule à utiliser pour la normalisation est donc inchangée.

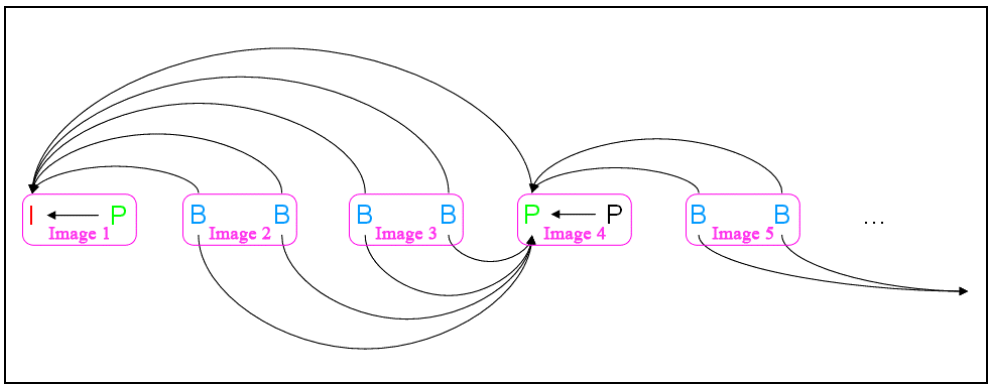


Figure 22 - Top field & bottom field : mécanisme de codage et référencement

Cette structure n'est pas sans poser problèmes puisque l'on se retrouve parfois à estimer des détails qui n'existent pas sur l'image de référence. Ce phénomène peut générer des perturbations lors de l'encodage des GoP avec des décalages d'un champ sur l'autre (ainsi que cela sera illustré Figure 35). Il faut donc adapter les méthodes de normalisation de vecteurs décrites en 2.3.2. Les distances inter-images sont doublées pour les premiers champs, et les seconds sont calculés à partir de leur(s) référence(s).

Pour conclure, ajoutons que l'entrelacement remet potentiellement en cause les métriques utilisées. Un pixel sur un champ représente dans la réalité deux pixels (en vertical). On peut même considérer l'entrelacement comme un sous-échantillonnage de moitié en vertical, accompagné d'un doublement de la fréquence. La taille des objets segmentés et les vitesses obtenues sont donc à adapter. Dans le cas d'utilisation d'histogrammes de répartition pour caractériser un objet ou permettre une sous-segmentation, il devient aussi nécessaire de remettre les distributions en cohérence.

2.2.1.3 Les vecteurs d'estimation de mouvement, données bruitées

Hormis Motion JPEG-2000, les différents standards de compression vidéo proposent globalement le même type de données : des vecteurs d'estimation de mouvement et des coefficients du domaine fréquentiel, ou coefficients transformés. Une certaine variabilité

existe, surtout entre MPEG-4 Part 10 / H.264 AVC et les normes précédentes. Pour améliorer la compression, les coefficients et vecteurs peuvent être prédits par rapport aux précédents, et les codeurs arithmétiques ou entropiques diffèrent. Toutefois, il est toujours possible d'obtenir des informations comparables, au prix d'un éventuel prétraitement recalculant ces données.

Dans la suite de cette section, la plupart des résultats et illustrations proposées correspond à l'analyse de la séquence *Speedway*, réalisée par l'Université Catholique de Louvain dans le cadre du projet ACTS MODEST, et parfois utilisée en banc test. Ici l'illustration est proposée sur un GoP représentatif de l'ensemble de la vidéo.
Le format est de 720x576 pixels en entrelacé, ce qui implique des champs de 720x288, soit 90x36 blocs (45x18 macroblocs). Les GoP comptent 12 images (24 champs).

Figure 23 – Aperçu de la séquence *Speedway*

Le premier objectif a été d'étudier l'information disponible à partir des vecteurs d'estimation de mouvement à l'intérieur d'une séquence.

Les données brutes sont envoyées au parseur qui analyse le format et les entêtes, traite le décodage entropique ou arithmétique, ainsi que la compensation de la prédiction intra pour H.264. En sortie de parseur, les composantes verticales et horizontales des vecteurs d'estimation de mouvement pour chaque macrobloc de l'image sont ainsi disponibles.
Pour permettre une validation visuelle plus parlante que la consultation de matrices et une illustration des résultats, une fonction permettant d'enregistrer les cartes de vecteurs a été développée. Les carrés gris qui quadrillent l'image correspondent aux origines de chaque macrobloc, et les 'traits noirs' sont la représentation des vecteurs d'estimation de mouvement, pointant vers l'origine du macrobloc (le point destination).

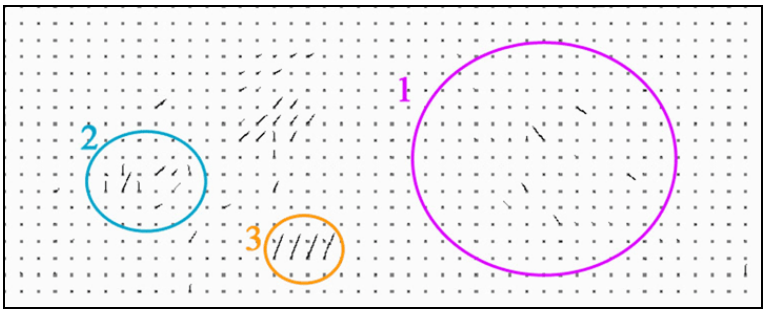


Figure 24 – Présence de bruit sur les vecteurs d'estimation de mouvement

Plusieurs remarques viennent à l'observation de ces résultats :

- Zone 1 : Peu de vecteurs sont présents, en raison du fait que les blocs codés Intra n'ont pas de vecteurs d'estimation de mouvement. Or, les séquences de vidéosurveillance présentent de nombreuses zones de stationnarités (ici seules les quatre voies de circulation évoluent sensiblement). Pour une vidéo compressée à 4Mbit/s telle que Speedway, cela autorise le codeur à utiliser beaucoup de blocs Intra pour les véhicules.
- Zone 2 : Les vecteurs sont assez bruités, par exemple sur la zone correspondant à la voiture de gauche.
- Zone 3 : Des zones unies ou texturées (arbustes) sans déplacement d'objet présentent des vecteurs d'estimation de mouvement importants.

Un traitement adéquat est donc nécessaire pour rendre ces données exploitables.

2.2.1.4 Le contenu des blocs DCT

Les études qui ont été faites sur le contenu des blocs transformés (DCT ou transformée entière) permettent d'extraire des informations pertinentes du domaine compressé, depuis une version sous résolue de la scène jusqu'à l'approximation des cartes de contours en basse résolution.

- **Les coefficients DC**

La première valeur de chaque bloc transformé correspond à la valeur moyenne de la luminance (puisque nous n'utilisons pas la couleur) sur l'ensemble du bloc. En reconstruisant une image à partir de tous ces coefficients, une image sous-échantillonnée d'un facteur 8 (ou 4 pour H.264), selon les deux axes, de l'originale est obtenue :

Figure 25 – Image reconstruite à partir des coefficients DC

C'est en utilisant cette reconstruction à toutes les images de la séquence, uniquement sur les blocs codés intra, que l'on peut s'apercevoir qu'une importante partie du mouvement ne passe pas par les vecteurs d'estimation de mouvement sur nombre de blocs. Concrètement, un codeur mesure l'impact de l'estimation de mouvement (vecteur et bloc lié à l'erreur résiduelle). Si le poids de cette solution est proche de celle consistant à codé le bloc en mode Intra (seuillage sur la différence de poids) et que la marge de débit est suffisante, le bloc sera codé en Intra, fournissant *a priori* une qualité optimale. Le nombre de blocs Intra dans les images prédites dépend donc à la fois du codeur et de son paramétrage. Pour forcer l'estimation de mouvement et valider une partie de nos outils, nous avons dans la suite été amenés à compresser une même séquence à des débits différents (Tableau 22 et Tableau 23).

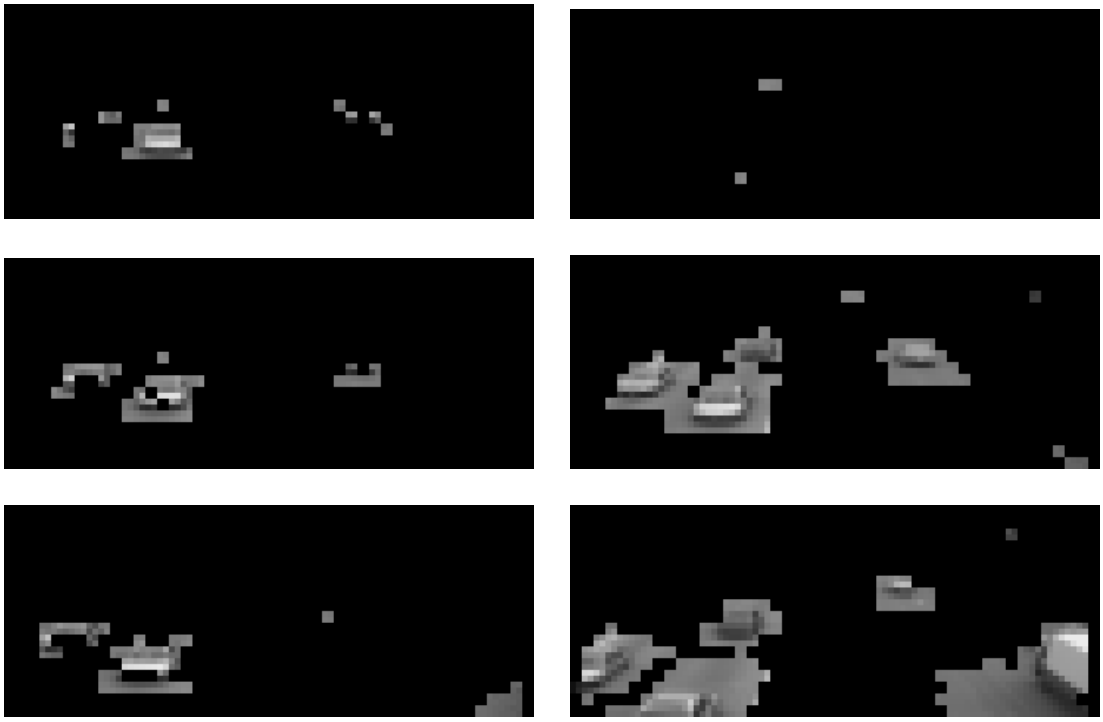


Figure 26 – Importance des blocs intra sur le mouvement

Cela souligne donc l'importance de reconstruire des vecteurs d'estimation de mouvement pour l'ensemble des blocs Intra. Cette information de luminance servira également à terme pour la segmentation par modélisation du fond dans le domaine compressé.

- **Les matrices de confiance**

Les matrices de confiance sont utilisées pour filtrer les vecteurs d'estimation de mouvement. Concrètement, le but est d'utiliser les coefficients AC_{01} et AC_{10} (Figure 27) pour déterminer une approximation des contours dans l'image.

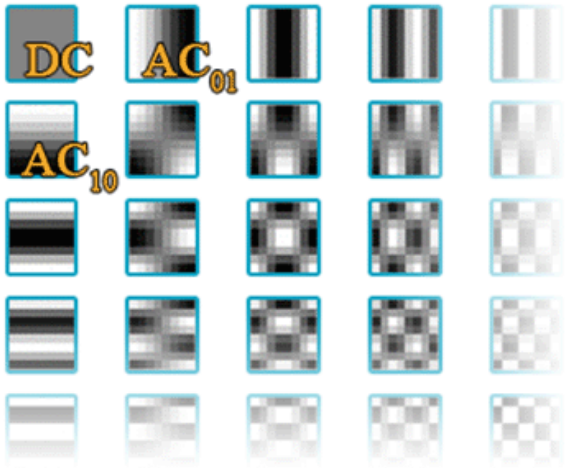


Figure 27 – Coefficients DCT considérés

Les coefficients AC_{01} et AC_{10} vont être utilisés pour approcher les valeurs du gradient en vertical et/ou horizontal et dresser une matrice de confiance quant aux valeurs des vecteurs. Le coefficient AC_{11} qui renseigne sur une direction à 45° pourrait venir compléter cette modélisation, ou même d'autres encore, mais pour une question d'accélération du traitement seules deux valeurs seront prises en compte, pour calculer :

$$Confiance = \sqrt{AC_{10}{}^2 + AC_{01}{}^2}$$

La carte des gradients est donc calculée, aboutissant à des résultats tels que ceux présentés Figure 28 et Figure 29.

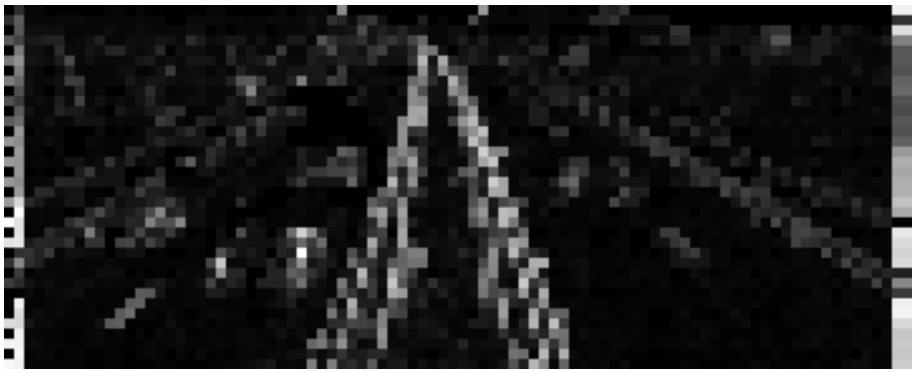


Figure 28 – Carte de confiance liée au gradient (*top field*)

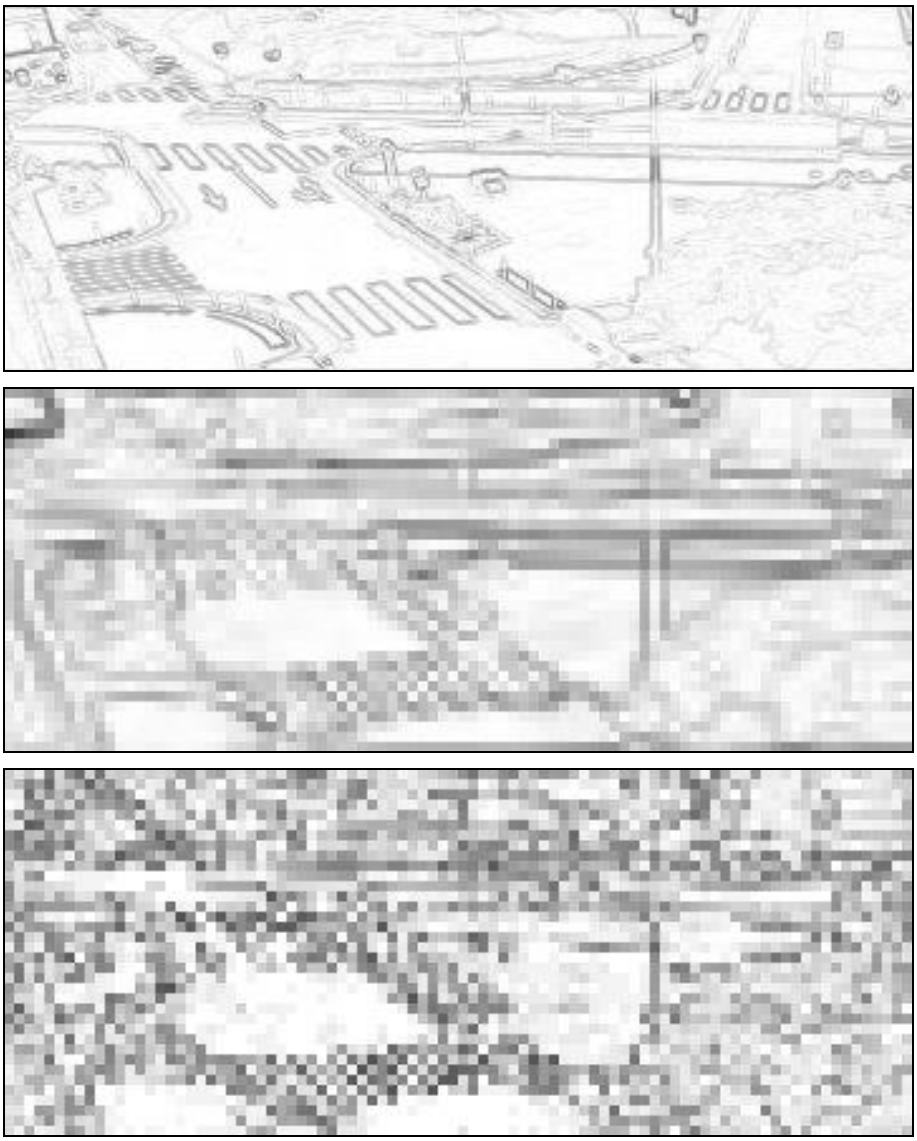


Figure 29 – Lien entre les cartes de confiance et le gradient de l'image

En haut : filtre deSobel appliqué à l'image pleine résolution, au milieu : redimensionnement par moyenne sur les blocs de l'image du haut (pour passer à l'échelle du bloc) ; en bas: carte de confiance (en négatif) obtenue à partir des coefficients DCT. Source : corpus Infom@gic.

Cette carte est obtenue de façon immédiate dans le cas d'une image I, puisqu'il suffit de calculer la norme euclidienne à partir des deux coefficients considérés. Toutefois, pour les autres images, seuls les quelques blocs Intra présents dans les images prédites permettent d'avoir une information parcellaire sur les contours présents.

2.2.2 Ce qui différencie les flux : vers une structure unifiée

Une fois les vecteurs et les coefficients transformés obtenus, peu de différences fondamentales persistent au niveau des données à traiter, du fait du jeu d'options restreint utilisé en vidéosurveillance. Les blocs n'ont pas nécessairement la même taille (généralement 4x4 pour H.264 ; et 8x8 pour MPEG-2 et MPEG-4 Part 2), mais cela revient à traiter des séquences de dimensions différentes une fois les coefficients extraits. Les vecteurs peuvent avoir de nombreuses références différentes, mais il faudra de toute façon les normaliser pour

uniformiser l'information. Leur précision diffère, du pixel au quart de pixel près, mais nous seuillerons les bits de poids faibles pour rester à une précision au pixel près (détaillé en 2.3.3). La principale différence, et qui constitue le défi majeur d'une architecture de traitement unifié, est la structure du flux compressé. En effet, la toute première étape pour analyser un flux vidéo compressé est donc d'en extraire les données nécessaires aux différents traitements. Si l'on retrouve d'une manière générale le même type d'information avec les différents standards ici considérés, la syntaxe des flux reste très différente selon que l'on parte de vidéo MPEG-2, MPEG-4 Part 2 ou Part 10.
Nous avons donc choisi d'utiliser différents parseurs selon le standard :

- En ce qui concerne MPEG-2, le parseur utilisé est celui du décodeur de référence du standard [MPEG-2 RS]. Les fonctions de décodage (DCT inverse, compensation de mouvement, etc.) ont été supprimées, puisqu'inutiles dans le cas présents et ralentissant le système.
- Pour MPEG-4 Part 2, nous avons développé spécialement un parseur.
- En ce qui concerne H.264, le décodeur développé par le laboratoire MMP a, comme pour MPEG-2, été allégé pour enlever la partie décompression. Ces trois parseurs permettent donc de prendre en compte ces différents types de flux, gérant indépendamment les caractéristiques propres à chaque standard comme la syntaxe évoquée ou le codage entropique ou arithmétique.

Une structure unifiée a ensuite été identifiée et mise en place. Pour chaque image, on crée plusieurs tableaux à la résolution du plus petit bloc, soit 8x8 pour MPEG-2 et 4 Part 2, et 4x4 pour H.264. Pour une résolution standard (SD 720x576), par exemple, on obtient ainsi des tableaux de 90x72 ou 180x144. Trois sont attribués pour les coefficients DC, AC_{01} et AC_{10} (transformée entière ou DCT), deux pour les vecteurs d'estimation de mouvement (MV_x, MV_y), les autres seront utilisés pour les résultats intermédiaires de filtrage et segmentation, et un pour la segmentation finale. Toutes les données n'étant pas disponibles pour chaque bloc (on trouve en général soit un vecteur, soit des coefficients transformés), cette structure sera majoritairement vide en sortie de parseur. Les modules de la chaîne de traitement auront pour tâche de compléter ces données.

2.3 La chaîne de traitements

La chaîne de traitements mise en place se compose de cinq sous-éléments principaux, présentés Figure 30. Contrairement aux travaux précédents, cette approche permet dans un premier temps d'exploiter les informations contenues dans le flux, que celui-ci soit codé en MPEG-2, MPEG-4 Part 2 ou H.264, puisque les données d'entrées de la chaîne sont celles issues des parseurs dédiés et stockées dans la structure unifiée. Par la suite les données sont uniformisées pour l'ensemble des blocs d'une séquence. La segmentation hybride obtenue permet d'unifier deux segmentations complémentaires pour optimiser les détections.

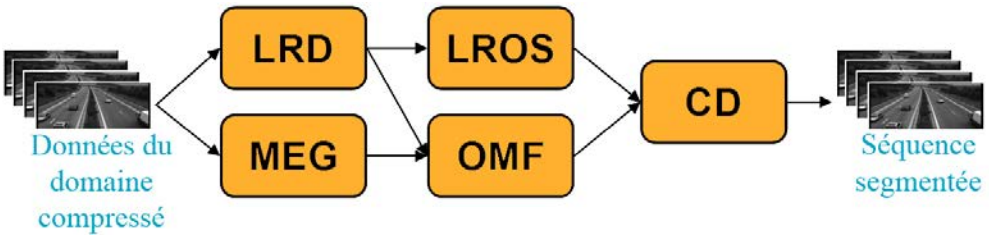


Figure 30 – Chaîne de traitement et ses modules :

LRD	*Low-Res Decoder*	Décodeur basse-résolution
MEG	*Motion Estimation Generator*	Générateur d'estimation de mouvement
LROS	*Low-Resolution Object Segmentation*	Segmentation basse-résolution
OMF	*Object Motion Filtering*	Filtrage des objets mobiles
CD	*Cooperative Decision*	Fusion multimodale

2.3.1 Le Décodeur Basse-Résolution, *LRD*

Ce module, appelé aussi LRD pour *Low-Res Decoder*, a en charge la reconstruction des images à la résolution du bloc pour les coefficients DC, AC_{01} et AC_{10}. Les informations d'entrée proviennent directement du parseur vidéo. Les coefficients sont directement recopiés pour l'ensemble des blocs Intra de la séquence. En revanche, pour les blocs prédits, une interpolation est nécessaire : pour chaque bloc, on recalcule un indice de confiance à partir des quatre (maximum) indices attribués aux blocs à partir duquel le bloc courant est reconstruit (formule pour un bloc 8x8) :

$$c_{i,j,t+1} = \frac{1}{64}\begin{bmatrix} c_{i+x/8,j+y/8,t} \times [[8-(x \bmod 8)] \times [8-(y \bmod 8)]] \\ + c_{i+x/8+1,j+y/8,t} \times [[x \bmod 8] \times [8-(y \bmod 8)]] \\ + c_{i+x/8,j+y/8+1,t} \times [[8-(x \bmod 8)] \times [y \bmod 8]] \\ + c_{i+x/8+1,j+y/8+1,t} \times [[x \bmod 8] \times [y \bmod 8]] \end{bmatrix}$$

avec (x,y) les composantes du vecteur d'estimation de mouvement.

Le nouvel indice $c_{i,t+1}$ est déterminé par une moyenne pondérée des quatre indices $c_{j,t}$ selon la surface qu'il recouvre sur chacun d'eux (Figure 31).

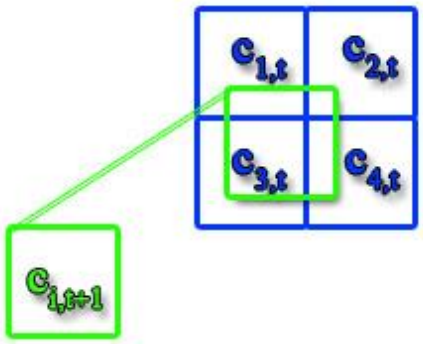


Figure 31 – Remise à jour de l'indice de confiance

(en vert, le bloc de l'image courante prédit à partir d'une zone 8x8 couvrant 4 blocs de l'image de référence).

Cette méthode est utilisée pour reconstruire les séquences basses résolutions correspondant aux coefficients DC et les deux AC.

Concrètement, les coefficients AC ne permettent pas de connaître l'emplacement d'une arête dans le cas où elle existe. Le nouveau coefficient calculé est par conséquent une approximation de la valeur réelle, que l'on ne pourrait obtenir précisément que par un calcul sur l'image décompressée.

Cette méthode de détermination de séquences basse-résolution peut donner lieu à des dérives, puisque l'on approxime à chaque nouvelle image. Toutefois selon la structure des GoP, ce problème est limité. En effet, la carte de confiance des images P est déterminée à partir de celle de la I ou P précédente. Sur une structure de GoP comme celle de *Speedway* (12 images, 3P), la dernière P est calculée après seulement trois itérations. Par conséquent, toute image B est au plus déterminée en 4 itérations depuis la I.

En revanche, sur des structures de GoP type IPPP…, l'erreur se propage sur des distances jusqu'à 25 images typiquement, ce qui introduit un effet de flou (Figure 32).

Figure 32 – Effet de flou après reconstruction par le LRD : Première image du GoP décompressé, puis image basse-résolution 1 (intra), 10 et 20 (prédite) – Source : corpus Infom@gic

Dans le cas d'un bloc codé en intra, l'indice de confiance est directement recalculé, ce qui remet à zéro toute erreur de propagation. Considérant l'importance du nombre de blocs intra observés sur les vidéos analysées, cela permet également de limiter des dérives.

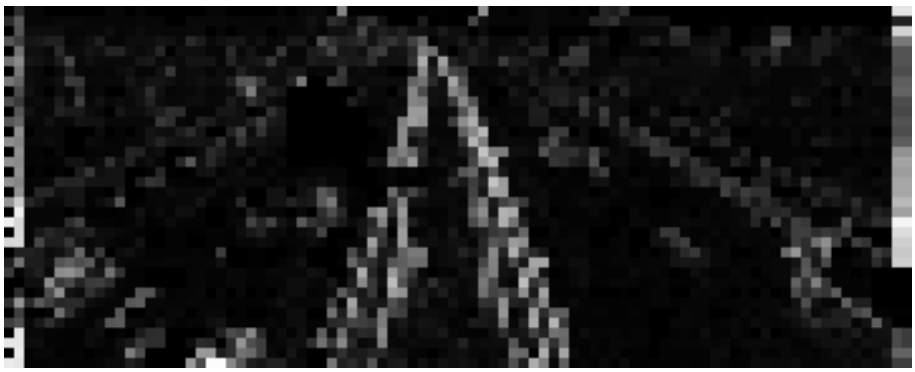


Figure 33 – image P du GoP

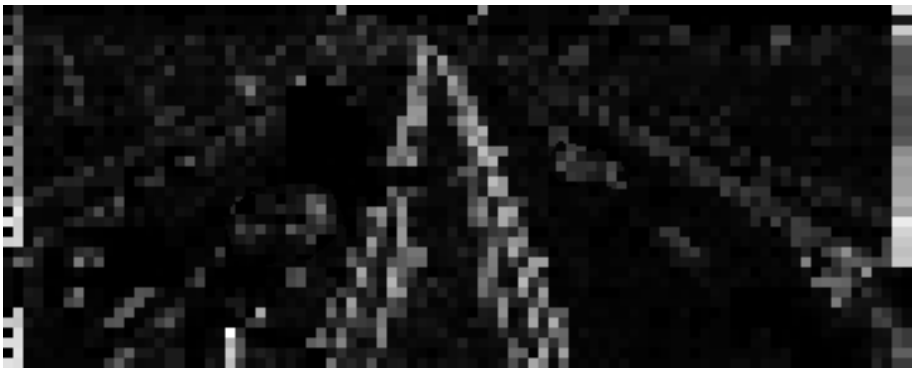


Figure 34 – Dernière image B du GoP

Comme signalé au moment de la discussion sur l'entrelacement, on s'aperçoit ici que celui-ci influe directement sur les valeurs des coefficients. Si la Figure 34 est très représentative des 'top fields' de la séquence, les 'bottom fields' ressemblent plutôt à ce qu'illustre la Figure 35.

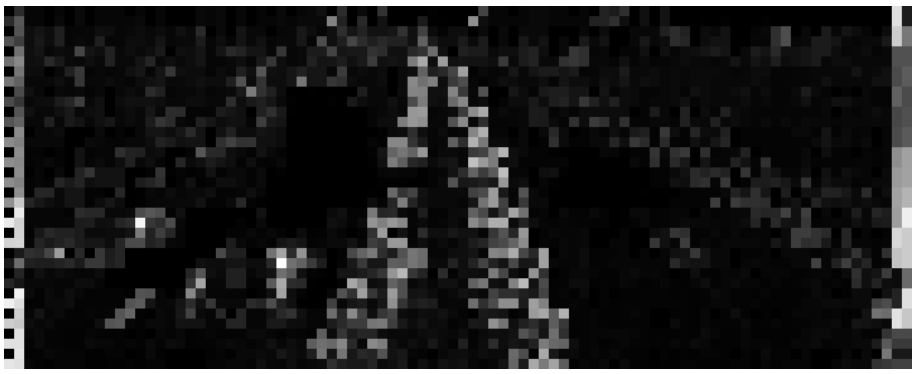


Figure 35 – Carte de confiance (bottom field)

Nous pouvons remarquer que la carte est très bruitée le long des lignes (terre-plein central ou pointillés). Dans l'absolu, un filtre médian sur l'image bruitée pourrait donc être utilisé. Nous avons préféré doubler la carte du top field, considérant qu'avec 20ms entre deux images il y a peu de différences. Cette dernière option s'avère être une approximation suffisante, et va une fois de plus dans le sens de l'optimisation du temps de calcul.
Ces cartes de confiance, une fois créées, serviront pour le seuillage des vecteurs d'estimation de mouvement.

2.3.2 Le Générateur d'Estimation de Mouvement, *MEG*

L'objectif du *MEG* pour *Motion Estimation Generator*, est complémentaire du *LRD*, et consiste à associer un vecteur à chaque bloc de la séquence. Deux étapes principales constituent le MEG : la normalisation des vecteurs et la génération de vecteurs pour les blocs codés intra.

2.3.2.1 Normalisation des vecteurs

La première étape consiste à normaliser les vecteurs selon la distance et la direction de l'image de référence. Cela nécessite de s'adapter à la structure du GoP et au mode de prédiction de l'image courante. Dans les cas d'images prédites P, il suffit de diviser le vecteur par le nombre d'images présentes depuis la dernière référence. Dans le cas de la Figure 19, ce

nombre est fixe : 3. Dans le cas de la Figure 20 en revanche, il devient nécessaire de compter cette distance pour chaque image de référence, ce qui impose de vérifier dans le flux le nombre d'image bidirectionnelles présentes avant la nouvelle référence (par exemple, le nombre de B entre les P 7 et 10 est nécessaire pour la normalisation des vecteurs de P 7). D'une manière générale, le vecteur retenu $V_{i,k}$ pour le bloc k d'une image P_i est donné par l'expression suivante :

$$V_{i,k} = -\frac{MV_{i,k}}{d_{P_{i-1},P_i}}$$

où $MV_{i,k}$ est le vecteur d'estimation de mouvement du bloc k de l'image P_i, et d_{P_{i-1},P_i} est la distance entre les deux images de références (P_{i-1} pouvant le cas échéant être une image Intra). Le signe moins est lié à l'estimation de mouvement. En effet, si un bloc « se déplace » dans une direction, la prédiction se faisant par rapport à une image de référence précédemment décodée, le vecteur est opposé au mouvement. Le but étant ici d'analyser les mouvements dans la scène, le vecteur est inversé pour être dans la direction du déplacement réel.
Pour une image B, les vecteurs peuvent pointer vers l'image de référence la précédant ou la suivant, et éventuellement vers les deux. Le vecteur normalisé correspondant est alors donné par :

$$V_{i,k} = -\alpha \frac{MV_{i,k,P-1}}{d_{P_{i-1},B_i}} + \beta \frac{MV_{i,k,P+1}}{d_{B_i,P_{i+1}}}$$

avec $(\alpha,\beta) = \left(\frac{1}{2},\frac{1}{2}\right)$ si $V_{i,k}$ est bidirectionnel,

$(\alpha,\beta) = (1,0)$ si $V_{i,k}$ pointe vers l'image de référence précédente,

$(\alpha,\beta) = (0,1)$ si $V_{i,k}$ pointe vers l'image de référence suivante.

Dans ce cas la première moitié de la somme correspond à l'image de référence précédente, et la seconde moitié à la référence suivante.
En MPEG-1, 2, et 4 Part 2 le numéro des images n'est pas présent dans le flux. Il n'y en a en effet pas besoin, seules deux images sont stockées à tout moment, I ou P, permettant le décodage de la P suivante et des B intermédiaires. Dans le cas d'un GoP variable, on ne peut se contenter d'aller chercher l'ordre d'affichage pour réorganiser avant analyse. Il faut en fait stocker les types de chaque image du GoP, l'image de référence suivante identifiée, et finalement effectuer la normalisation des vecteurs selon le type d'image.
Avec H.264, chaque bloc d'une image reconstruite pouvant avoir des références dans des images différentes du GoP, la normalisation se généralise ainsi :

$$V_{i,k} = \sum_j \alpha_j \frac{MV_{i,k,Tj}}{d_{T_j,T_i}}$$

avec $V_{i,k}$ le vecteur du bloc k de la trame T_i considérée,

$\alpha \in \left\{0,\pm 1,\pm\frac{1}{2}\right\}$ selon le nombre et la direction de la référence,

$MV_{i,k,Tj}$ le vecteur d'estimation de mouvement du bloc k de l'image T_i

en référence à la trame T_j,
d_{T_j,T_i} la distance entre la trame courante et la référence considérée.

En ce qui concerne les blocs codés par prédiction intra, le vecteur associé n'ayant aucun sens en termes de mouvement (codage sans différence temporelle, à l'intérieur d'une même image), ils sont traités comme les blocs intra.

2.3.2.2 Déterminer un vecteur pour les blocs codés intra

Là où le LRD devait traiter les blocs prédits, le *MEG*, après voir normalisé les vecteurs, calcule un vecteur pour chaque bloc intra de la séquence. Cette tâche s'appuie sur une hypothèse de mouvement constant. A partir des images précédentes dans la séquence, une translation de chaque vecteur est effectuée selon leur valeur. Si deux vecteurs pointent vers le même bloc, une moyenne est appliquée. Cette hypothèse se justifie par l'intervalle typique de 40 ms séparant 2 images (à 25 images par seconde), ne laissant apparaître que très rarement de fortes variations de vecteurs sur des mouvements réels (à part crash dans un mur, explosion, ou autre incident à forte accélération).
La combinaison *LRD/MEG* permet donc d'obtenir pour chaque bloc de la séquence les mêmes informations, un vecteur, homogène à l'ensemble des vecteurs de la séquence, et trois coefficients transformés, qui seront utilisés par les modules suivants.

2.3.3 Le Filtrage des Objets Mobiles, *OMF*

Ce module s'appuie sur les résultats du LRD et du MEG, à savoir les vecteurs normalisés et les cartes de confiance. Dans un premier temps, l'objectif est de supprimer le bruit qui n'a pas de lien avec un mouvement réel dans la scène, puis de filtrer le bruit persistant sur les vecteurs traités.

2.3.3.1 Les cartes de confiance

Les cartes de confiance permettent de valider la pertinence des vecteurs d'estimation de mouvement établis par le codeur.
Concrètement, la compensation de mouvement ne correspond pas nécessairement à la réalité telle qu'elle est filmée, c'est-à-dire au déplacement d'un objet réel dans la scène, mais à une minimisation d'erreur sur une fenêtre de taille prédéfinie à l'encodage. Le vecteur permet donc la recopie (suivi éventuellement d'une correction d'erreur) d'une zone précédemment décodée sur l'image actuelle. Mais si cette zone est totalement unie, le vecteur peut pointer indifféremment vers n'importe quel endroit dans ces limites. Ce problème est connu sous le nom de problème d'ouverture ('*aperture problem*'), présenté Figure 36.
A celui-ci s'ajoute une variante : Le problème de 'mur blanc' ('*blank wall*'), illustré par la Figure 37, se produit lorsqu'il s'agit de prédire un bloc le long d'une arête. Le même cas de figure que précédemment se présente tout le long de l'arête.

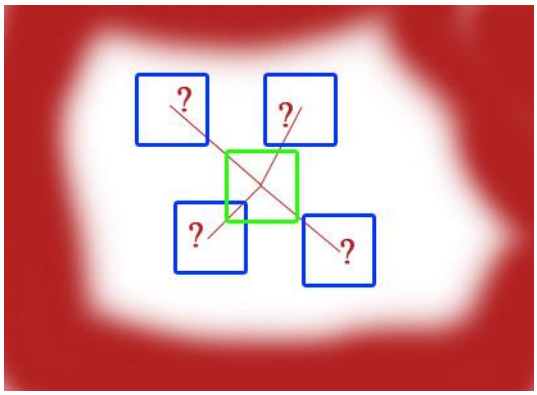


Figure 36 – Problème d'ouverture (*Aperture*)

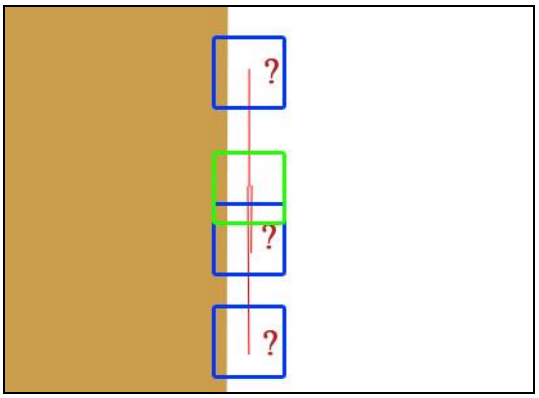


Figure 37 – Problème de mur blanc (*Blank Wall*)

Pour un bloc à prédire dans l'image courante (en vert), plusieurs candidats sont possibles (exemples en bleu) dans la ou les images de référence.

Il faut retenir qu'en cas d'*aperture*, on ne peut donner aucun crédit au vecteur d'estimation de mouvement puisque celui-ci ne correspond pas nécessairement à un mouvement. S'il s'agit de *blank wall*, seule la direction orthogonale à l'arête est pertinente.
L'Object Motion Filtering permet donc dans un premier temps de diminuer voire supprimer l'effet des *blank walls* et *aperture*. Concrètement, un seuillage est effectué pour chaque vecteur selon les valeurs de la carte de confiances calculée dans ce module. Si l'indice de confiance est inférieur à une portion (paramètre de l'algorithme) de l'indice maximum, le vecteur est mis à zéro.

2.3.3.2 Filtrer les vecteurs

Les remarques mentionnées en 2.2.1.3 demandent une étude sur les vecteurs bruts, principalement à cause du besoin de filtrage. Il est donc nécessaire de caractériser la répartition des vecteurs, en ayant au préalable déterminé un espace de représentation. Pour chaque image, un graphe 3D a été créé, sur lequel les deux premiers axes correspondent aux composantes horizontale et verticale de chaque vecteur, le troisième axe permettant de reporter le nombre de vecteurs possédant ces valeurs.
La première remarque concerne la présence d'un pic correspondant au vecteur (0,0). Ceci est attendu : bien que du bruit soit présent, sur une séquence de vidéosurveillance il est logique qu'une majorité de vecteurs, que l'on peut associer au fond (immobile a priori), ait deux composantes nulles. On obtient ainsi une représentation telle que la Figure 38. Le problème est que l'on ne peut se rendre compte de la répartition des vecteurs non nuls. L'ensemble des

vecteurs (0,0) a par conséquent été retiré au moment de la représentation graphique. Cela conduit à la Figure 39 ,correspondant à la même image du flux que la précédente.

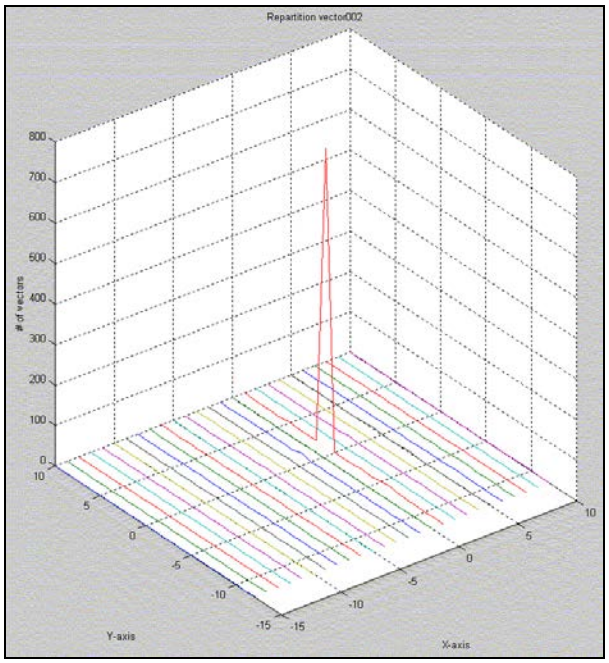


Figure 38 – Répartition totale des vecteurs

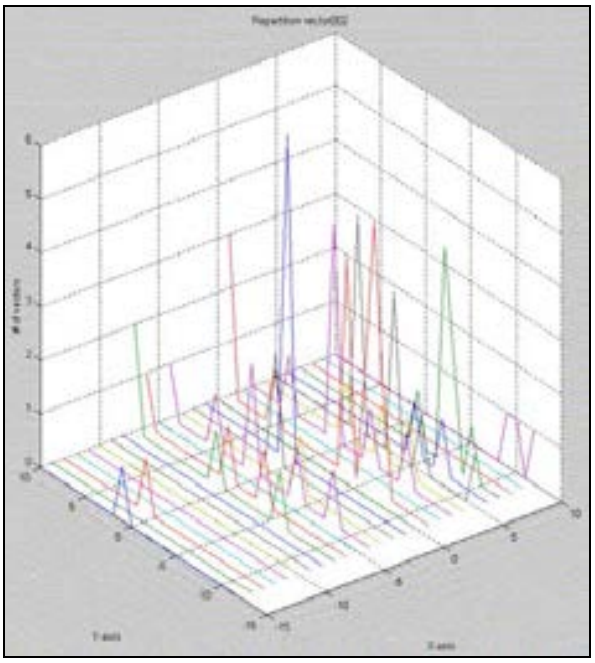
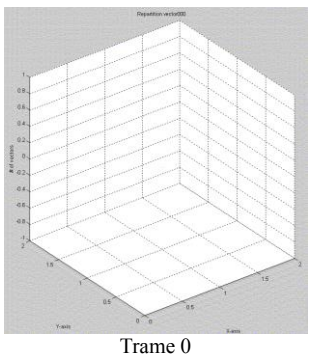
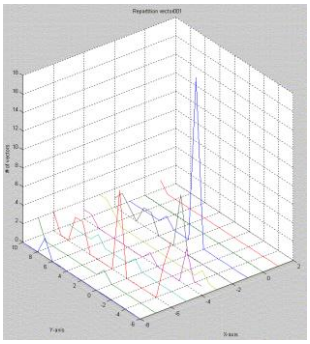


Figure 39 – Répartition après soustraction des vecteurs (0,0)

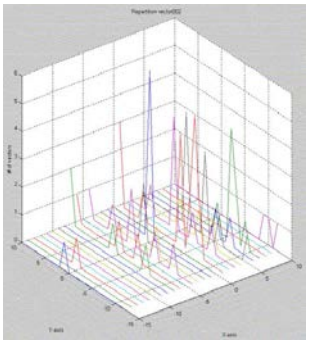
On obtient le même type de graphe sur l'ensemble des trames de la séquence test, comme on peut le constater Figure 40. La première trame correspondant à une image codée en intra, il est naturel de ne trouver aucun vecteur d'estimation de mouvement.



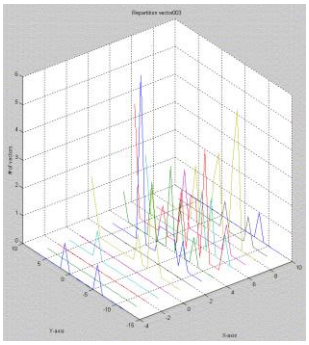
Trame 0



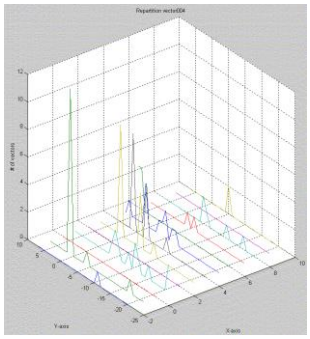
Trame 1



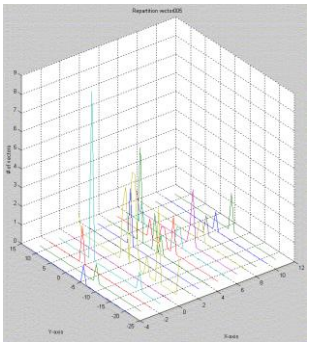
Trame 2



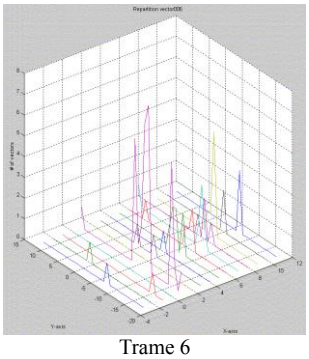
Trame 3



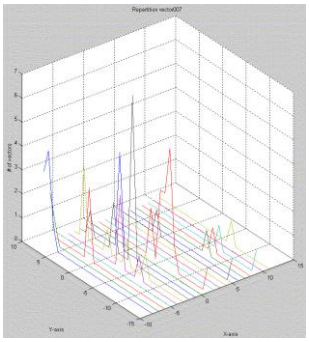
Trame 4



Trame 5



Trame 6



Trame 7

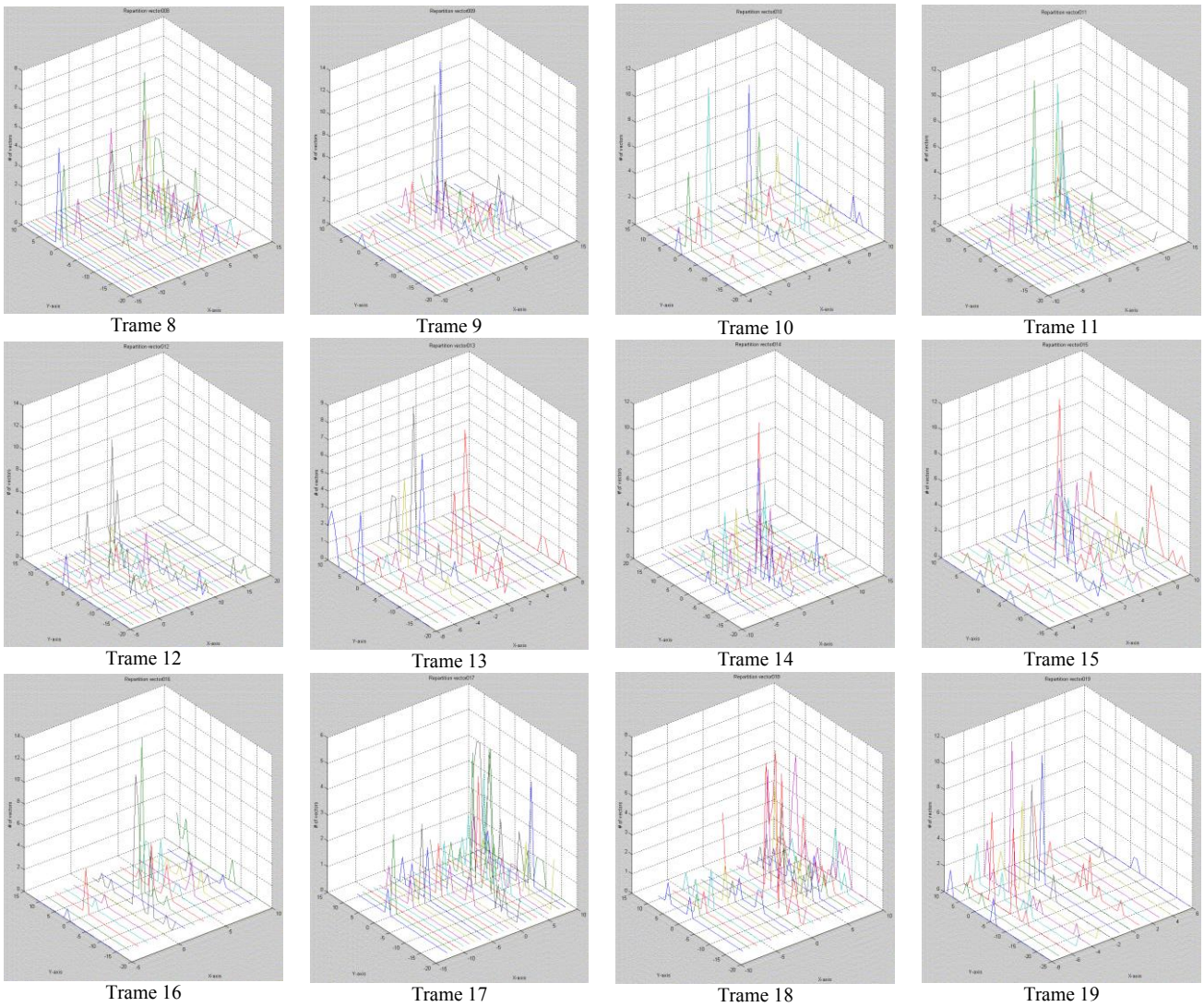


Figure 40 – Répartition sur les trames d'une séquence

Les outils statistiques permettent d'identifier le type de bruit et les traitements qu'il sera possible d'appliquer. Dans un premier temps les moyennes et écarts types ont été calculés, aboutissant à des moyennes en X autour de -2,5 pixels et en Y de -1,50 pixels, avec des écarts types allant selon les trames de 4 à 12 pixels.

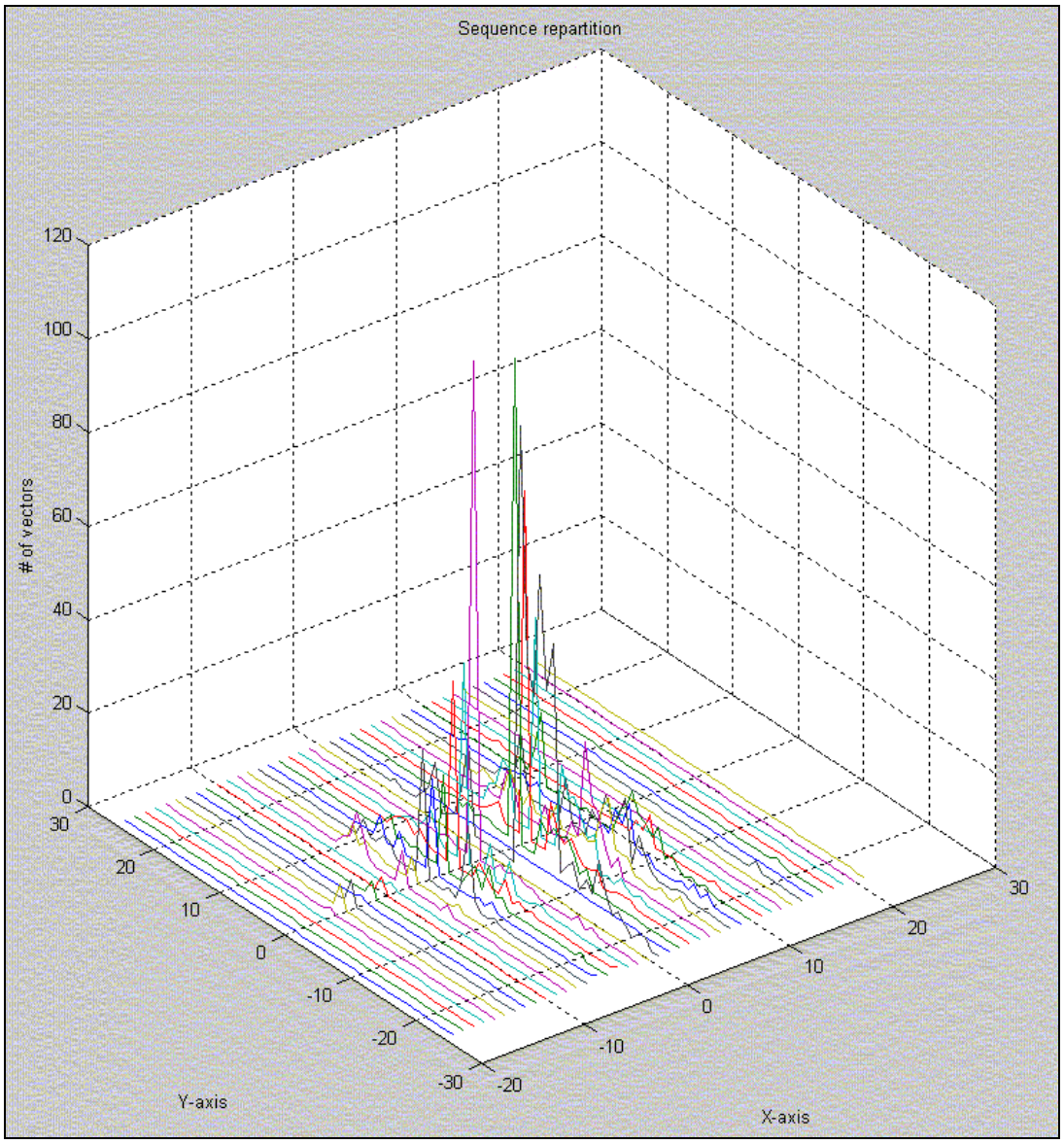


Figure 41 – Répartition sur l'ensemble d'une séquence (*Speedway*)

L'agrégation de ces données sur la séquence complète a été effectuée, comme le montre la Figure 41. On remarque sur cette figure le 'trou' laissé au centre suite au retrait des vecteurs de composantes nulles, ceux-ci s'élevant à plus de 15 000 contre moins de 120 pour le deuxième pic le plus haut. On obtient dans ce cas les valeurs présentées Tableau 9.

Tableau 9 – Modélisation statistique du bruit sur les vecteurs

Moyenne en X	-2,42	Kurtosis en X	3,62
Moyenne en Y	-1,58		
Ecart type en X	2,92	Kurtosis en Y	4,87
Ecart type en Y	6,83		

Cette modélisation a été appliquée à un corpus représentatif de séquences possibles en vidéosurveillance, qui sont présentées dans le chapitre suivant.

Tableau 10 - Corpus utilisé pour la modélisation des vecteurs

	Speedway (séquence 2)	Caretaker - Métro Turin	WCAM Séquence Outdoor 1	WCAM Séquence Outdoor 2 (de nuit, fortement bruitée)	Séquence statique (plan fixe)
Format	MPEG-2	MPEG-4	MPEG-2	MPEG-2	MPEG-4
Largeur	720	704	352	352	720
Hauteur	576	288	288	288	576
Entrelacement	✓	×	×	×	✓
Images par seconde	25	6	25	25	25
Référence de la séquence	A	B	C	D	E

Concernant le contenu, la séquence dans le métro de Turin représente des piétons dans une station (borne d'achat et tourniquets), les séquences de WCAM en extérieur, des scénarios mêlant voitures et piétons de jour ou de nuit. La séquence statique permet quant à elle une validation sur un plan fixe (l'évolution d'une image à l'autre venant principalement des bruits de capteur).
Les mêmes statistiques ont été extraites, comme présentées Tableau 11.

Tableau 11 - Statistiques des vecteurs sur le corpus

Séquence	A	B	C	D	E
Moyenne en X	-1,17	-17,9	-8,42	-21,4	-4,23
Ecart type en X	5,56	42,05	20,8	36,0	9,28
Kurtosis en X	14,79	4,39	3,92	5,02	6,47
Moyenne en Y	-1,15	-3,86	-6,87	-8,42	-0,71
Ecart type en Y	8,27	16,7	12,35	18,3	3,29
Kurtosis en Y	16,92	7,28	7,85	8,31	3,25

Les valeurs généralement considérées pour assimiler une courbe à une gaussienne correspondent à une valeur de Kurtosis comprise entre 2 et 4. Les valeurs obtenues sortent majoritairement de ce cadre. Une valeur au-delà de 4 indique que l'on pourra traiter une partie du bruit en lui attribuant un caractère impulsionnel.
L'*OMF* applique donc ensuite un filtre médian au niveau spatial (la taille de fenêtre étant un paramètre de l'algorithme), puis un second filtre dédié au niveau temporel. Concrètement, ce dernier étudie l'historique du bloc sur les deux dernières images, et s'il a un mouvement régulier (pas de discontinuité ni de valeurs aberrantes), valide le vecteur. Dans le cas contraire, il est une fois de plus mis à zéro.

2.3.4 La Segmentation Basse-Résolution, *LROS*

Le module *Low-Res Object Segmentation* utilise les séquences reconstruites à partir des coefficients DC pour établir un modèle du fond à partir de moyennes et écarts types. Ces derniers sont générés à partir des valeurs des coefficients DC de chaque bloc et les valeurs instantanées sortant de ce modèle sont attribuées au premier plan. Une période d'apprentissage est donc nécessaire à l'initialisation de l'algorithme pour que les premiers

modèles soient disponibles. Un filtre médian permet de lisser la carte de segmentation obtenue.

2.3.5 La Décision Coopérative, *CD*

L'OMF et le *LROS* aboutissent chacun à une segmentation de l'image, *a priori* comprenant les zones où se situent les objets mobiles. Mais chaque outil à ses défauts. L'OMF, qui s'appuie principalement sur les vecteurs d'estimation de mouvement, procure de meilleurs résultats pour les images P et B, alors que le LROS, utilisant les coefficients DCT des blocs intra et ceux reconstruits par le LRD, se révèle meilleur sur des images intra.
L'*OMF* détecte très mal voire pas du tout les petits objets (de la taille d'un à deux blocs) tels que les piétons loin de la caméra. Cela est dû principalement à la compensation du mouvement qui ne procure pas de suivi cohérent pour des petites zones, et donc au filtrage de sortie qui, s'il permet de s'affranchir du bruit restant, élimine aussi les blocs segmentés isolés. De son côté, le *LROS* est précis pour les images codées Intra, mais perd en précision à mesure que l'on s'éloigne de ces images, et donc que la reconstruction basse-résolution devient floue.
Au final, une étude subjective, comparant visuellement les segmentations issues de l'OMF et du LROS, a été réalisée. Les résultats attendus, différenciant LROS et OMF sur les images I, P et B, ont permis d'établir une stratégie pour la fusion par le module de décision coopérative : des poids sont attribués à chaque segmentation. Sur les images intra, celui du LROS est beaucoup plus élevé que celui de l'OMF (90%-10% en pratique), puis la tendance s'inverse, d'autant plus que la distance à l'image intra est grande (jusqu'à 10%-90% pour les dernières images du GoP). Ce paramétrage permet globalement d'obtenir des segmentations assez constantes dans le temps, comme le montre l'exploitation des résultats dans le chapitre suivant.

2.3.6 La chaîne de traitement complète

La Figure 42 propose la chaîne de traitement complète éclatée, détaillant les fonctions des cinq modules précédemment détaillés, et qui est donc à mettre en regard de la Figure 30. A ceux-ci s'ajoute une dernière étape, correspondant à l'extraction de descripteurs.
Pour les différentes applications qui seront détaillées dans le chapitre 4, différents descripteurs pourront être utilisés. En sortie de segmentation, les données sont ici obtenues image par image. On extrait ainsi :

- Les coordonnées de chaque blobs,
- Les dimensions des blobs (longueur, largeur, et nombre de blocs couverts),
- La position du centroïde,
- Le ou les vecteurs d'estimation de mouvement associé.

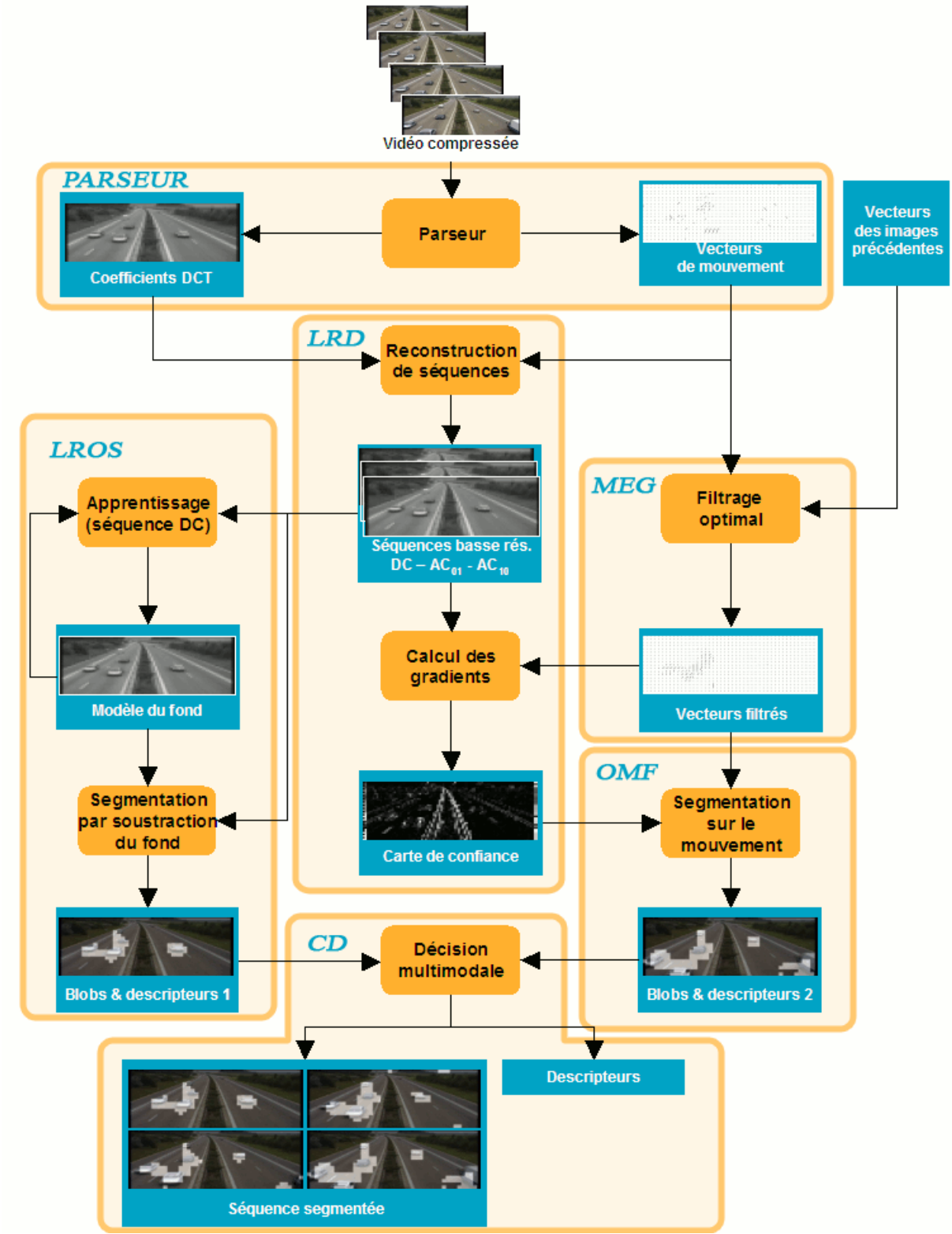


Figure 42 – Segmentation des objets mobiles sur une image

2.4 Optimisation des temps de traitements

L'analyse de vidéo dans le domaine compressé possède l'intérêt principal d'utiliser en partie le travail effectué par le codeur vidéo pour simplifier les tâches de traitement et ainsi accélérer le temps de calcul nécessaire pour obtenir la segmentation des objets mobiles dans une

séquence. Les travaux de l'état de l'art font ainsi régulièrement mention d'analyse à 3 ou 5 fois le temps réel, alors que les outils du domaine décompressé peinent à atteindre une fois le temps réel. Dans le cadre du projet Caretaker par exemple (décrit en 3.2.1), les partenaires s'appuyant sur une segmentation au niveau pixellique se maintenaient au temps réel, mais sur des images 720x288 (prise en compte de la trame paire seule sur des séquences entrelacées), à 6 images par seconde.

2.4.1 Simplification des données

Selon les standards, les coefficients DC doivent être codés sur 11 bits (de par la construction de la transformée en cosinus discrète). Il s'agit du nombre minimum de bits nécessaires pour obtenir la précision suffisante lors de la transformée inverse. Pour des raisons d'uniformisation des traitements, entre coefficients DC et AC, et pour optimiser la gestion mémoire, les trois bits de poids faible des coefficients DC sont ignorés, permettant de les stocker sur un octet.
Cette approximation, outre les avantages précédemment présentés, permet également de s'affranchir d'une partie du bruit qu'il peut exister sur les coefficients DC et perturbe la modélisation du fond dans le domaine compressé. L'impact concret tient dans un modèle moins précis (8 fois moins précis sur 8 bits que sur 11), mais qui s'est révélé lisser les valeurs moyennes sur les blocs, perturbées à la précision maximale simplement par le bruit de capteur (voir l'exploitation des résultats sur la séquence).

Par ailleurs, la précision des vecteurs a évolué avec les différents standards : du pixel près avec MPEG-1 jusqu'au quart de pixel avec H.264. Si cette finesse de prédiction se révèle intéressante d'un point de vue compression, elle n'apporte pas d'amélioration quant à la segmentation dans le domaine compressé. Le module de filtrage du MEG seuillant entre autre les vecteurs de moins de un ou deux pixels (paramètre empirique de la chaîne de traitement, à 2 pour MPEG-2 et MPEG-4 Part 2, à 1 pour H.264 qui présente des blocs plus petits), conserver cette précision est inutile. La troncature au pixel près permet ici également de coder chaque vecteur sur 2x8 bits (de -128 à +127 pixels, dans chaque direction), sans impact sur la suite des algorithmes.

2.4.2 Choix algorithmiques

D'une manière générale, il existe généralement de nombreuses implantations différentes d'algorithmes proches permettant d'aboutir à des résultats comparables. Lorsque ce cas de figure s'est présenté, le choix s'est systématiquement porté sur l'outil présentant la complexité la plus faible pour accélérer les temps de traitements.
Parmi ces nombreux partis pris, l'hypothèse de mouvement constant a déjà été évoquée. Au sein du MEG, il n'y a *a priori* pas de raison de privilégier la continuité du mouvement à un modèle affine, ou même d'un ordre supérieur. Il manque généralement un jeu de vecteurs par GoP, correspondant à l'image intra. Même dans le cas d'une accélération extrême (exemple de l'explosion ou de l'accident), le risque est d'attribuer un mouvement faux sur une image, qui sera ensuite corrigé par l'image prédite suivante. L'impact négatif est donc limité. En revanche, le gain est significatif, puisqu'au lieu de modéliser 2 ou 3 paramètres pour un vecteur, qui demanderait dans l'absolu un historique plus long que deux images, le MEG se contente de suivre le mouvement des blocs sur ces deux images et d'appliquer la translation correspondante pour attribuer une valeur moyenne aux blocs intra.

La modélisation du fond dans le domaine compressé s'appuie elle aussi sur l'un des algorithmes les plus simples possibles pour la soustraction du fond : des tests ont été réalisés en utilisant des modèles à base d'histogrammes ou de mélanges de gaussiennes (*GMM*). Les résultats étaient au final peu supérieurs au modèle valeur moyenne / écart type retenu (moins de 5% sur la détection des objets, mais 10 à 15% sur la précision des contours). En effet, en n'employant qu'un coefficient par bloc, le moyennage limite les effets de scintillement par exemple qui mettent en avant des stratégies de modélisation plus complexes. En revanche, dans des cas particuliers, comme la présence d'affiches déroulantes (métro de Rome) ou d'escalators, les motifs récurrents sont segmentés inutilement.
Globalement, cette simplification du modèle permet de n'occuper que deux octets en mémoire par bloc (moyenne / écart type), contre 256 dans le cas des histogrammes par exemple (en codant toutes valeurs prises sur 8 bits).

2.4.3 Optimisation des algorithmes

Même si les outils retenus sont ceux présentant les plus faibles temps de calculs pour les résultats les moins dégradés possibles, il est toujours possible de choisir une implantation plutôt qu'une autre pour un algorithme donné, permettant une fois de plus d'optimiser les temps de traitement.
Ainsi les modules de morphologie mathématique ont été étudiés pour trouver la meilleure approche informatique possible. Pour le filtrage des résultats de la segmentation, érosion et dilatation sont utilisées. Les implantations en deux passes (mesure de distance au noir ou au blanc, puis seuillage) ont été retenues. Pour l'étiquetage des blobs, les premiers développements utilisaient une approche récursive, plus facile à mettre en place mais gourmande en ressources mémoire et processeur (pile – *stack* – et cycles). L'outil finalement retenu est en trois passes. La première a lieu sur l'image, partant d'en haut à gauche et affectant des indices croissants masques de segmentation. Lorsqu'un élément du premier plan est à la droite ou au dessous (en 4-connexité) d'un pixel du premier plan déjà traité, la même valeur lui est attribuée. Lorsque deux blobs se rejoignent plus bas dans l'image (donc lorsqu'il s'agit au final du même blob), les valeurs à apparier sont stockées dans un tableau. La seconde passe a lieu sur le tableau, réduisant les paires pour grouper tous les blobs qui doivent l'être, et leur attribuant leur valeur définitive (de 1 au nombre total d'objets donc). La dernière passe balaie de nouveau l'image pour affecter les valeurs aux blobs. Cela permettra, outre le comptage des objets mobiles par image, de les adresser indépendamment les uns des autres pour l'extraction de descripteurs ou vignettes.

2.4.4 Optimisation du code

Dans un contexte de thèse CIFRE, tous les développements ont été effectués avec des contraintes d'industrialisation des différents outils à court ou moyen terme. Ainsi l'ensemble des logiciels produits par le laboratoire MMP de Thales est dès le départ conçu pour être déployé sur des cibles diverses (depuis d'importants serveurs multi-cœurs jusqu'à des plates-formes embarquées très basses consommation). Ce parti pris entraîne quelques contraintes, mais permet également de penser un code optimisé dans son ensemble, ce qui appuie l'objectif de temps de traitement réduit visé par l'analyse dans le domaine compressé.

Outre l'uniformisation des outils, le passage de différentes données (vecteurs, coefficients DC) sur 8 bits permet surtout une meilleure gestion de la mémoire utilisée par l'algorithme. En effet, les systèmes d'exploitation et les unités de calcul sont aujourd'hui généralement de 32 ou 64 bits pour un ordinateur, et peuvent descendre à 16 voire 8 bits pour des cibles embarquées comme des DSP ou FPGA. Pour les accès mémoire (lecture, écriture, copie, initialisation, etc.), des mots binaires au format natif du système sont synonyme de performances optimales. Pour toute l'exploitation sur ordinateur, le système d'exploitation utilisé pour ces travaux étant 32 bits, les données étaient par exemple traitées 4 par 4 lors de copies de mémoire. Si 11 bits avaient été conservés, ils auraient préférablement été stockés sur deux octets (perdant ainsi 5 bits) pour conserver l'alignement mémoire au niveau octet, mais doublant le volume total de données.

Concernant les perspectives de portage, l'ensemble des formats est redéfini via un « s » ou « u » précisant si la valeur est signée ou non (*signed* ou *unsigned*), suivi du nombre de bits la codant. Ainsi les vecteurs sont des s8, et les coefficients AC des u8. Cette règle est utilisée pour tous les développements vidéo à MMP et permet par exemple de lever certains doutes sur les tailles de données entre système d'exploitation et unités de calculs (un CHAR sur 8 bits sur cible x86 ou sur 16 bits pour un C5416 de Texas Instrument, un LONG sur 32 ou 64 bits selon les versions de windows en x86, et sur 40 bits sur plateformes C6 de Texas Instrument par exemple). Par ailleurs, l'intégralité du code est en C ANSI, pour accélérer les procédures de portages sur cibles si nécessaire. Nous reviendrons sur ces perspectives dans le chapitre suivant.

De nombreux éléments simples ont également contraint les développements pour obtenir des temps de calcul aussi faibles que possible. Nous citerons ainsi à titre d'exemple l'inversion des boucles permettant de parcourir une image. Spontanément le code ressemblait initialement à :

```
for (x=0 ; x<width ; x++)
{
	for (y=0; y<height; y++)
	{
		traitement(paramètres);
	}
}
```

A l'exécution, le traitement sera ici effectué colonne par colonne, donc changeant de ligne à chaque nouvelle valeur. Or, d'un point de vue de stockage mémoire, une image est stockée ligne par ligne, donc un saut est nécessaire à chaque nouveau traitement. Si l'on inverse les boucles on a :

```
for (y=0 ; y<height ; y++)
{
	for (x=0; x<width; x++)
	{
		traitement(paramètres);
	}
}
```

Ainsi, le traitement sera appliqué à toute une ligne avant de passer à la suivante. La gestion mémoire est ainsi optimale puisque le déplacement d'un élément à l'autre est d'une taille (un octet généralement dans le cas présent), au lieu de L octets (L étant la largeur de l'image). De plus il n'y a pas de retour en arrière dans la mémoire : l'image est traitée entièrement en parcourant d'une traite la zone mémoire valeur par valeur.

Sur des boucles contenant des nombreux calculs répétés de nombreuses fois, il est aussi pertinent de pré-calculer les indices qui seront utilisés dans les boucles internes. Ainsi, les images étant concrètement des tableaux à une dimension dans l'implantation, adresser le pixel de coordonnée (x,y) correspond à cibler la valeur (x+y*Largeur_Image) du tableau. Le pré-calcul de l'indice se traduit alors comme suit :

```
pre_indice=0;
for (y=0 ; y<height ; y++)
{
        for (x=0; x<width; x++)
        {
                traitement(x+pre_indice);
        }
        pre_indice += width ;
}
```

L'avantage est que le calcul sur le pré-indice, correspondant concrètement au choix de la ligne, n'est fait qu'une fois par ligne et non pour chaque pixel. De plus, cette méthode n'utilise que des additions, généralement traitées en un cycle processeur, quand les multiplications peuvent en requérir jusqu'à 4 selon la cible.

Dans la même optique, les décalages de registre sont utilisés autant que possible. Les blocs ayant eux-mêmes comme taille des puissances de 2, de nombreuses valeurs sont multipliées ou divisées par 2, 4, etc. jusque 64 pour l'interpolation selon la zone couverte dans le LRD. Or en binaire, multiplier ou diviser par 2 puissance n, c'est effectuer un décalage à gauche ou à droite de n bits. Dans le cadre d'une division, on passe d'une dizaine de cycles à un seul cycle processeur.

Au final, ce sont toutes ces règles plus ou moins contraignantes qui permettent d'optimiser la chaîne dans son ensemble et ainsi obtenir des temps de calcul réduits.

2.4.5 Temps de traitements obtenus

L'évaluation des temps de traitement a été effectuée en analysant divers corpus (codés en MPEG-2, MPEG-4 Part 2 et H.264) dans le domaine compressé, sur des séquences de même résolution (720x576), à 25 images par seconde. Le code a été instrumenté pour déterminer le temps effectif nécessaire aux traitements, puis le nombre total d'images d'un corpus a été divisé par le temps total requis pour les traitements sur ce corpus, ce qui permet de s'affranchir des dérives liées aux temps variables entre les I, P et B.
Dans les prochains tableaux, la première ligne liste les différents modules de la chaîne. La seconde ligne précise le nombre d'images traitées par seconde, en activant les modules de traitement les uns après les autres (d'où les performances décroissantes). La troisième ligne donne ainsi le temps total nécessaire pour traiter une image à mesure que l'on active les

modules. Enfin la dernière ligne donne le temps nécessaire à un module pour traiter une image.

Tableau 12 – Temps de calcul sur des séquences MPEG-2 (Intel Core 2 6700 @ 2.66GHz, calculs sur un cœur, 2 Go de RAM)

Module	Parseur	+ LRD	+ MEG	+ OMF	+ LROS	+ CD
Nombre d'images traitées par seconde (fps)	726.2	710.2	447.6	424.1	398.2	363.4
Temps total par image (ms)	1.377	1.408	2.234	2.358	2.511	2.752
Temps du module par image (ms)	1.377	0.031	0.826	0.124	0.143	0.241

Tableau 13 – Temps de calcul sur des séquences MPEG-4 Part 2 (Intel Core 2 6700 @ 2.66GHz, calculs sur un cœur, 2 Go de RAM)

Module	Parseur	+ LRD	+ MEG	+ OMF	+ LROS	+ CD
Nombre d'images traitées par seconde (fps)	278.7	276.5	223.17	217.53	211.15	200.8
Temps total par image (ms)	3.588	3.617	4.481	4.597	4.740	4.980
Temps du module par image (ms)	3.588	0.029	0.864	0.1162	0.143	0.240

Tableau 14 – Temps de calcul sur des séquences H.264 (Intel Core 2 6700 @ 2.66GHz, calculs sur un cœur, 2 Go de RAM)

Module	Parseur	+ LRD	+ MEG	+ OMF	+ LROS	+ CD
Nombre d'images traitées par seconde (fps)	121.8	120.1	93.8	89.6	84.9	78.6
Temps total par image (ms)	8.211	8.323	10.664	11.166	11.779	12.722
Temps du module par image (ms)	8.211	0.112	2.341	0.502	0.613	0.943

La première remarque porte sur le parseur : la complexité syntaxique croissante depuis MPEG-2 jusqu'à H.264 se manifeste par des temps nécessaire pour accéder aux données du domaine compressé en forte augmentation : depuis 1.377 ms pour MPEG-2 jusque 8.211 ms pour H.264.
En revanche, une fois la structure unifiée remplie par le parseur, les temps de traitement des différents modules sont comparables. La dernière ligne est quasi identique entre MPEG-2 et MPEG-4 Part 2, et est environ le quadruple pour H.264. Ceci vient de la taille des blocs, 4x4 au lieu de 8x8. Il y a donc 4 fois plus de blocs à traiter, et les temps sont quatre fois plus long.

2.5 Validation des choix algorithmiques et discussion

Comme exposé dans le chapitre consacré à l'état de l'art, de nombreuses méthodes qui s'appuient soit sur les vecteurs soit sur les coefficients DCT existent, mais les solutions hybrides s'avèrent généralement tirer le meilleur parti des différents jeux de données présents

dans le domaine compressé. Nous avons donc choisi d'aborder le problème en nous appuyant sur ces deux segmentations qu'il faut fusionner. Les cinq fonctions que nous venons de détailler permettent effectivement d'aboutir à ce résultat, mais elles ont de plus été validées indépendamment les unes des autres pour justifier l'implantation globale. Cela a été rendu possible par la mise au point de métriques, qui autorise au final de suivre l'évolution de performance d'un outil algorithmique le long de séquences.

2.5.1 Evaluation du décodeur basse-résolution, LRD

Dans l'absolu, un LRD parfait reconstruirait une séquence qui correspondrait à un sous-échantillonnage (d'un facteur correspondant à la taille des blocs) de la séquence dans le domaine décompressé. La première étape consiste donc effectivement à s'appuyer sur la vidéo décompressée, puis à moyenner pour chaque bloc l'ensemble des pixels pour obtenir le résultat optimum du LRD (LRD_{opt}).
Afin de mesurer les performances du LRD, une métrique a été mise au point, sur le modèle de la *MSE*, ou *Mean Square Error* pour Erreur Quadratique Moyenne. La déviation du LRD est ainsi donnée par :

$$LRD_{dev} = \frac{1}{width \times height} \sum_{i,j} \left[LRD_{opt}(i,j) - LRD(i,j) \right]^2$$

où i et j sont les index en largeur et hauteur à l'intérieur d'une image.

Cette métrique permet d'identifier au sein d'une séquence les images pour lesquelles des erreurs de reconstruction manifestes apparaissent, ainsi que celles pour lesquelles le décodeur se comporte bien. Cette caractéristique est visible Figure 43, qui présente les variations de LRD sur une séquence d'une centaine d'images.

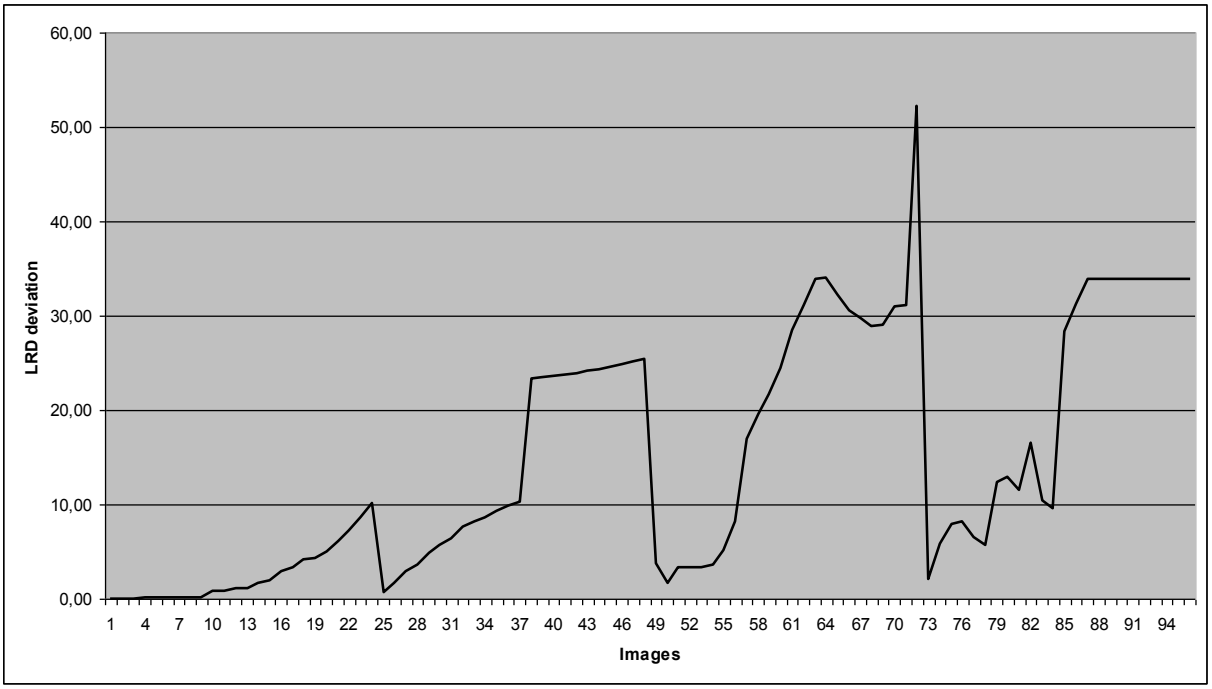


Figure 43 – Métrique d'évaluation du LRD

De par sa définition, la déviation du LRD est théoriquement nulle lorsqu'une image codée intra est comparée à l'image correspondante, décompressée puis sous-échantillonnée. La déviation augmente le long d'un GoP. Lorsqu'il y a peu de mouvement et/ou de variation, comme c'est le cas ici pour les 25 premières images, les valeurs de déviation du LRD restent faibles. A chaque nouveau GOP (ici de 24 images), la déviation du LRD retombe, comme on le note pour les images 24, 48, 72.
En revanche, lorsqu'un objet apparaît soudainement dans la scène (image 38 par exemple), la déviation du LRD augmente brutalement. Cela vient principalement du fait que le LRD ne prend pas en compte l'erreur résiduelle qui permet de compenser fortement ce type de modification dans le contenu.
A contrario, lorsque des blocs intra sont présents dans des images prédites, ils compensent la dérive qui s'opère sur un bloc, et permettent de faire redescendre la déviation. Ce phénomène est observable aux images 62 à 68.

2.5.2 Evaluation du générateur d'estimation de mouvement, MEG

Mesurer la performance du MEG nécessite de pouvoir comparer des vecteurs déterminés par un codeur vidéo avec ceux générés par le module. Il faut donc des blocs intra avec des vecteurs, ce qui en principe n'existe pas en MPEG-2, MPEG-4 ou H.264 (sauf vecteurs permettant une correction d'erreur, ou vecteurs de *concilement*).
Une méthode pour contourner cet obstacle est la génération de séquences artificielles. Un carré en translation dans une vidéo, avec des paramètres de déplacement connus, est une solution. Toutefois, le choix du vecteur de mouvement ne se faisant pas selon les mouvements réels, mais selon un critère de minimisation d'erreur entre pixel, il n'est pas assuré que le vecteur calculé, s'il est différent de celui mis en paramètre de la génération de séquence de synthèse, soit pour autant différent de celui qu'aurait calculé un codeur vidéo.
Une autre approche a donc été mise au point, permettant d'accéder aux vecteurs d'estimation de mouvement réellement calculés par le codeur vidéo, le tout pour des blocs intra. Concrètement, une même séquence a été codée deux fois ; avec des tailles de GoP variant du simple au double, comme illustré Figure 44. Une première séquence sera ainsi codée par exemple avec des GoP de 6 images (IPPPPP ou IBBPBB), alors qu'une seconde en comportera 12. Le module MEG est alors appliqué aux images prédites de la seconde séquence correspondant aux images intra de la première.

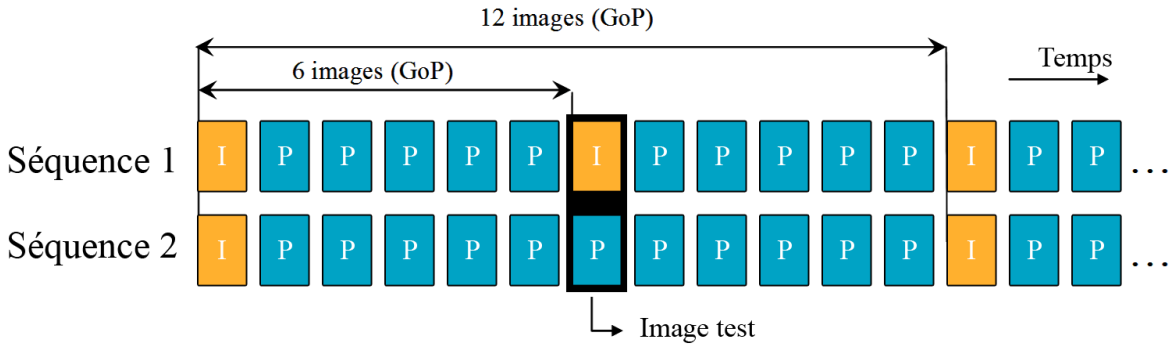


Figure 44 – Sélection des images tests pour le MEG

Ainsi, chaque image test fournit les blocs codés intra depuis la séquence 1, et les blocs correspondant prédits depuis la séquence 2. On extrait les résultats du MEG sur ces images

(soit 1 sur 12, ce qui nécessite des séquences longues pour l'évaluation). On considère ici le résultat optimum MEG_{opt} comme étant les vecteurs extraits de la séquence 2, et l'on définit la déviation du MEG par :

$$MEG_{dev} = \frac{1}{width \times height} \sum_{i,j} \sqrt{\left(\Delta MEG_{X,i,j}\right)^2 + \left(\Delta MEG_{Y,i,j}\right)^2}$$

$$\text{Avec } \Delta MEG_{X,i,j} = MEG_{opt_{n,X}}(i,j) - MEG_{n,X}(i,j)$$

$$\text{et } \Delta MEG_{Y,i,j} = MEG_{opt_{n,Y}}(i,j) - MEG_{n,Y}(i,j)$$

$\Delta MEG_{X,i,j}$ et $\Delta MEG_{Y,i,j}$ correspondent à la différence entre l'une ou l'autre des composantes du vecteur fourni par le codeur et celui reconstruit par le MEG. Tout comme pour le facteur de déviation du LRD, il est possible de visualiser l'évolution de MEG_{dev}, permettant d'extraire des images saillantes, comme illustré Figure 45.

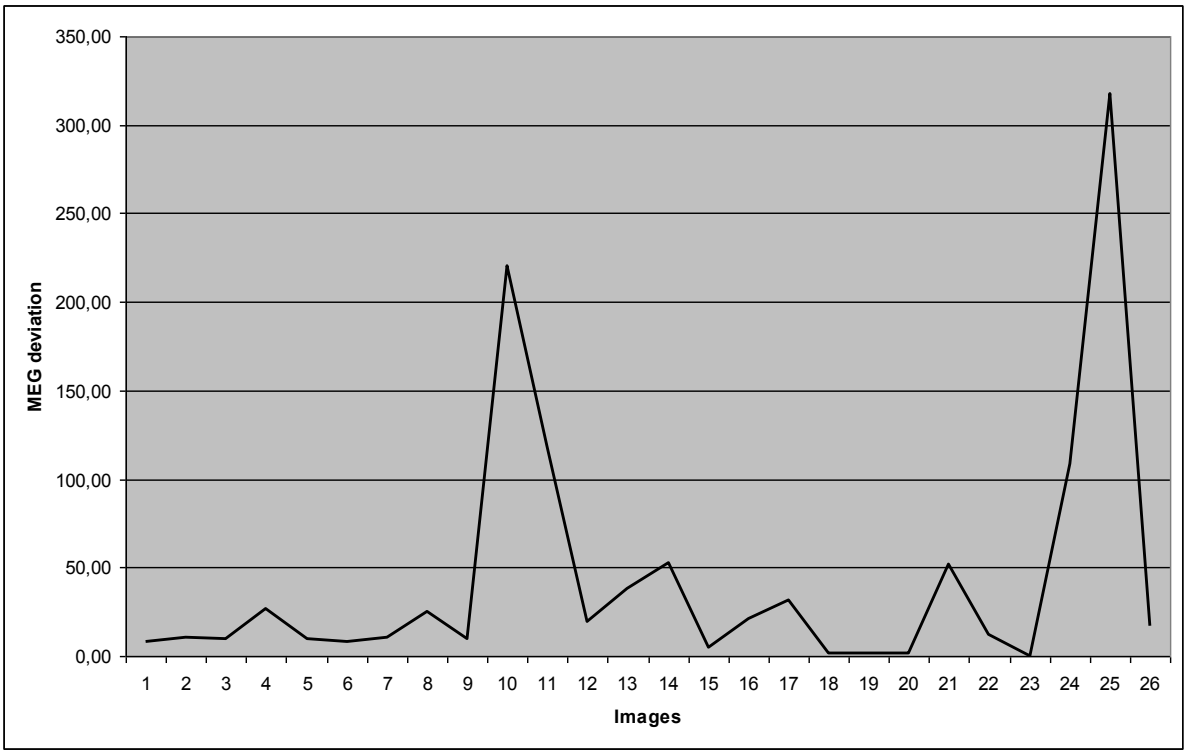


Figure 45 – Déviation du MEG

Si, dans le cas du LRD, l'évolution de la déviation selon les GoP et la présence de mouvement étaient prévisibles, celle du MEG l'est beaucoup moins. De par la construction des séquences de validation, une grande variabilité sur les courbes s'explique par le caractère discontinu des images prélevées sur les séquences (1 image sur 12 typiquement).
D'une part, des pics importants sont identifiés (images 10 et 25 Figure 45), et permettent de comprendre pourquoi le MEG n'a pas réagi aussi bien que sur les autres images intra. Dans ces cas, les deux images précédant directement l'intra contenaient un véhicule entrant dans le champ de vision, et qui ont été codés en intra lors de la compression. Ce comportement, attendu de la part du codeur vidéo, a eu pour effet de supprimer les vecteurs liés au mouvement sur ces images, et donc de fausser l'interpolation réalisée par le MEG. Sur la

séquence avec l'image prédite en revanche, après deux images codant le nouvel objet en intra, l'estimation de mouvement a commencé à attribuer des vecteurs aux blocs considérés.
D'autre part, les images 18 à 20 sont des images codées intra au sein de plans fixes dans la séquence. Aucun objet mobile n'est présent, ce qui explique la reconstruction presque parfaite sur ces blocs.

2.5.3 Discussion sur les modules de segmentation, OMF, LROS, CD

Initialement, une métrique a également été développée pour l'évaluation du module de segmentation. Il s'agit d'une comparaison entre une vérité terrain (*ground truth*) et les résultats obtenus par les modules OMF, LROS et CD. Le principe consiste à sur-échantillonner les segmentations dans le domaine compressé, pour obtenir des masques à la même résolution que les images décompressées. La métrique s'appuie alors sur la différence entre les deux masques de segmentation :

$$OMF_{dev} = \frac{1}{width \times height} \sum_{i,j} \left| FRS_{Full-res}(i,j) - FRS_{Low-res,OMF}(i,j) \right|$$

$$LROS_{dev} = \frac{1}{width \times height} \sum_{i,j} \left| FRS_{Full-res}(i,j) - FRS_{Low-res,LROS}(i,j) \right|$$

où $FRS_{Full\text{-}res}$ est la vérité terrain de la segmentation sur l'image pleine résolution (*Full Resolution Segmentation*),
et $FRS_{Low\text{-}res,OMF/LROS}$ sont les segmentations pleine-résolution obtenues à partir des segmentations basse-résolution de l'OMF et du LROS par sur-échantillonnage.

Deux problèmes ont été soulevés par l'utilisation de ces métriques. Le premier est le temps nécessaire à la création de vérités terrain. Des outils de segmentation rapide, travaillant au niveau pixellique, ont servi de point de départ, puis le détourage a été repris à la main. Certains contours restent subjectifs (au niveau des ombres, d'éventuels reflets, etc.), et il faut d'une manière générale près d'une demi-heure de travail par image pour obtenir un résultat convenable. Une seule séquence a donc été détourée (provenant du corpus Infom@gic présenté dans le chapitre suivant).
Le second problème soulevé tient au fait que ces métriques se sont avérées peu concluantes. Il est en effet difficile de comparer des segmentations obtenues à des résolutions différentes (surtout avec un facteur 4 voire 8). Un objet correctement segmenté avec son voisinage, du fait de la « résolution bloc » de l'approche, peut générer de fortes valeurs de déviation OMF ou LROS, alors qu'il reste tout à fait pertinent pour l'exploitation des résultats. Au niveau pixellique, les résultats de segmentation sur cette séquence test approchaient les 30% d'erreur sur les modules LROS et OMF, et 25% une fois la fusion multimodale ajoutée, soit un pixel sur quatre mal classé.

Une seconde approche a ensuite visé à compter le nombre d'objets segmentés image par image. Ici, l'approche dans le domaine compressé montre des limites, puisque les modèles simples aboutissent par exemple à des détections dans la végétation, ou à deux blobs sur une personne (torse et jambes séparés), voire à des non-détections. Des résultats de segmentation sur les images consécutives de différentes séquences sont présentés dans le chapitre suivant et illustrent bien ces discontinuités (p119).
Au final, une approche d'un plus haut niveau sémantique a donc été choisie, et sera abordée dans le chapitre suivant.

2.6 Conclusion

En étudiant les données brutes présentes dans les flux compressés en MPEG-1, 2, 4 Part 2 et 10 / H.264, un schéma de segmentation générique a été mis au point. Dans le cadre de la vidéosurveillance, les jeux de paramètres des différents codeurs vidéo sont assez similaires. Ainsi, une fois abstraction faite de la syntaxe propre à chaque standard, tâche dévolue aux différents parseurs dédiés, la mise en place d'une structure unifiée permet d'utiliser une chaîne de traitement commune à tous ces standards. Cette dernière permet de s'affranchir du type d'image et de bloc en s'appuyant sur l'historique de la séquence pour compléter les informations disponibles par des interpolations lorsque cela est nécessaire. Ainsi, pour chaque bloc, un vecteur d'estimation de mouvement, un coefficient DC, AC_{01} et AC_{10} sont disponibles, permettant d'effectuer deux segmentations fondées d'une part sur le mouvement à partir des vecteurs ayant subi un filtrage dédié, d'autre part sur une modélisation du fond basse résolution. La première se révèle efficace sur les images prédites, les zones mobiles importantes et les mouvements rapides, alors que la seconde permet davantage de détecter des objets sur le long terme de par sa meilleure gestion des images Intra. La combinaison des deux permet donc d'obtenir une segmentation image par image des objets mobiles. Le choix du domaine compressé, qui soulève parfois de nombreuses fausses alarmes et non détections, est donc compensé partiellement par cet algorithme dual. Toutefois, cette segmentation image par image reste parfois imprécise, et il faut jouer sur les paramètres pour privilégier un faible taux de fausses alarmes ou de non-détection. Ce compromis, commun à tous les systèmes de segmentation, peut être partiellement optimisé par la prise en compte d'outils de lissage temporel.
Hormis la problématique de passage à l'échelle et du traitement de volume conséquent de flux de vidéosurveillance, le prochain chapitre aborde concrètement une problématique liée à l'investigation et à la recherche d'objets dans les vidéos. Ces considérations passent entre autre par la mise en place d'outils de suivi d'objets, qui répondent au problème de lissage de la segmentation dans le temps.

Chapitre 3

Expérimentation et validation grandeur réelle

Résumé du chapitre

Ce chapitre présente les résultats obtenus lors des expérimentations qui ont été menées pour valider la chaîne de segmentation détaillée dans le chapitre précédent. Pour le passage à l'échelle, un corpus de test présentant une importante variabilité autour des séquences de vidéosurveillance a été construit. Cette étape a été possible grâce à différents projets collaboratifs ou internes à Thales pour lesquels nous avons apporté différentes contributions dans le cadre des présents travaux. Ces projets et les corpus liés sont ainsi décrits dans un premier temps, puis les résultats sont discutés, couvrant aussi bien les verrous levés que ceux restant encore à déverrouiller. Un premier exemple applicatif est alors détaillé : une chaîne de traitement permet de répondre à des besoins d'identification dans le cadre d'investigations.

Sommaire du chapitre

3.1 Introduction

Analyser des flux de vidéosurveillance, afin d'en extraire des informations pertinentes pour les exploitants, nécessite de pouvoir s'adapter à un maximum d'événements susceptibles d'être captés par le réseau de caméras. Du point de vue du développement, cela implique de disposer d'un corpus de séquences représentatifs des situations réelles, et de valider les outils mis en place sur un panel de situations aussi large que possible.
Les différents projets qui ont sous-tendus les présentes recherches seront présentés dans une première sous-section. Nous décrirons ensuite les différents corpus associés, en indiquant les normes mises en jeu, la variabilité des contenus et les défis à relever. Enfin, les résultats obtenus seront détaillés et discutés en termes de forces et faiblesses des outils mis au point, ainsi qu'en termes d'atteinte des objectifs initialement fixés.

3.2 Présentation des projets

Qu'il s'agisse du département ARTEMIS de TELECOM SudParis ou du laboratoire *MultiMedia Processing* de THALES Communications, les entités d'encadrement de cette thèse sont fortement impliquées dans les travaux de recherche collaboratifs, au niveau français et européen. Ainsi les développements et avancées proposés ont pour la plupart fait l'objet de contributions pour plusieurs de ces projets, qui ont à la fois fourni des données concrètes sur lesquelles travailler, mais également précisé les contextes d'application avec les objectifs à atteindre.

3.2.1 Caretaker

3.2.1.1 Présentation

Tableau 15 – Projet CARETAKER

Nom du projet	CARETAKER
Date de début de projet	Mars 2006
Date de fin de projet	Novembre 2009
Type de projet	Européen, FP 6, IST-2004-2.4.7
Partenaires	THALES Communications (France), Multitel (Belgique), INRIA (France), Kingston University (Angleterre), IDIAP (Suisse), ATAC et GTT (Italie), Solid (Finlande), Brno University of Technology (République tchèque).

Caretaker est l'acronyme pour *Content Analysis and REtrieval Technologies to Apply Knowledge Extraction to massive Recording*.
Ce projet européen avait pour but d'étudier, de développer et d'évaluer une analyse de contenu fondée sur la connaissance multimédia, les composants d'extraction de connaissance et les sous-systèmes de gestion de métadonnées dans le contexte de prise en compte

automatique de situations, de diagnostic et d'aide à la décision. Plus précisément, CARETAKER s'est intéressé à l'extraction de connaissances structurées à partir de grandes bibliothèques multimédias enregistrées via des réseaux de caméras et microphones déployés dans des sites réels. Les flux audiovisuels produits, outre les problèmes de surveillance et de sécurité, peuvent représenter une source d'informations utiles s'ils sont stockés et analysés automatiquement. CARETAKER avait pour but de modéliser et gérer deux types de connaissance :

- d'une part, celle relative aux multiples utilisateurs (opérateurs de sécurité, preneurs de décisions) et représentée par leurs besoins, leur définition de scénarios type et leur capacité à fournir des descriptions de contextes pour les données capteurs ;
- d'autre part, la connaissance du contenu, caractérisée par une première couche d'événements primitifs pouvant être extraits des flux de données brutes, telles que les sons ambiants, l'estimation de la densité de la foule ou la trajectoire des objets, et une seconde couche d'événements sémantiques plus élevés, définis sur une analyse de séquences plus longues et à partir de relations plus complexes entre les événements primitifs et de plus haut niveau.

Les deux types de connaissance sont modélisés au travers d'ontologies avec les modèles probabilistes associés et exploités pour les méthodologies d'extraction de contenu fondées sur des approches pilotées par les données et les scénarios préétablis. Les métadonnées obtenues sont incorporées aux systèmes de gestion de connaissance fournissant des possibilités de requêtes (en ligne) sur la connaissance ainsi découverte.
CARETAKER est construit autour de huit partenaires européens : THALES Communications (France), Multitel (Belgique), INRIA (France), Kingston University (Angleterre), IDIAP (Suisse), ATAC et GTT (Italie), Solid (Finlande) et Brno University of Technology (République tchèque).
Le projet a été lancé au 1er mars 2006 pour une durée de trente mois, et a donc servi de projet support à la présente thèse.

3.2.1.2 Contribution

CARETAKER a fourni le premier corpus vidéo utilisé pour la validation des outils développés lors du passage de MPEG-2 à MPEG-4 Part 2. L'objectif de THALES, concernant la modalité vidéo, était principalement une étude de faisabilité concernant l'analyse dans le domaine compressé dans un contexte concret, avec l'utilisation d'une infrastructure complexe existante. Les résultats attendus comportaient donc les masques de segmentation des piétons dans les métros de Rome et Turin, avec des temps de calcul suffisamment faibles pour permettre de traiter plusieurs flux en temps réel sur une même plateforme informatique.

3.2.2 Infom@gic

3.2.2.1 Présentation

Tableau 16 – Projet INFOM@GIC

Nom du projet	Infom@gic
Date de début de projet	Juin 2006
Date de fin de projet	Juin 2009
Type de projet	Pôle de compétitivité CAPDIGITAL
Partenaires	21 dont THALES, l'ONERA, TELECOM Paristech et EADS pour le sous projet UrbanView.

Infom@gic est un projet visant à l'étude d'un moteur de recherche multimédia comportant plus de 20 partenaires. Les données multimédias concernent le texte, le son, l'image et la vidéo. Parmi les partenaires, THALES, l'ONERA, TELECOM Paristech et EADS sont les contributeurs du cas d'usage UrbanView. Ce sous-projet s'est focalisé sur l'extraction de métadonnées dans un but d'investigation à l'aide de caméra de vidéosurveillance urbaine. EADS Innovation Works était responsable du sous-projet et apportait sa contribution en termes d'indexation hors ligne (sans contrainte de temps réel), de recherche dans la base de données créées, ainsi que l'intégration des différents applicatifs des partenaires dans le démonstrateur. L'ONERA a mené des études sur la super-résolution, permettant d'obtenir des images de tailles importantes à partir des vignettes consécutives d'un même objet, ainsi que les signatures et caractérisation 3D des objets. TELECOM ParisTech a pour sa part proposé un outil de mesure de similarité et d'appariement d'objets par descripteurs visuels, autorisant entre autre le suivi multi-caméras.

3.2.2.2 Contribution

Dans le cadre du projet Infom@gic, la contribution de la présente thèse s'est concentrée sur la possibilité de lier l'analyse dans le domaine compressé au cadre de l'investigation avec interrogation de larges bases de données. Là où l'équipe d'EADS partait sur l'hypothèse d'une indexation réalisée préalablement à toute requête, nous avons proposé d'exploiter la capacité d'analyse accrue du domaine compressé pour traiter aussi vite que possible des vidéos fraîchement obtenues. L'infrastructure déployée sur le terrain, par exemple par une municipalité, serait ainsi capable d'indexer en temps réel le contenu filmé. En cas d'incident, des séquences pourraient être collectées auprès d'entreprises, commerces ou particuliers ayant des caméras à proximité de la zone. Le démonstrateur global intègre donc la possibilité de lancer une recherche en direct à partir d'un fichier vidéo pour en extraire les véhicules répondant aux différents critères de recherche (détails paragraphe 3.5). Nous avons ainsi fourni un applicatif prenant en paramètres un nom de fichier à analyser couplé à une requête de recherche comportant une zone dans le champ de la caméra, les images de début et fin de recherche, ainsi que la couleur du véhicule recherché. La sortie est une liste de véhicules, avec les vignettes correspondantes extraites de la séquence.
C'est également dans le cadre de ce projet qu'a été développé le parseur MPEG-4 Part 2 dédié à l'analyse dans le domaine compressé et que les outils ont été portés depuis MPEG-2.

3.2.3 Vanaheim

3.2.3.1 Présentation

Tableau 17 – Projet VANAHEIM

Nom du projet	VANAHEIM
Date de début de projet	Février 2010
Date de fin de projet	Juillet 2013
Type de projet	Européen, FP 7/2007-2013 n°248907
Partenaires	Multitel ASBL(Belgique), THALES Communications (France), INRIA (France), IDIAP (Suisse), Thales Italia (Italie) Gruppo Torinese Trasporti – GTT (Italie) RATP (France) University of Vienna (Austriche).

VANAHEIM, pour *Video/Audio Networked surveillance system enhAncement through Human-cEntered adaptIve Monitoring*, est un projet dans la lignée du projet CARETAKER. Egalement financé par l'Union Européenne, il regroupe 8 partenaires pour étudier et intégrer des outils audio et vidéo innovants dans les réseaux de surveillance utilisés dans les environnements de transports urbains (stations de métro et de train).
Les partenaires abordent les problématiques de recherche sur les traitements par ordinateurs de l'audio et de la vidéo (Multitel, IDIAP, INRIA, THALES Communications), de mise au point de système de surveillance (THALES Italia), d'opérateurs de transport public (GTT, RATP), et d'aspects éthiques et éthologiques (Université de Vienne).
Les développements s'articulent autour de trois thèmes :

- La sélection automatique de capteur pour la supervision des murs vidéos : il est aujourd'hui impossible pour les opérateurs de surveiller efficacement l'ensemble des flux captés. A Turin par exemple, 28 moniteurs sont disponibles pour afficher les 800 flux produits par les caméras. La plupart du temps, les scènes affichées sont vides, alors que d'autres caméras filment des zones d'activités (généralement sans incident). La probabilité de regarder le bon flux au bon moment est très faible. L'objectif du projet est de mettre au point des modèles caractérisant le contenu des flux vidéo, depuis la tâche simple de sélection de scènes prioritaires, *i.e.* présentant une activité face à celles sans mouvement, jusqu'à la sélection de vues lorsque toutes présentes une activité. Dans le cas de l'audio, le besoin de sélection non-supervisée est encore plus critique, puisque le principe de mosaïque ne peut pas s'appliquer à l'audio du fait de la nature 'transparente' du son.

Figure 46 – En vidéosurveillance, trop d'information peut désservir les opérateurs.

- L'analyse statistique sur le long terme pour ajouter une planification des paramètres des applications : à l'heure actuelle, la modélisation du comportement humain n'est pas adaptée à des environnements à l'échelle 1. La compréhension de scènes s'appuyant sur des caractéristiques de localisation n'est pas suffisamment aboutie, et le besoin de caractérisation robuste centrée sur les personnes persiste. VANAHEIM étudiera donc les possibilités offertes par trois niveaux d'analyse : à l'échelle de l'individu (détection et description d'une personne et de ses activités), à l'échelle d'un groupe (détection de petits groupes de personnes et des interactions entre leurs membres), et enfin à l'échelle d'une foule (modélisation des flots de personnes). L'objectif est ainsi la détection d'événements pour la sécurité, et la remontée de rapports pour la prise en compte de la situation en temps réel.

Figure 47 – Selon les horaires, la fréquentation des transports varie fortement, demandant une éventuelle adaptation des outils de modélisation.

- La surveillance centrée sur les personnes à partir d'analyse audio et vidéo : aujourd'hui les régies de transport doivent faire face aux problèmes de capacité, et les gestionnaires expriment le besoin d'analyser la dynamique des passagers, problématique en raison de la complexité et variété de leurs comportements. Le projet vise donc à identifier et caractériser des structures dans le comportement collectif. En étudiant en continu des données telles que les emplacements, les trajectoires, les activités spatiotemporelles (marche, attente, …), les interactions entre passagers et équipements, ainsi que les données contextuelles (période de la journée, densité d'utilisateurs, …), l'objectif est d'estimer des tendances à l'échelle de toute l'infrastructure (identification des allées les plus usitées, les zones qui restent peu fréquentées, les trajets types dans l'infrastructure, …).

Figure 48 – L'objectif de VANAHEIM est d'améliorer les outils déjà en place pour les régies de transport.

3.2.3.2 Contribution prévue

Dans le cadre du projet VANAHEIM, le laboratoire MMP de THALES s'appuie sur son expertise duale en audio et vidéo et propose une approche fondée sur une analyse conjointe audio et vidéo bas niveau. L'objectif est d'analyser la corrélation entre les deux modalités, et leur capacité à fournir un modèle de l'activité et des comportements servant de point d'entrée aux caractérisations haut-niveau des autres partenaires.
Pour la vidéo, l'analyse dans le domaine compressé visera à segmenter rapidement les objets mobiles et à améliorer l'existant vis-à-vis des objets de petite taille et ceux présentant une faible vitesse de déplacement. La reconstruction des séquences basse-résolution sera donc ici enrichie pour exploiter au mieux les différentes caméras fixes des régies de transport. La solution de suivi sera rendue plus robuste pour lisser la segmentation dans le temps et éviter fausses alarmes et non-détections.

Concernant l'analyse multimodale, l'objectif est, à partir de l'étude de la corrélation entre audio et vidéo, de déterminer dans un premier temps le type d'événement qui peut être détecté et caractériser de manière fiable. A partir de cette étude, l'extraction de motifs à grand horizon temporel sera pris en compte, afin d'obtenir par exemple des comportements différents des outils selon les horaires et la fréquentation des trains.

3.2.4 Les projets internes à THALES

3.2.4.1 Contexte

En dehors de l'utilisation étudiée à travers les différents projets collaboratifs, liée à un contexte de vidéosurveillance avec un réseau de caméras fixes déjà déployées sur le terrain, l'analyse vidéo dans le domaine compressé a aussi fait l'objet d'une proposition d'intégration pour les caméras Margot 3000V de THALES. Il s'agit de caméras dans le domaine visible et infrarouge ou thermique, pouvant par exemple être employées pour la protection des frontières ou la lutte contre le trafic de stupéfiants (les échanges étant couramment réalisés de nuit). Différentes solutions concurrentes existent, et les différents industriels cherchent à enrichir leurs offres avec des fonctionnalités supplémentaires.

3.2.4.2 Proposition

Dans ce cadre, le laboratoire MMP a proposé son expertise sur la compression vidéo (H.264 en l'occurrence) pour permettre la transmission sans fil de vidéo depuis une ou plusieurs caméras vers un poste de commandement par exemple. Nous avons par ailleurs proposé d'intégrer une détection d'objet mobile au plus près de la caméra, dans un boîtier basse-consommation qui pourrait prendre en charge la compression et l'analyse conjointe.
A notre demande et en réponse à nos spécifications, un corpus représentatif a été réalisé, comprenant une caméra fonctionnant en *step and stare* : la caméra tourne d'un angle déterminé, puis s'arrête pour un laps de temps fixé, au court duquel est menée l'analyse de la vidéo. Différentes séquences, avec des personnes, véhicules et/ou animaux ont été acquises pour évaluer les outils dans ce contexte.
La contribution comprend par ailleurs l'adaptation des outils au standard H.264, ainsi que l'intégration de l'analyse dans le codeur vidéo, permettant de s'affranchir de l'étape de *parsing* du flux. Au final l'extraction de zone en mouvement n'ajoute qu'un très faible surcoût à l'étape de compression vidéo (6% en temps de calcul, négligeable en espace mémoire).

3.3 Les corpus exploités

L'objectif des présents travaux est avant tout de présenter un outil d'analyse dans le domaine compressé capable de traiter une majorité de flux représentatifs des solutions déployées de vidéosurveillance. Outre la prise en charge des différents standards évoqués, cela couvre également le contenu des séquences. Un point d'orgue a été mis sur la validation des outils développés sur un corpus le plus large possible. Cet aspect comprend ainsi des séquences d'intérieur et d'extérieur, avec des piétons, véhicules ou animaux, dans un contexte public ou privé, depuis des autoroutes ou transports en commun jusqu'à des bureaux, et enfin dans le domaine visible ou infrarouge / thermique.

3.3.1 Scènes d'intérieur

Une séquence a été acquise spécifiquement pour évaluer la pertinence des solutions de filtrage sur les vecteurs d'estimation de mouvement. Il s'agit d'un plan fixe réalisé dans le laboratoire MMP. Les seules variations de l'image sont liées aux conditions d'éclairage (très stables sur 15 minutes) et surtout au bruit de capteur. Le codeur vidéo a comme attendu sollicité l'estimation de mouvement sur les zones d'à-plats, alors qu'aucun déplacement réel n'avait lieu.

Figure 49 – Aperçu de la séquence plan fixe

Tableau 18 – Plan fixe

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Fixe_1	1	15 minutes	1 Mb/s	220 Mo

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-2 & MPEG-4 Part 2	Reference softwares	SP (MPEG-4)	12 images IBBP…	720x576 Progressif	25fps

Parmi les séquences d'intérieur, celles du projet Caretaker proposent des acquisitions faites dans le métro de Rome. Cinq caméras ont été mises en place pour le projet, et des acquisitions ont été effectuées, dont une campagne sur 24 heures en continu. Les vues couvrent un quai, ses accès via escaliers et escalators, couloirs et plateformes avec les tourniquets de validation de titre de transport, ainsi qu'illustré Figure 50. Le corpus comprend des séquences « naturelles », avec des usagers des transports, et des séquences « actées », réalisées pour le projet, avec des événements comme du vandalisme sur les distributeurs, altercation entre deux personnes, resquillage (saut des tourniquets), etc. Ces séquences ont été utilisées par certains partenaires pour de la détection d'événements anormaux et levée d'alarmes.

Tableau 19 – Corpus Caretaker – métro de Rome

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Care_1	5	160 heures	300kb/s	170 Go

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-4 Part 2	Carte de compression IEI IVC8371P	ASP	64 images IPPP…	720x576 Progressif	6fps

Figure 50 – Aperçu du corpus Caretaker – Rome

Les séquences du métro de Turin, toujours dans le cadre de Caretaker, tirent avantage de l'infrastructure disponible. Les transports ont été dimensionnés en taille et dispositifs de sécurité pour les Jeux Olympiques d'hiver 2006. Toutes les stations sont construites selon la même architecture, et toutes équipés de caméras et micros (pour un total de 534 caméras). Les vues couvrent les accès depuis la surface, les différents couloirs, escaliers ou escalators, les distributeurs et guichets, les quais et même les tunnels (Figure 51).
Le matériel vidéo et réseau permet de s'interfacer directement au système de surveillance complet, et de cibler la ou les caméras pressenties. Certaines scènes du corpus ont été extraites préalablement car présentant un intérêt pour la régie de transport (présence de nombreux vélos, personne qui danse sur le quai, voitures qui descendent les escaliers d'accès à la station – dans le cadre du tournage d'une publicité, etc.).

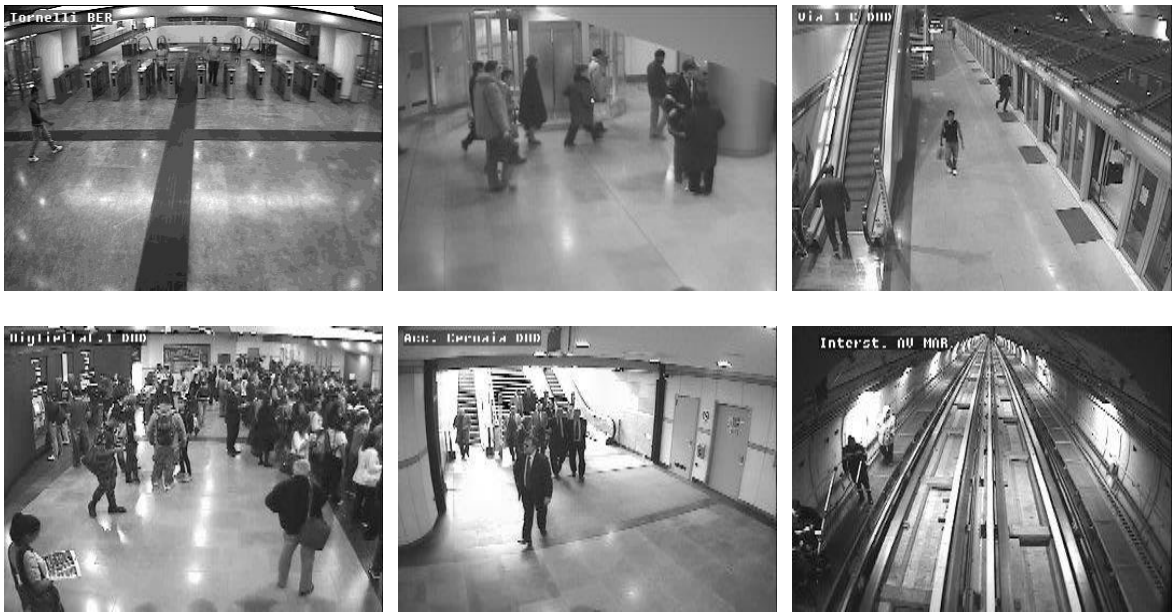


Figure 51 – Aperçu du corpus Caretaker – Turin

Tableau 20 – Corpus Caretaker – métro de Turin

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Care_2	534	55 heures	400kb/s	75 Go

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-4 Part 2	Codeur vidéo Funkwerk	ASP	25 images IPPP…	704x576 Entrelacé	6fps

3.3.2 Scènes d'extérieur

La première séquence à avoir été utilisée au tout début des présents travaux est la séquence *Speedway*, réalisée par l'équipe du Pr. Benoît Macq de l'Université catholique de Louvain La Neuve. A partir de la séquence non compressée, deux codages ont été réalisés, d'abord en MPEG-2 lors de la première étude de faisabilité, puis en MPEG-4 Part 2 lors de la migration des outils. Cette séquence, filmée depuis un pont au dessus d'une autoroute, permet de tester à la fois la détection des objets mobiles (des véhicules en l'occurrence), ainsi que les outils de filtrage face au vecteurs d'estimation de mouvement présents sur les zones d'à-plats (route) ou fortement texturées (buissons principalement).

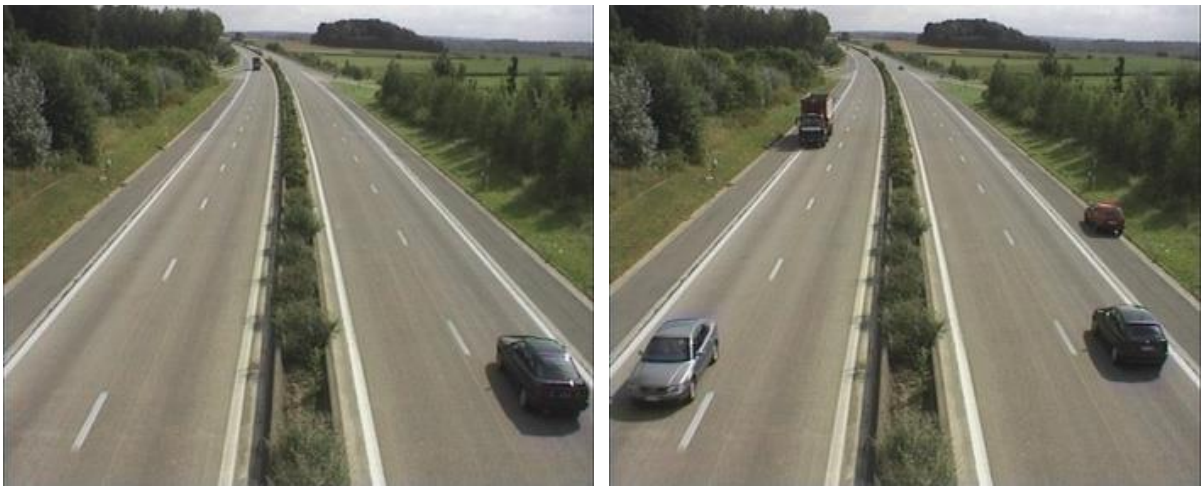


Figure 52 – Aperçu de la séquence Speedway

Tableau 21 – Séquence Speedway

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Speed_1	1	10 heures	3 Mb/s	12 Go

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-2 & MPEG-4 Part 2	Reference software	Baseline	12 images IBBP…	720x576 Entrelacé	25fps

Deux corpus ont été utilisés dans le cadre du projet Infom@gic. Le premier a été réalisé spécifiquement pour le projet, à l'aide de deux caméras placées à environ 6m de hauteur filmant un même carrefour avec une incidence de 30° environ, et une troisième caméra filmant une rue voisine, permettant entre autre aux partenaires de chercher à fusionner les véhicules d'une vue à l'autre. Le placement des caméras est représentatif des vues obtenues lors de l'utilisation de caméras sur mâts ou sur façade d'immeuble, cas le plus courant en vidéosurveillance extérieur. Ce corpus n'étant pas libre de droit, il ne sera pas visuellement présenté, toutefois la validation des outils d'analyse a également été conduite sur ces séquences.

Tableau 22 – Corpus Infom@gic

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Inf_1	3	90 heures	2, 3 & 4 Mb/s	360 Go

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-4 Part 2	MPEG-4 Reference software	Baseline	25 images IPPP…	704x576 Entrelacé	25fps

Le second corpus du projet Infom@gic est le corpus NGSIM, pour Next Generation SIMulation [NGSIM], qui propose des séquences acquises en extérieur sur une zone étendue spatialement. Il s'agissait à l'origine de proposer un corpus libre de droit servant pour la modélisation des flots de circulation. A termes, c'est devenu un outil de tests et d'évaluation idéal pour les algorithmes liés à la vidéosurveillance. Les séquences utilisées appartiennent à la sous-partie Peachtree. Il s'agit d'une artère d'Atlanta (Figure 53) qui a été filmée par 8 caméras synchronisées, placées au niveau du toit d'un gratte-ciel (deux fois 15 minutes). Le corpus étant initialement en JPEG, chaque vidéo a été compressée pour ces travaux en MPEG-4 Part 2 à 2, 3 et 4 Mbit/s, afin de mesurer l'impact de débits différents sur l'analyse dans le domaine compressé.

Figure 53 – Peachtree street, artère filmée par le corpus NGSIM

Figure 54 – Aperçu du corpus NGSIM – Peachtree : 8 caméras filmant une même artère

Tableau 23 – Corpus NGSIM Peachtree

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Inf_2	8	12 heures (2 fois 15 minutes sur 8 caméras, à 3 débits différents)	2, 3 & 4 Mb/s	16 Go

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
MPEG-4 Part 2	MPEG-4 Reference software	Baseline	25 images IPPP…	640x480 Progressif	25fps

3.3.3 Séquences dans l'infrarouge et/ou le thermique

L'utilisation de l'analyse dans le domaine compressé sur des vidéos infrarouges et/ou thermiques correspond à l'étude liée à la Caméra Margot 3000V de THALES. Les séquences comportent des scènes d'extérieur, avec des plongeurs, hélicoptères, piétons / joggeurs ou même des hyènes. Une partie du corpus présente des séquences non stabilisées, une autre avec les phases de *step and stare* décrites précédemment, et enfin la dernière partie comprend des plans à partir de caméras fixes.

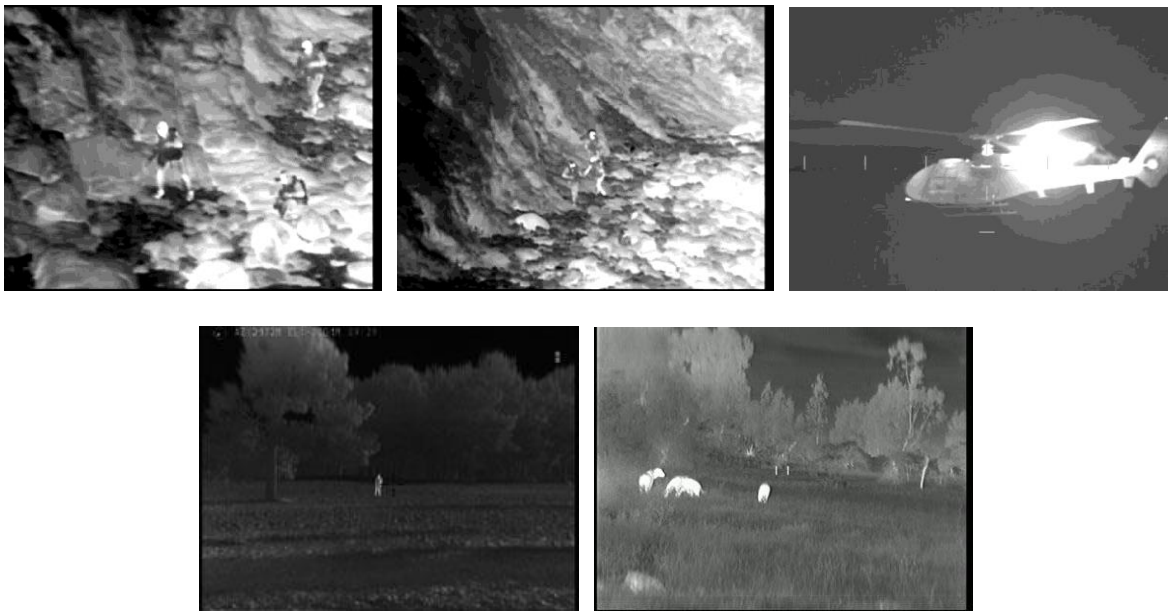


Figure 55 – Aperçu des vidéos en imagerie thermique.

Tableau 24 – Séquences infrarouges ou thermiques

Référence corpus	Nombre de caméras	Durée	Débit	Volume de données
Margot	1, nombreuses séquences	Séquences de quelques secondes à plusieurs dizaines de minutes	1 Mb/s	Confidentiel

Standard vidéo	Codeur vidéo	Profil	Format de GOP	Format	Frame rate
H.264	Codeur THALES MMP	Baseline	25 images IPPP…	640x480 Progressif	25fps

3.4 Résultats de segmentation

Bien que les différents corpus de validation des outils présentent une grande variété dans leur format, leurs options, les objets filmés et leurs caractéristiques, les points forts et points faibles qui se dégagent de l'analyse dans le domaine compressé sont communes aux différentes vidéos de tests.

3.4.1 Ce qui fonctionne

3.4.1.1 La prise en charge des différents standards

L'approche visant à mettre au point une structure unifiée commune à l'analyse des flux MPEG-2, MPEG-4 Part 2 et H.264 s'est avérée payante, puisque les mêmes outils de reconstruction des données (coefficients transformés et vecteurs) sont directement utilisables et fournissent des résultats corrects avec le même jeu de paramètres (qui peut être affiné selon la vitesse et la taille des objets segmentés) : lors de notre premier essai sur le corpus Rome après avoir travaillé sur le corpus Turin, les non-détections étaient inférieur à 10% avant tout paramétrage.
Dans le cadre des vidéos thermiques, l'approche est toutefois un peu différente de la méthode globale présentée dans le chapitre précédent : l'étude technique a été réalisée avec pour objectif d'intégrer le logiciel d'analyse à la brique de compression H.264 (Figure 56).

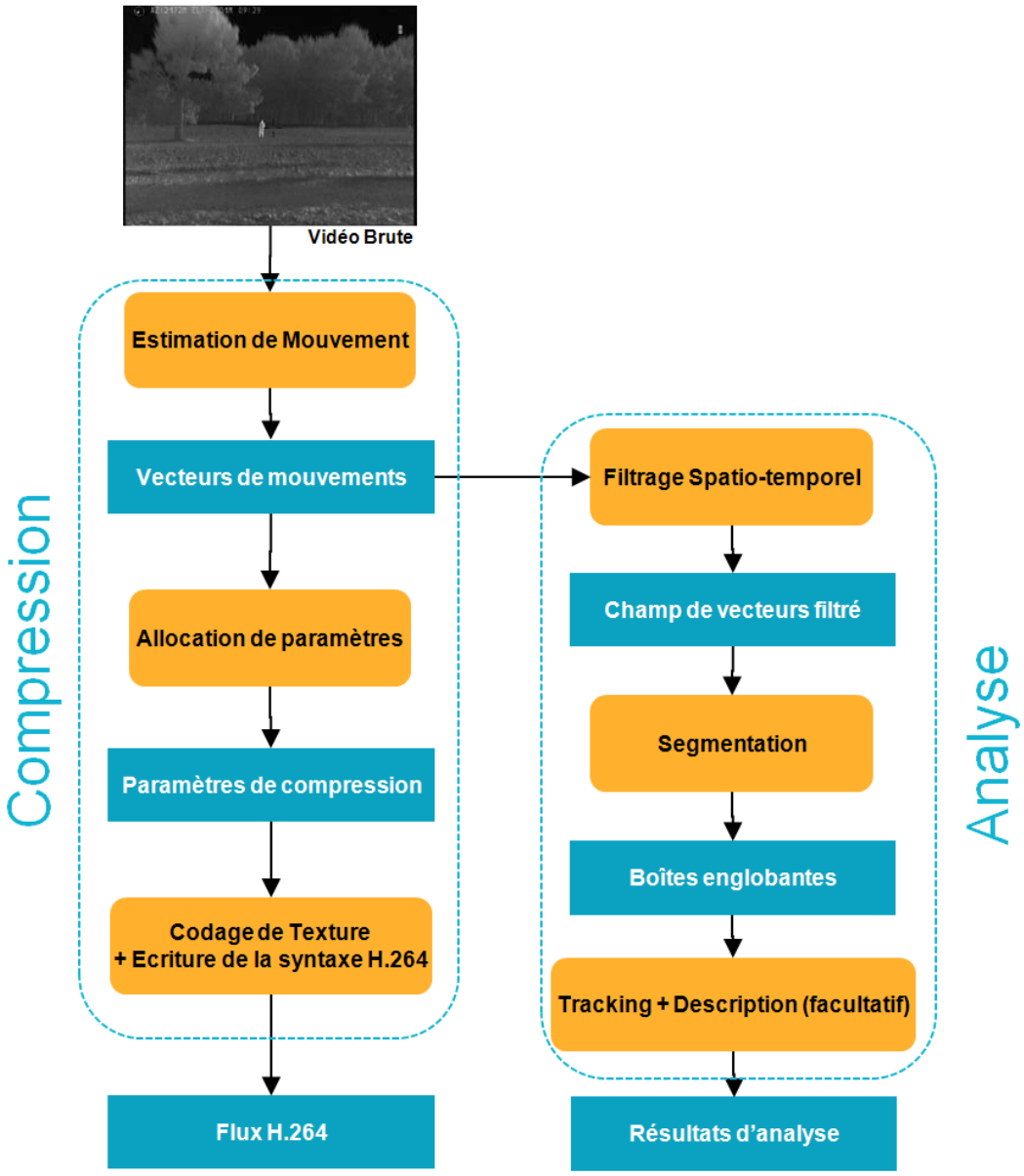


Figure 56 –Compression et analyse conjointe de vidéo infrarouge

La segmentation se fait donc uniquement sur les vecteurs d'estimation de mouvement. Cette modification, de par la structure modulaire de l'application développée, a été facilement réalisée. Concrètement, les modules utilisés ici sont donc seulement les MEG et OMF. Le parseur n'est en effet pas utile puisque les informations sont directement transmises avant le codage CABAC ou CAVLC, et les modules s'appuyant sur les coefficients transformés (LRD, LROS, CD) sont également désactivés.

3.4.1.2 Le filtrage des vecteurs

Dans le cadre de la séquence 'plan fixe', des vecteurs étaient initialement présents sur les trois principales zones d'à-plats (fenêtre, mur et sur-bureau, Figure 57). Le filtrage mis au point par utilisation de cartes de confiance permet de supprimer 98% de ces vecteurs. Ceux restants sont éliminés après segmentation par morphologie mathématique, puisqu'ils représentent des blobs de petites tailles isolés les uns des autres. Au final, aucun objet n'est segmenté sur la séquence 'plan fixe'.

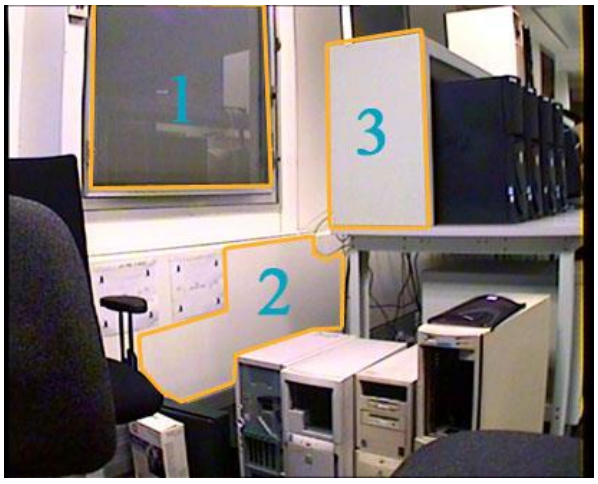
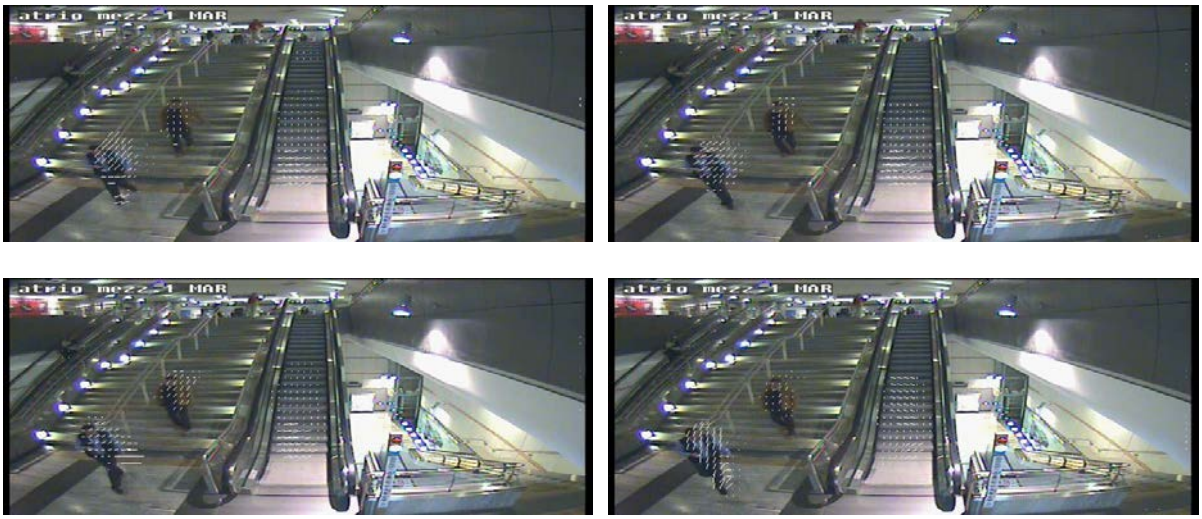


Figure 57 – Principales zones présentant des vecteurs bruités

Sur les autres séquences des corpus, quelques fausses-alarmes persistent, notamment sur les séquences thermiques. Celles-ci, principalement imputables aux petits mouvements de caméras, seront détaillées en 3.4.2. D'une manière générale, le filtrage des vecteurs permet d'obtenir des zones connexes présentant un mouvement cohérent, et reste homogène au cours du temps.



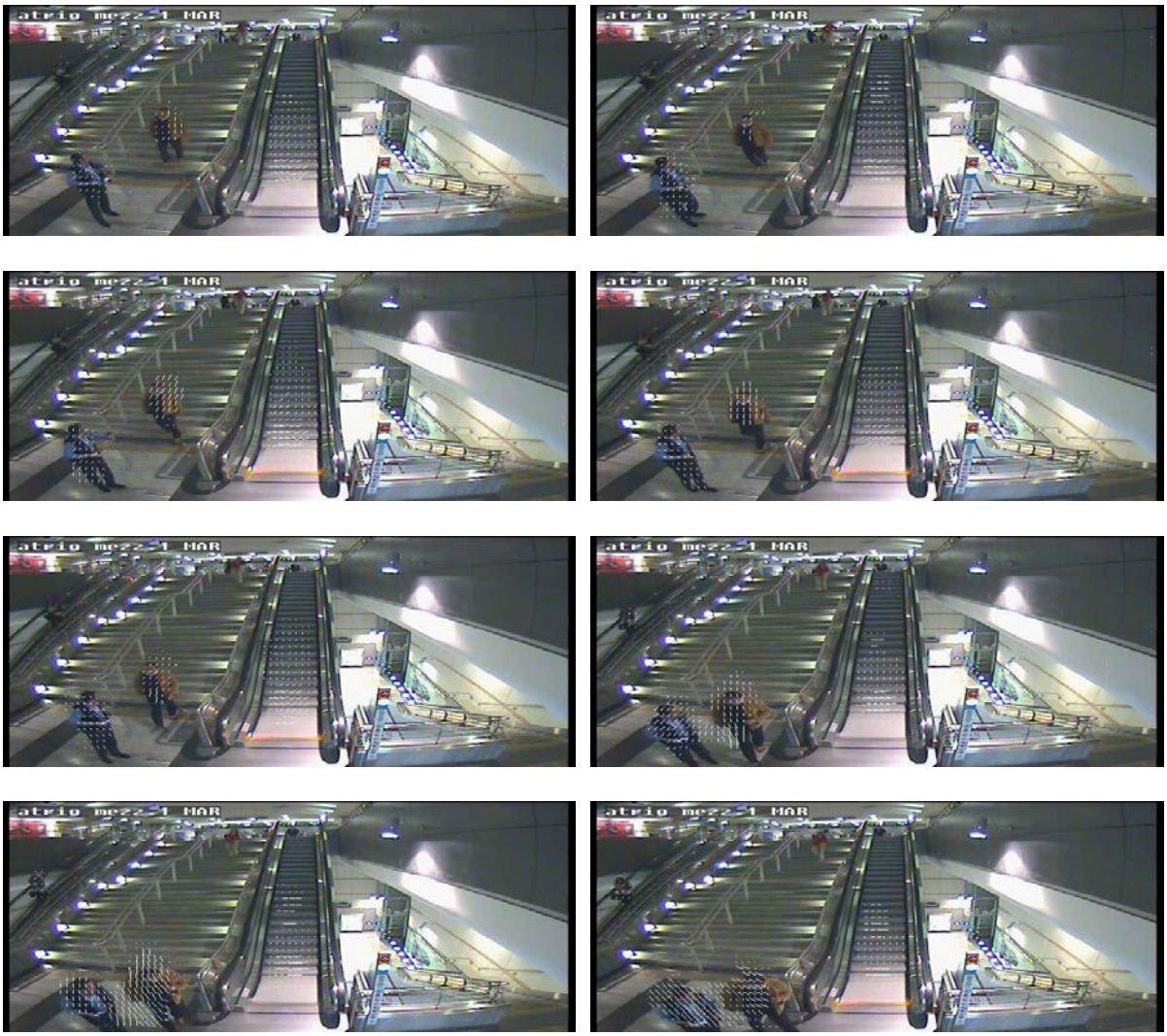


Figure 58 – Aperçu des vecteurs filtrés sur une séquence

3.4.1.3 La détection d'objets suffisamment grands et rapides

L'objectif premier était de pouvoir traiter les séquences de vidéosurveillance des projets CARETAKER et INFOM@GIC. Même si les objets sont de natures différentes, à savoir des véhicules en extérieur pour l'un, et des piétons en intérieur pour l'autre, le placement et les caractéristiques physiques des caméras permettent d'obtenir des comportements homogènes des objets mobiles dans les deux cas : en étant placé beaucoup plus près des utilisateurs des transports en intérieur que des véhicules en extérieur, la détection se fait sur des blobs de taille globalement comparable (de trois à une dizaine de blocs de hauteur et/ou largeur) et des vitesses également du même ordre de grandeur (de 5 à 50 pixels par image dans la majorité des cas). Ainsi les figures ci-dessous illustrent-elles les contours d'objets obtenus dans des cas variés.

Pour des objets suffisamment loin les uns des autres (2 blocs), la segmentation est par ailleurs cohérente au cours du temps, ce qui permet de suivre un objet mobile depuis son entrée dans le champ de la caméra jusqu'à sa sortie (Figure 59).

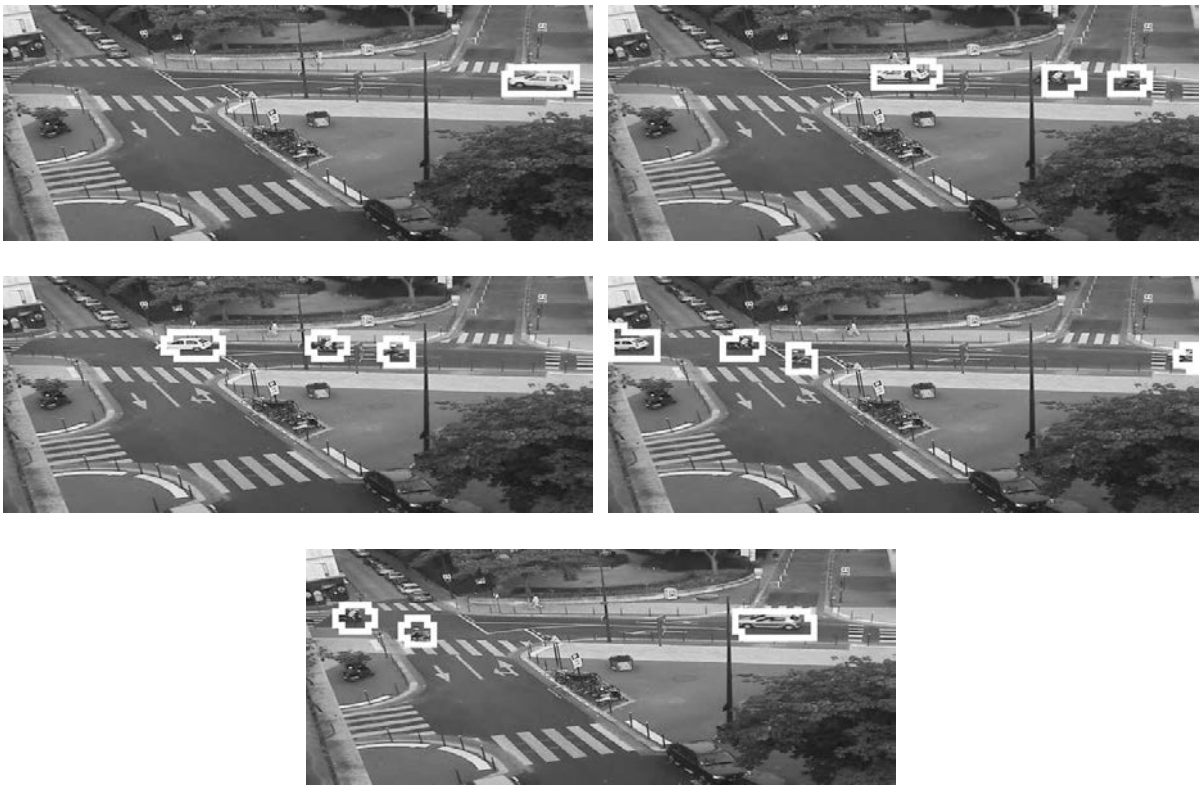


Figure 59 –Détection de véhicules (contours directement obtenus après segmentation) – Consistance de la segmentation au cours du temps.

Dans le cas des séquences provenant du métro de Turin, le marbre et les reflets qu'il génère a été problématique pour l'ensemble des partenaires du projet CARETAKER. En plus de l'objet, sa réflexion et son ombre s'ajoutent généralement au blob segmenté, selon les vues. La Figure 60 illustre ce point, avec des blobs déformés en bas à droite pour la personne au premier plan. Si en cas de (très) faible fréquentation, lorsqu'un petit nombre de personnes est présent dans le champ, ce problème n'introduit pas de fusion des objets segmentés, il peut en revanche induire des segmentations de groupes de personnes et non d'individu en cas d'affluence (cf 3.4.2). Capitalisant sur les avancés des partenaires sur ce point, et prenant en compte l'accroissement de complexité soulevé, nous n'avons pas adressé ce point plus en détail. Les résultats présentés par la Figure 60 sont donc représentatif de la segmentation en cas de reflets et d'ombres portées.

Figure 60 –Détection de piétons

3.4.1.4 La reconstruction de séquence basse résolution

En complément de la métrique détaillée dans le chapitre précédent (paragraphe 2.5.1), nous avons visualisé les résultats du LRD afin de juger subjectivement son comportement. Nous avons vu qu'il pouvait générer du flou dans le cadre de long GoP de structure IPPP… tels que ceux du corpus INFOM@GIC. *A contrario*, pour les séquences issues du métro de Turin (Figure 61), ces séquences sont plus facilement exploitables, et un poids plus fort a été attribué, au sein du module de décision coopérative, à la segmentation LROS. Cela est *a priori* dû au codeur hardware équipant les stations qui propose une meilleure estimation de mouvement, et donc aboutit à des séquences basses résolutions plus proches de la séquence décompressée (moins de phénomène de flou). Il est par ailleurs intéressant de noter que l'escalier mécanique présent sur la dernière image est détecté en continu (également visible Figure 58), mais sur les marches du bas seulement. Cela tient au fait que la texture devient plus petite avec la distance, diminuant les gradients dans les blocs, et aboutissant au filtrage par les cartes de confiance sur la moitié supérieure de l'escalier.

Figure 61 –Reconstruction du fond et détection de piétons

3.4.1.5 Les vidéos infrarouges ou thermiques

L'usage de H.264 permet ici de traiter des blocs 4x4, ce qui quadruple effectivement le nombre de données à traiter dans le domaine compressé, mais autorise en revanche une détection d'objets plus petits. Ainsi l'algorithme d'analyse a-t-il été paramétré pour pouvoir détecter des piétons en mouvement à 800m de la caméra (2 blocs conjoints). Cela tient compte des requêtes de l'utilisateur final, internes à THALES. Il est préférable, dans le cas d'activités critiques comme la surveillance de trafic de stupéfiants ou de frontières, de diminuer au maximum les non-détections, même si cela doit se faire au détriment du nombre de fausses alarmes. Piétons comme véhicules venant de face (à gauche Figure 63) sont ainsi détectés, même à grande distance dans la scène. Quelques images subsistent pour lesquelles les individus ne sont pas ou sont mal segmentés (à droite Figure 62), mais un filtrage temporel permet de lisser la détection et ainsi à la fois réduire non-détections et de fausses alarmes (voir 3.5.2.1). Sur une séquence, tous les individus se déplaçant sont segmentés : même si des non détections persistent sur certaines images, les objets mobiles sont correctement détectés (boîte englobante couvrant entre 80% et 150% de la surface de l'objet) sur deux tiers des images, ce qui assure une levée d'alarme en cas de mouvement suspect.

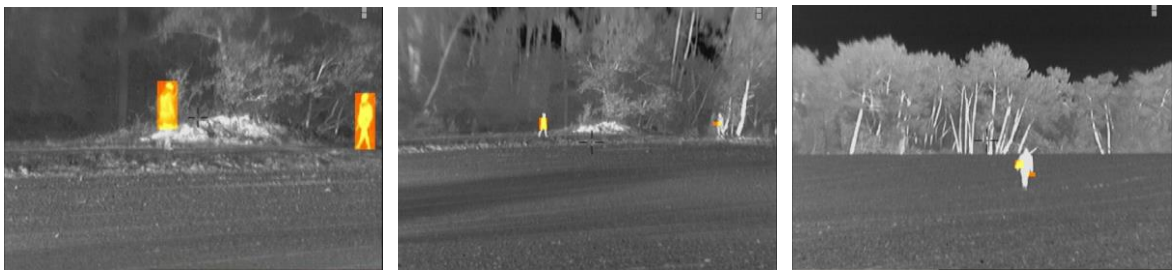


Figure 62 –Détection de piétons marchants en H.264

De par les caractéristiques de H.264 et du jeu de paramètres utilisés pour cette application, la taille minimum des objets segmentés est ici de 8x4 ou 4x8 pixels (deux blocs voisins), se déplaçant à une vitesse minimale de 2 pixels par image (seuil choisi pour permettre la détection des piétons à 800m). Des sur-segmentations persistent sur 5 à 15% des images (à droite Figure 62) ; d'autres ne cadrent qu'une portion d'un objet (au centre).
Le cadre applicatif s'appuyant sur un fonctionnement en *step and stare* de la caméra, il était important de présenter un temps d'initialisation minimal. Cela est entre autre rendu possible

par l'abandon de la reconstruction basse résolution, puisque l'absence de soustraction du fond implique qu'il n'est pas nécessaire de générer le modèle du fond. Pour réduire la latence, le filtrage temporel sur les vecteurs utilise les deux dernières images de la séquence, introduisant donc un temps d'initialisation de deux images, pour une latence finale d'une image (simplement compensée à l'affichage pour la surimpression des zones détectées).

Figure 63 –Détection d'objets de petites tailles en H.264. A gauche, piétons marchant à 800m et véhicules arrivant de face ; à droite, fausses alarmes liées à la stabilisation.

3.4.2 Ce qui échoue

3.4.2.1 La détection des objets immobiles ou lents

Si d'une manière générale, la segmentation par soustraction du fond sur les séquences basses résolution améliore la détection, certains cas persistent ou les non-détections persistent. Ainsi, dans le cas où un objet mobile serait déjà présent dans la scène au moment de l'initialisation du modèle, il restera non segmenté jusqu'à se mettre en mouvement. Ce phénomène n'est pas propre à l'analyse dans le domaine compressé, mais se manifeste à chaque fois qu'une méthode de comparaison image courante / modèle est employé.

Dans le cas des séquences vidéo thermiques, nous devions proposer une solution pour identifier une personne immobile (Figure 64). Les outils mis au point ne permettent pas de répondre à ce problème. Nous avons proposé une solution alternative de détection de 'points chauds' par un seuillage sur les valeurs les plus élevées selon leur voisinage, mais cette approche ne permet que de résoudre des cas précis comme les images ci-dessous. En cas de forte chaleur, l'individu se détachera beaucoup moins du fond (voir sera plus froid que l'environnement), et il arrive aussi de segmenter le soleil s'il est dans le champ de vision, ou des sorties de ventilation etc. D'une manière générale, l'analyse dans le domaine compressé telle que développée dans les présents travaux ne permet pas la détection d'objets immobiles, et demandera donc la mise au point d'outils connexes pour ces cas limites.

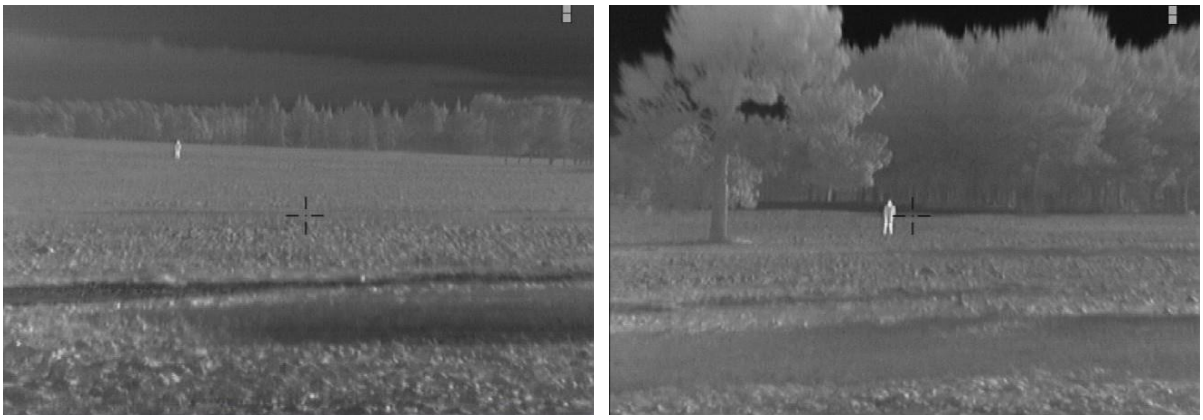


Figure 64 –Personne immobile en H.264 : sans mouvement, il n'y a pas de détection

Lors de l'utilisation d'un modèle du fond, une fenêtre temporelle de remise à jour du modèle est également utilisée : cela permet par exemple de suivre une voiture lorsqu'elle circule puis se gare, pour ensuite l'assimiler au fond après une période d'inactivité correspondant à la fenêtre temporelle. Dans le cadre de l'analyse dans le domaine compressé, en plus de cet aspect, la présence de nombreux blocs codés intra peut aggraver ce problème, principalement sur les séquences où la scène est globalement immobile. Ainsi, sur la Figure 65, les personnes qui attendent sur les images du haut sont bien identifiables dans les séquences reconstruites, mais ne sont pas pour autant segmentées. Sur les deux trames du dessous, les agents de sécurité à la sortie des tourniquets se déplacent très peu, et sont également assimilés au fond.

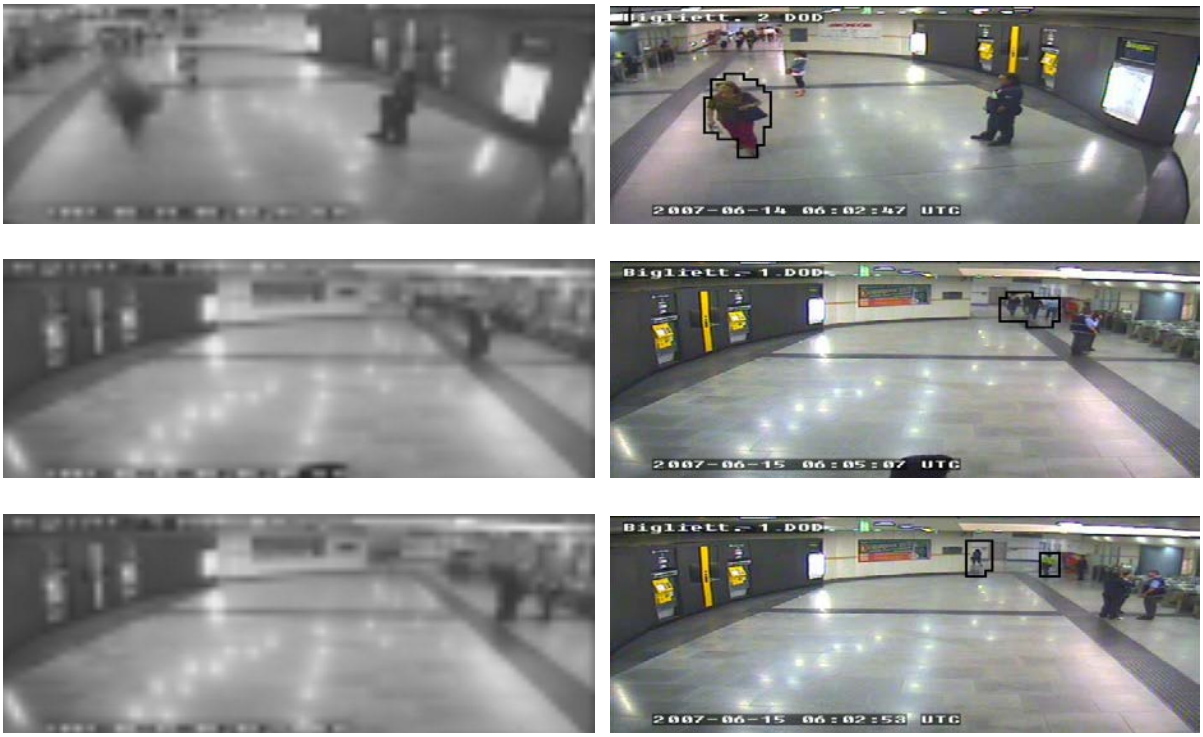


Figure 65 –Objets lents non détectés

3.4.2.2 Les mouvements de caméra

L'ensemble des algorithmes étudiés et développés ont été mis au point pour des caméras fixes. Des outils existent pour segmenter des objets mobiles face à une caméra elle-même

mobile, y compris dans le domaine compressé [Ewerth et al., 2007 ; Kas et Nicolas, 2009], mais le cadre applicatifs des différents projets collaboratifs liés à ces travaux ne requérait pas *a priori* d'alourdir les modèles et temps de calcul avec la compensation du mouvement de la caméra.
Toutefois, deux cas se sont présentés où la segmentation est prise en défaut. Pour le projet CARETAKER, la vue depuis le quai dans le métro de Rome est filmée depuis une caméra suspendue au plafond proche de la voie (Figure 66). A l'arrivée et au départ des trains, la caméra vibre, et toute l'image se déplace verticalement de plus ou moins 6 pixels par rapport à sa position 'au repos'. Des vecteurs d'estimation de mouvement sont présents sur toute la surface, et si le filtrage par carte de confiance permet de lever le doute sur certains blocs, celui par voisinage sur les valeurs des vecteurs puis la morphologie mathématique élargie au contraire ces grandes zones segmentées, et n'apporte donc pas de solution valide à ce problème.

Figure 66 –Segmentation en cas de vibrations

Le second cas vient des prises de vues en extérieur pour les séquences thermiques. Des mouvements de caméra horizontaux sont présents régulièrement, dus soit à l'opérateur, soit à des rafales de vent. Lorsque cela se produit, les zones à gradient horizontal marqué sont segmentées ; les autres sont filtrées par l'utilisation des cartes de confiance. Ce phénomène est illustré à droite Figure 63. Sur l'image de gauche Figure 67, les zones texturées du premier plan sont segmentées alors que sur la moitié haute elles ne le sont plus : ce phénomène est imputable à la valeur des coefficients AC_{01} plus marqués sur les zones d'herbes qu'en plein ciel.

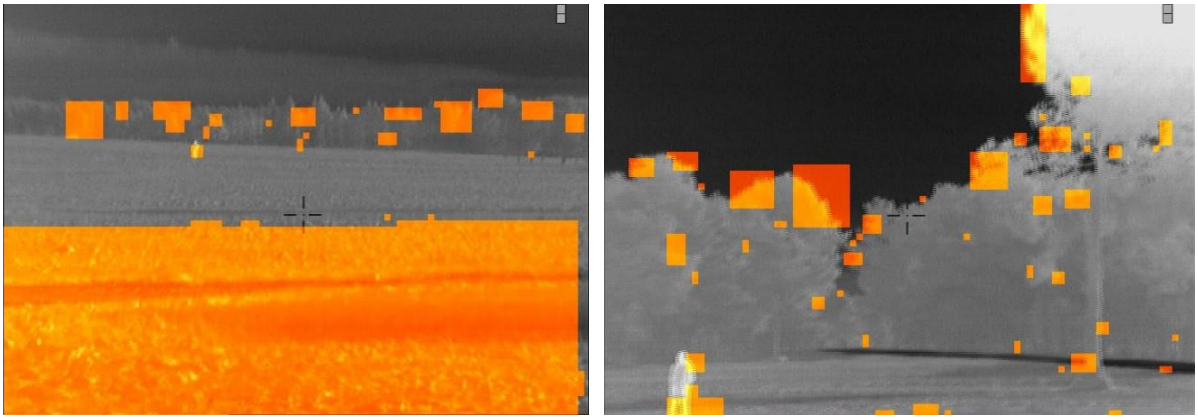


Figure 67 –Résultats de segmentation de personne immobile
(à gauche, caméra fixe ; à droite, caméra mobile)

3.4.2.3 Les groupes d'objets proches

En traitant les données du domaine compressé, l'algorithme mis au point réduit très fortement le volume de données en entrée. Le principal point est que ce ne sont plus des pixels mais des blocs qui sont pris en compte. Si d'un côté cette caractéristique permet d'atteindre des temps de calcul de l'ordre de 3ms par image, d'un autre elle implique une importante perte de précision sur les contours des objets. Ainsi lorsque deux objets sont proches l'un de l'autre, ils seront segmentés de manière groupée. Concrètement, cette distance minimum diffère selon les standards : pour MPEG-2, les objets doivent être écartés d'au moins trois blocs, alors que pour MPEG-4 Part 2 et H.264 seul un bloc doit séparer deux objets. Sur la Figure 68, les voitures à gauches présentes un mouvement homogène et sont suffisamment proches pour former, vis-à-vis de l'analyse dans le domaine compressé, un ensemble connexe se déplaçant ensemble, donc n'aboutit qu'à une seule segmentation.

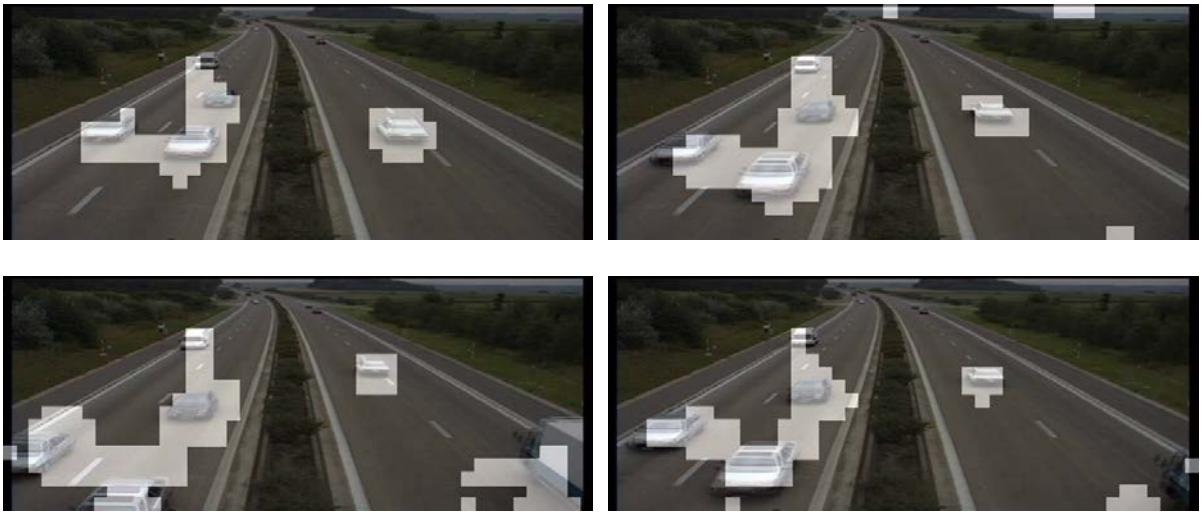


Figure 68 –Détection de véhicules proches (contours directement obtenus après segmentation) – MPEG-2

En revanche, à partir de MPEG-4 Part 2, la distance est réduite à un bloc. Cette amélioration *a priori* est due à la précision des vecteurs de mouvement accrue qui améliore la sélection de blocs lors de l'estimation de mouvement, et donc permet de distinguer plus facilement un bloc immobile au milieu de blocs mobiles. Dans le cas de zones denses en termes d'objets mobiles, cela ne suffit toutefois pas à distinguer les objets les uns des autres. Ainsi dans le métro de Rome aux heures de pointes, les outils proposés se contentent de suivre des groupes de personnes sans être capable de distinguer des individus (Figure 69).
A défaut de pouvoir suivre individuellement les piétons ou véhicules, cette segmentation par groupe d'objets permet tout de même de répondre à certains besoins exprimés (au paragraphe 3.2.3.2), tels que l'analyse de la dynamique à grande échelle (sur plusieurs heures, jours voire semaine) des flots et du mouvement pendulaire des personnes.

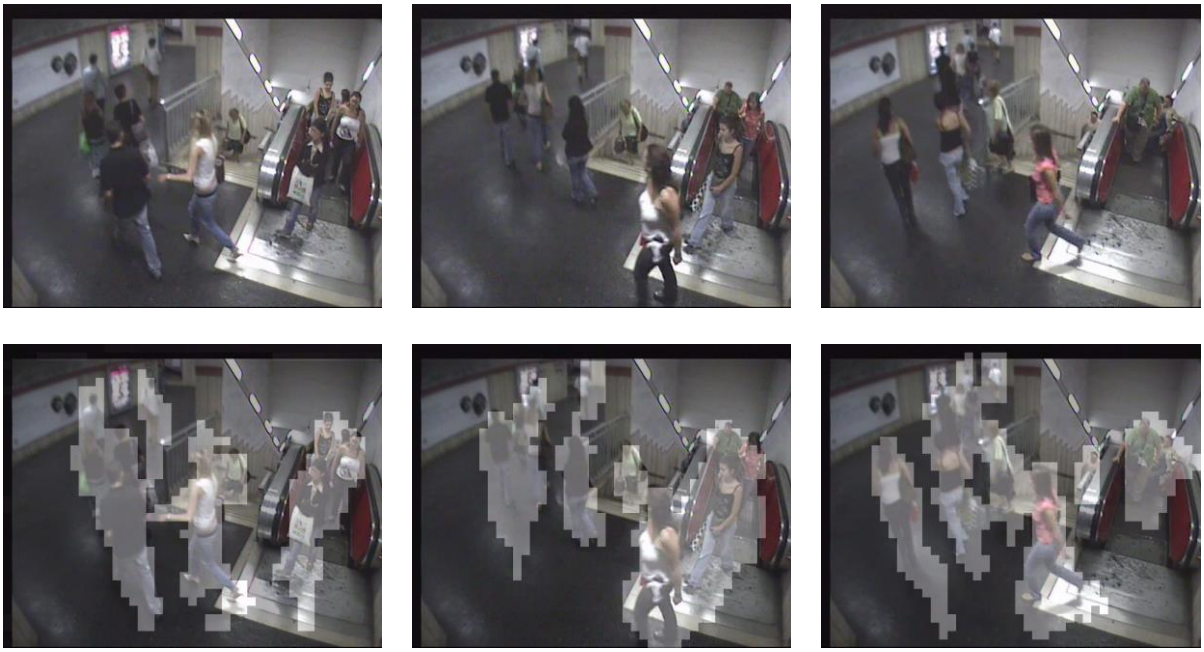


Figure 69 –Détection de piétons groupés (contours directement obtenus après segmentation) – MPEG-4 Part 2

La prise de vue peut faciliter la tâche de segmentation, mais des cas resteront toujours critiques. Ainsi pour le corpus NGSIM, le choix de caméras placées au sommet d'un gratte-ciel permet d'observer à grande distance les véhicules. Lorsqu'ils sont suffisamment éloignées les uns des autres (3 mètres sur ces vidéo), chaque voiture, bus ou camion est identifié individuellement (image de gauche Figure 70). En revanche, pour des véhicules se déplaçant dans la même direction sur des files voisines (image de droite), la segmentation fusionne ces voisins depuis leur entrée dans le champ de vision jusqu'à leur sortie (vitesse homogène de par la limitation à 30 mph en ville). Il est toutefois intéressant de noter que le véhicule blanc se déplaçant en sens inverse est pour sa part isolé du groupe immédiatement au-dessus. Cette particularité est le résultat du filtre médian qui dans ces cas a tendance à séparer des groupes de vecteurs de directions opposées.

Figure 70 –Détection de véhicules (contours obtenus après détection de boîtes englobantes) – MPEG-4 Part 2

3.4.2.4 Les petits objets

D'une manière générale, les codeurs vidéo ont tendance à compresser les mouvements de petits objets (un à deux blocs) par des blocs intra comme nous l'avons déjà évoqué, ou bien par l'intermédiaire des blocs résiduels. Il arrive malgré tout que certains de ces déplacements soit effectivement codés par un vecteur. Toutefois, pour éliminer une partie du bruit résiduel sur les vecteurs d'estimation de mouvement et la segmentation, les différents filtrages et morphologie mathématique mis en place suppriment les blobs d'un bloc de côté. Au final, que ce soit sur les véhicules au loin sur *Speedway*, les piétons sur le corpus INFOM@GIC (Figure 71) ou tout autre objet de petite taille, l'analyse dans le domaine compressé n'effectue pas de détection.

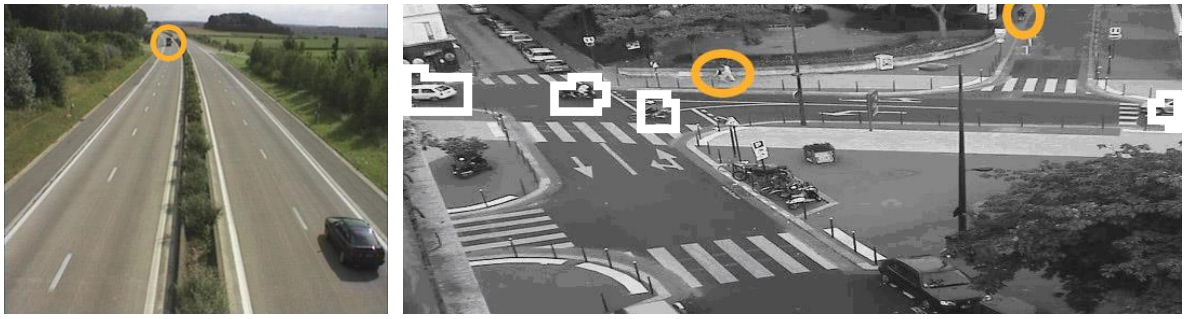
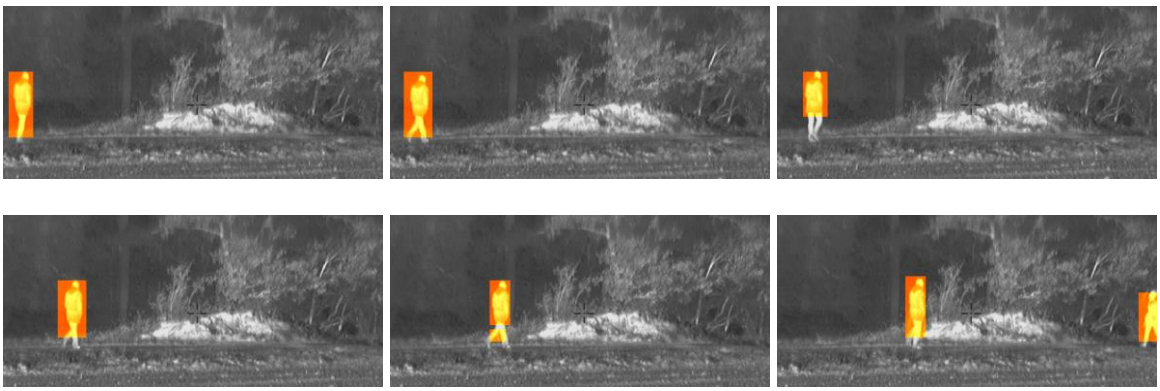


Figure 71 –Non détection des petits objets

3.4.2.5 La cohérence temporelle des segmentations

Ainsi que cela a été illustré par les figures page 97, pour des objets suffisamment grands et se déplaçant rapidement, la segmentation est cohérente au cours du temps du point de vue de la détection et de la forme du blob obtenu. Toutefois, lorsque des cas limites se présentent, nous avons vu que des non détections apparaissaient. Sur la longueur d'une séquence, ces manquements ne sont pas constants : des blobs sont présents sur certaines images, avec de la sur- ou sous-segmentation. Dans le cas de la Figure 72, si les piétons sont visuellement bien identifiés par la segmentation, d'éventuels post-traitements informatiques seront perturbés par la variabilité des taille de blobs (surtout lors de non détection des jambes), par la sur-segmentation (blobs indépendant pour le torse et les jambes, voire les pieds indépendants) et d'éventuelles non détection ou fausse alarmes lorsque la caméra bouge (image au centre).



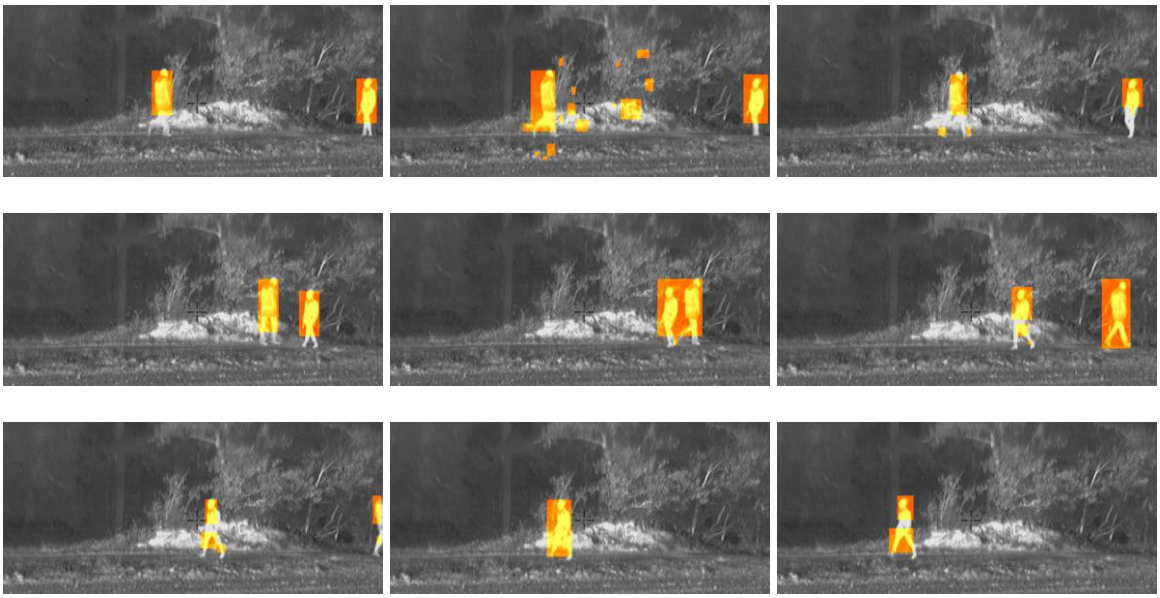


Figure 72 –Cohérence temporelle de la segmentation dans les cas critiques

Le fait de ne pas avoir de suivi d'objet ni de filtrage temporel sur la segmentation est ainsi à l'origine d'effets de scintillement lors de l'affichage (du aux changements de dimensions), de détections qui varient comme nous venons de l'évoquer, et enfin de bruit persistant dans les zones texturés, liés aux mouvements, même faibles, de la caméra.

3.4.3 Discussion intermédiaire

L'algorithme d'analyse dans le domaine compressé proposé dans le cadre des présents travaux permet de segmenter les objets mobiles couvrant au moins deux blocs de l'image, se déplaçant à au moins deux pixels par image, et n'est pas à l'origine de fausses alarmes intempestives sauf en cas de mouvement de la caméra. Sur une séquence, ces objets sont détectés à 100%, mais des non-détections persistent ponctuellement (moins d'une image sur vingt). Par ailleurs les blobs couvrent des zones variables sur ces objets, concrètement entre 80 et 150% de leur surface dans 90% des cas. Cela tient parfois à la sur-segmentation, notamment sur les jambes des personnes, ou parfois aux problèmes d'ombres et de reflets qui étendent les objets. Les différents modules algorithmiques fonctionnent indépendamment et surtout lorsqu'ils sont utilisés ensembles. La segmentation à partir des vecteurs d'estimation de mouvement permet à elle seule d'obtenir des taux de détection de l'ordre de 85% (lors des tests sur H.264), et la désactivation de la reconstruction des séquences basse-résolution diminue le temps de calcul de 18%. Ainsi pour des applications où le temps de traitement est le maître-mot, il sera envisageable de n'utiliser que les modules MEG et OMF en plus du parseur de flux.

Nous avons proposés des outils autour de notre approche pour des cas particuliers, comme la détection de points chauds, mais ces derniers restent des solutions pour des cadres applicatifs très précis. Dans le cadre du projet INFOM@GIC, nous devions répondre à la problématique d'investigations menées sur d'importantes bases de données. Dans cette optique, une méthode de suivi des objets était nécessaire. Outre l'identification de trajectoires pour les différents objets, le suivi nous a permis d'apporter le lissage temporel qui faisait défaut à nos outils. Ainsi, les blobs liés à la sur-segmentation se retrouvent liés au même objet, une homogénéisation des tailles de blobs est possible, et les non détections qui pouvait persister

sont compenser par l'appariement des blobs sur les images précédant et suivant les cas critiques.

3.5 Exemple de chaîne de traitement complète

Dans le cadre de la validation des traitements dans le domaine compressé lors du passage à l'échelle sur d'importantes bases de données, un démonstrateur a été réalisé répondant à un contexte d'investigations. L'objectif, répondant au projet Infom@gic, est d'identifier des objets mobiles au sein de vidéo compressées. L'application globale d'extraction de pistes sur requêtes est présentée Figure 73, et est détaillée dans les paragraphes suivants.

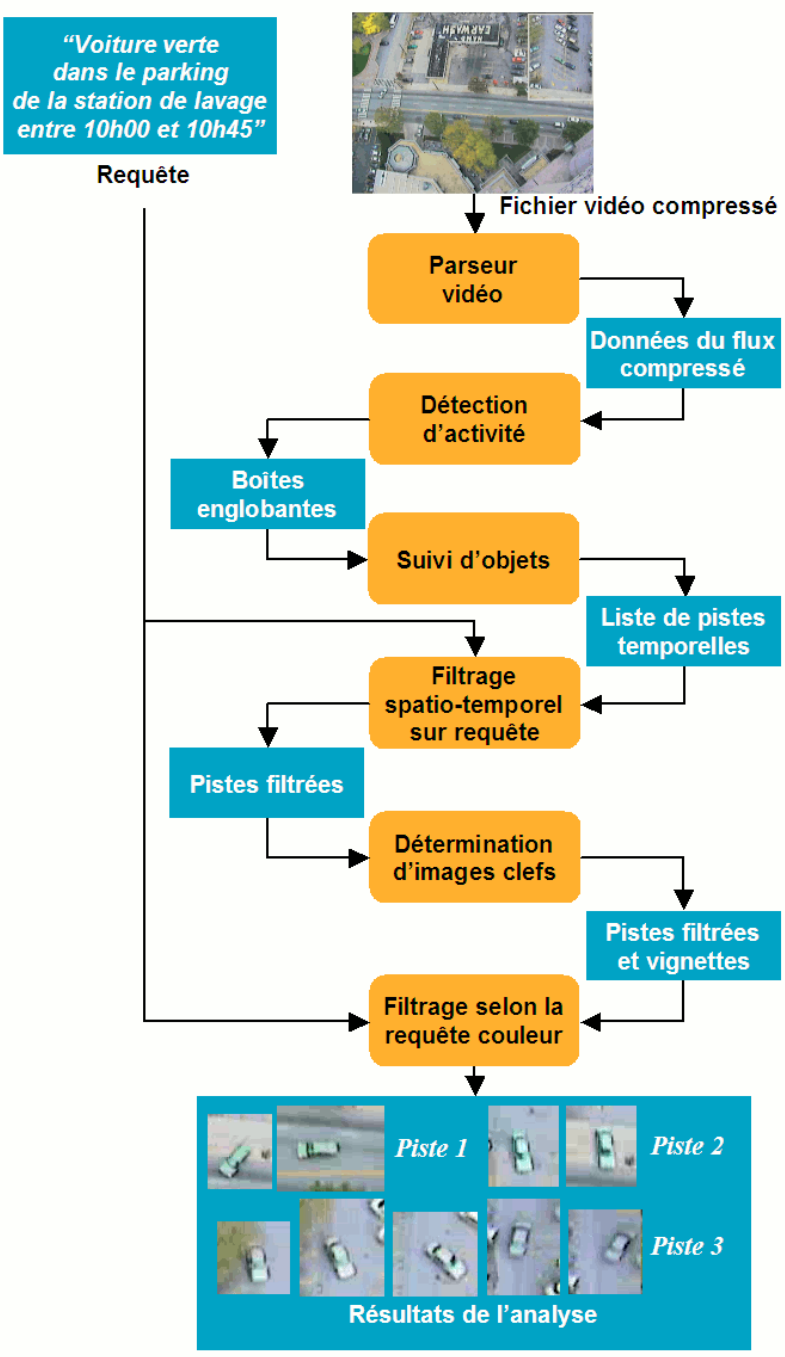


Figure 73 – Chaîne de traitement dans un but d'investigation

3.5.1 Interface de requête

Le scénario type support de cette application est le suivant : suite à un incident, une enquête est lancée sur un site. Les services d'ordre recueillent divers témoignages, et les vidéos enregistrées par les différents réseaux de surveillance couvrant la zone (caméras municipales, bancaires, installation dans les magasins ou stations services, etc.). Plutôt que de réquisitionner des agents pour visualiser une à une l'ensemble des séquences. Un outil informatique est chargé d'effectuer un premier tri sur les suspects potentiels. Les résultats obtenus sont alors présentés avec la possibilité d'obtenir différents niveaux de détails jusqu'à la relecture de la portion de vidéo identifiée.
Concrètement, les illustrations suivantes sont issues de la contribution de THALES pour le projet Infom@gic, s'appuyant donc sur le corpus Peachtree, en version autonome (non intégrée au démonstrateur global UrbanView). Imaginons qu'un vol ait eu lieu, et que certains témoignages orientent les recherches vers une voiture verte suspecte, passée sur le parking de la station de lavage entre 10h et 10h45 selon les témoins. Une fois les vidéos obtenues auprès des différentes sources de surveillance, elles sont chargées sur ordinateur. La généricité des outils permet de prendre en compte différents standards qui peuvent être déployés (dans notre cas, toujours MPEG-2, MPEG-4 Part 2 et 10).

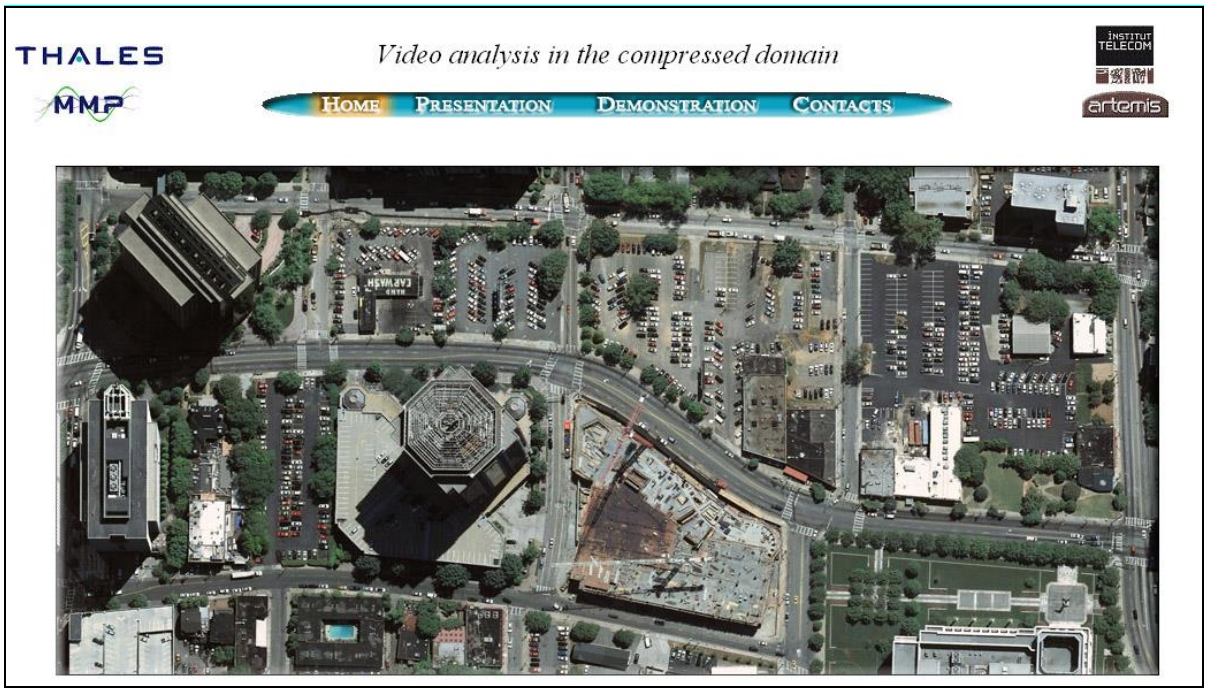


Figure 74 – Accueil de l'interface : localisation du corpus

L'interface proposée à l'utilisateur, Figure 74, s'appuie sur du PHP, ce qui autorise l'accès depuis n'importe quel navigateur html. L'utilisateur dispose de différents champs de requêtes, ainsi qu'illustré Figure 75. Il peut choisir la vue couverte selon la ou les caméra(s) disponible(s), puis la zone à l'intérieur du champ de vision. Les heures de début et de fin déterminant l'intervalle de temps à couvrir sont ensuite à définir. Enfin, il peut choisir la couleur de l'objet à rechercher, avant de lancer l'analyse.

Figure 75 – Formulaire de requête

Le programme d'analyse est alors exécuté. Dans un premier temps, la segmentation est celle présentée dans le chapitre précédent. Pour répondre à la problématique de l'investigation, des outils logiciels additionnels ont dû être développés.

3.5.2 Détermination des résultats répondant à la requête utilisateur

Pour déterminer les objets qui correspondent à la recherche en cours, il est dans un premier temps nécessaire de segmenter tous les objets mobiles sur la séquence. La seconde étape consiste à effectuer un suivi temporel de ces objets pour créer des pistes. Enfin, une identification de la couleur est menée.

3.5.2.1 Extraction des pistes cibles

Les différentes images binaires obtenues servent dans un premier temps à extraire les boîtes englobantes (*bounding boxes*) de chaque objet (Figure 76). Une méthode de suivi a été développée pour permettre l'agrégation temporelle tout en conservant un nombre d'images analysées par seconde supérieur à 350.

Figure 76 – Boîtes englobantes obtenues après segmentation

Dans un premier temps, des pistes sont créées. Concrètement, une piste est une succession de descripteurs liés à un objet mobile. Ces descripteurs comprennent :

- le numéro de l'image où l'objet est détecté,
- les coordonnées de la boîte englobant l'objet dans l'image,
- les coordonnées du centroïde du blob segmenté,
- un vecteur de déplacement instantané (en pixels/image),
- le lien vers le fichier correspondant à la vignette extraite de la séquence,
- la couleur dominante de l'objet.

Pour chaque boîte identifiée, le meilleur candidat à l'appariement est recherché dans l'image précédente. Cela est réalisé en s'appuyant à nouveau sur une hypothèse de mouvement constant :

- pour chaque objet, le vecteur de déplacement instantané est attribué, correspondant à la moyenne des vecteurs d'estimation de mouvement (normalisés et filtrés) sur le blob,
- les coordonnées (points extrêmes et centroïde) sont alors translatées de ce vecteur depuis l'image précédente vers l'image courante,
- la boîte englobante la plus proche de cette projection, selon un critère de distance comparant coordonnées, taille, largeur, hauteur, et le ratio largeur sur hauteur, est appariée avec la piste en cours,
- si cette distance est au-delà d'un seuil, paramètre du système, l'appariement n'est pas réalisé.

Figure 77 – Exemple de pistes détectées

En cas d'absence de candidats éligibles, une vérification est faite sur les pistes non fermées susceptibles de correspondre à l'objet en cours. En effet, nous avons vu lors de la présentation des différents résultats que, si d'une manière générale, la segmentation est globalement homogène au cours du temps, il arrive que des non détections aient lieu sur une ou plusieurs images consécutives. Dans ce cas, à partir du dernier vecteur déplacement et de l'écart entre l'image courante et la dernière de la piste, la position de l'objet est projetée. Cet artifice très peu coûteux en temps de calcul permet de lisser le suivi et d'éviter la création intempestive de nouvelle piste à chaque discontinuité dans la segmentation. Si le nombre d'images entre l'image courante et la dernière de la piste est trop important, ou si la distance parcourue est trop grande (paramètres du système, typiquement 25 images et 150 pixels, selon l'angle de vision de la caméra et la proximité des objets), la piste est fermée, puisque trop d'incertitude existe.

Si une fois de plus aucune piste ne répond aux critères d'appariement, une nouvelle est créée, correspondant à l'initialisation d'un nouvel objet. Cette hypothèse est de plus favorisée lorsqu'un objet est détecté sur les bords de l'image avec un vecteur entrant. De la même manière, pour une piste arrivant au bord de l'image avec un vecteur sortant, si aucune détection n'est faite dans les *n* images suivantes, la piste est fermée (*n* paramètre du système, par défaut 3).

Cet algorithme de suivi simplifié permet d'avoir un impact moindre sur la rapidité du système par rapport à un filtre de Kalman ou autre implantation concurrente. Il a tendance à favoriser la surreprésentation des pistes avec le jeu de paramètres par défaut. Ce choix est délibéré, d'autres valeurs, notamment sur la distance et l'écart maximum entre l'image courante et la dernière d'une piste, permettent une meilleure agrégation, par exemple en cas d'occultation par un arbre ou un autre objet, mais aboutissent parfois à de faux appariements. Dans le cadre d'une investigation, il vaut mieux présenter deux pistes correspondant au même objet que ne pas faire apparaître un résultat. En termes de temps de calcul, le nombre d'images analysées par seconde passe, pour le corpus Infom@gic, de 380 à 360. Le système reste donc bien au-delà du temps réel et autorise la prise en charge de données volumineuses.

L'extraction des pistes permet de lister toutes les positions consécutives d'un objet avec l'évolution de sa taille au cours du temps, et les vecteurs de déplacements instantanés. Un premier filtrage est opéré par rapport à la requête de l'utilisateur, à la suite duquel ne seront conservés que les objets étant passés au bon endroit et au bon horaire, comme illustré Figure 73. A défaut de pouvoir faire directement le lien entre l'heure précisée et le numéro des images, il est nécessaire de donner l'heure de début de la séquence, et son nombre d'images par seconde. Le logiciel parcourt ensuite la liste des pistes, vérifiant à chaque fois le couple position / numéro d'image par rapport à la requête.

En ce qui concerne le choix de la zone à l'intérieur de laquelle l'objet doit être trouvé, l'application d'analyse prend en paramètre les coordonnées de la zone, ce qui impose une surface rectangulaire. Dans l'absolu, un masque pourrait également servir de requête, autorisant des formes quelconques, mais cette implantation n'a pas été nécessaire dans le cadre des projets supports. La Figure 75 présente un exemple des six différentes zones identifiées sur l'une des caméras du corpus *Peachtree*. Dans les faits, c'est l'interface graphique qui donne en pixels les coordonnées de la zone choisie. Ainsi, un simple outil de traçage de rectangles sur une image extraite du corpus permet de faire le lien entre les zones d'intérêts et le paramétrage de l'analyse.

3.5.2.2 Classification de la couleur

Dans l'absolu, l'identification de la couleur d'un objet pourrait se faire à partir des coefficients DC des blocs chromas liés à la zone segmentée. Toutefois, un bloc chroma représente une zone de 16x16 pixels en 4:2:0 en MPEG-2 et MPEG-4 Part 2 (format couleur le plus répandu car présentant le gain le plus important en taille de flux). Or, les objets sont segmentés à l'échelle du bloc 8x8, et n'en couvrent pas toujours la totalité (route autour d'un véhicule, mur derrière un piéton, etc.). Nous avons donc privilégié un niveau intermédiaire de décompression. Parmi la liste des images successives comportant un objet, les images Intra du flux sont identifiées. Une transformée inverse est alors appliquée aux blocs de ou des image(s) intra comportant l'objet. La décompression est ainsi réduite au minimum. Dans l'exemple illustré Figure 76, cela implique qu'une seule image sur 25 sera décompressée, sur environ 5% de sa surface, soit l'équivalent d'un bloc sur 500.
Pour chaque pixel, une couleur va alors être attribuée selon les différentes couleurs requêtes possibles. Ces couleurs ont été identifiées comme représentatives des différents témoignages reconnus par les forces de police françaises. L'espace colorimétrique des vidéos compressées est Y,Cb,Cr. Les premières couleurs identifiées sont le noir et blanc, via un seuillage sur la composante de luminosité Y (les plus hautes valeurs sont classées comme du blanc, les plus faibles comme du noir). Ensuite, chacune des couleurs requêtes ayant été projetées dans l'espace (Cb, Cr), chaque pixel est apparié avec sa couleur la plus proche. Un histogramme permet de compter le nombre d'appariement pour chaque couleur, et la plus représentative est conservée comme couleur principale de l'objet. Il serait ici également possible d'extraire une couleur secondaire, ce qui peut s'avérer pertinent pour le suivi de personnes (pantalon et veste de couleurs différentes) ou de véhicule (camion avec tracteur et remorque de deux couleurs), mais n'a toutefois pas été mis en place faute d'exemple pertinent dans le corpus Infom@gic.
Cette approche de classification couleur n'est pas très précise, dans le sens où ce n'est pas la couleur réelle de l'objet qui est déterminée mais celle parmi les requêtes qui en est la plus proche. Toutefois, elle est particulièrement rapide en exécution et aboutit aux résultats attendus sur les corpus de vidéosurveillance.
Un second filtrage est alors effectué (bas de la Figure 73), renvoyant alors les objets présents au bon endroit au bon moment, et de la bonne couleur, donc répondant à l'ensemble des paramètres de la requête.

3.5.3 Interface de résultats

Le logiciel d'analyse génère pour chaque piste d'une part une liste de vignettes répondant à la requête, et d'autre part un fichier php qui sera interprété par l'interface graphique.
Les résultats sont alors affichés dans un tableau, comme présenté Figure 78. Apparaissent alors pour chaque piste :

- Une vignette qui montre l'objet segmenté,
- Le numéro de la piste correspondant,
- La couleur de l'objet déterminée,
- Le nom du fichier dans lequel la piste a été identifiée,
- Deux boutons permettant d'obtenir des détails supplémentaires.

Le premier bouton (*Show Thumbnails*) permet d'afficher plusieurs vignettes de la même piste (obtenues à partir des portions d'images Intra décompressées pour la classification couleur). Cette option est présentée Figure 79. L'autre bouton (*Play video*) permet de lancer la vidéo correspondante directement sur la portion dans laquelle l'objet est présent (*via* les numéros d'images de début et fin de piste).

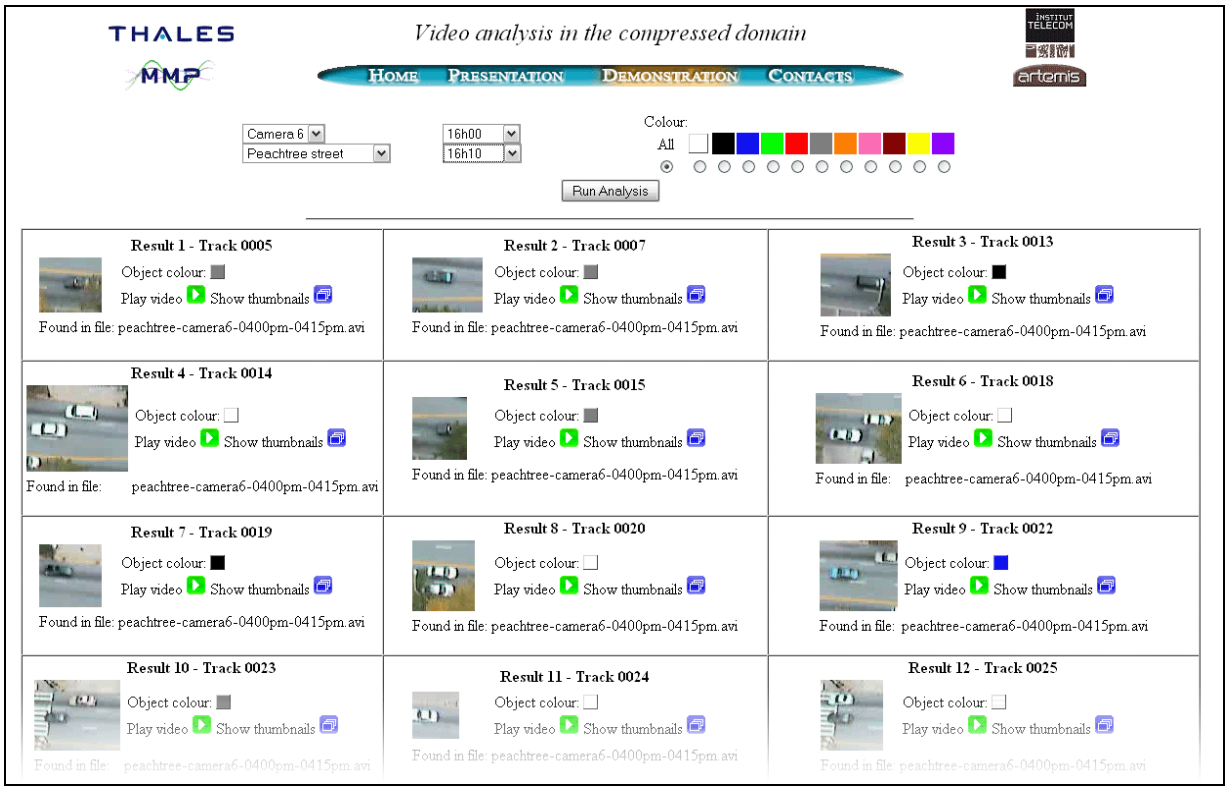


Figure 78 – Tableau de résultat

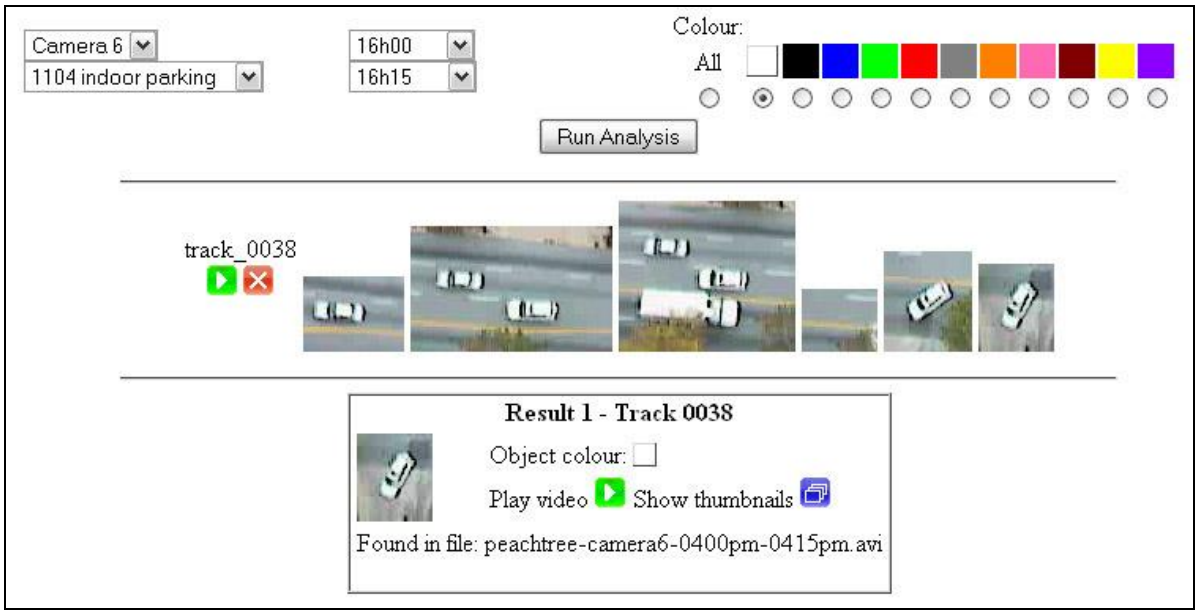


Figure 79 – Vignettes liées à une piste

Comme évoqué précédemment, le jeu de paramètres utilisé pour l'établissement des différentes pistes privilégie un objet par piste plutôt qu'une piste par objet. Cela se traduit par la présence éventuelle de plusieurs pistes pour un même objet, mais réduit le problème que représenteraient plusieurs objets sur une même piste. L'interface utilisateur permet d'identifier rapidement des pistes appartenant au même objet, puisqu'une rapide visualisation de la ou des vignettes par piste permet à l'opérateur de faire le lien entre les différents résultats.

3.6 Discussion

Globalement, cette application concrète de l'analyse dans le domaine compressé répond aux attentes initiales : une heure de vidéo est analysée en 2 minutes, et si certaines pistes sont doublées en raison des limites du suivi utilisé, l'interface de visualisation des résultats permet de ne pas trop pénaliser l'utilisation de l'outil. Parmi les points à améliorer, le principal reste la précision de la segmentation. Dans le cadre du corpus *Peachtree*, la méthode d'analyse dans le domaine compressé touche à ses limites : lorsque deux véhicules se déplacent dans la même direction, à la même vitesse, en étant proche l'un de l'autre (fait courant sur une artère principale, notamment aux heures de pointe), ils sont presque systématiquement segmentés comme un objet unique. La piste générée est alors faussée, et la classification couleur biaisée par le plus gros véhicule du groupe. D'une manière générale, pour que l'application fonctionne correctement à haut niveau (interface utilisateur), il faut qu'à bas niveau (flux compressé), au moins deux blocs séparent deux objets distincts. Selon les réseaux de surveillance considérés, leur implantation et la densité des flux d'objets, les résultats peuvent varier très fortement. Ces limites ne sont pas propres au domaine compressé, considérant par exemple le problème encore non-résolu même au niveau pixellique de suivi robuste de personnes dans une foule, mais la faible résolution de travail sur les vidéos compressées diminuent les performances des systèmes.
Selon ces considérations, l'avenir de l'analyse dans le domaine compressé serait plutôt lié

- Soit à des plates-formes spécifiques basse ou très-basse consommation, type caméra intelligente ou capteur déposé. Il s'agit de l'un des aspects critiques pour THALES, qui pourrait ainsi proposer des outils de surveillance de zone à forte autonomie et à bas coût.
- Soit à des tâches de prétraitement. En version autonome, les résultats et le démonstrateur proposés montrent les limites. Les objets trop petits, ou trop proches les uns des autres, ne sont pas précisément segmentés de façon homogène et continue sur une séquence. Toutefois, avec des seuils plus faibles, il est possible, au prix de l'augmentation des fausses alarmes, de diminuer drastiquement les non-détections (moins de 1%). Une fois ces zones d'intérêt (*ROI*) identifiées, il devient possible d'appliquer différents processus ciblant alors ces zones. L'analyse dans le domaine compressé n'a alors plus comme objectif de remplacer des outils au niveau pixellique, mais de préparer le travail pour accélérer les temps de traitements à suivre.

Ces différents outils, qui s'appuient sur l'analyse dans le domaine compressé, font l'objet de le chapitre suivant.

Chapitre 4

Enrichissement du flux : vers de nouveaux services et applications

Résumé du chapitre

L'analyse dans le domaine compressé permet d'identifier, rapidement et pour un coût en temps de calcul et mémoire réduit, les zones d'intérêt d'une image dans une séquence vidéo. Nous avons considéré cette segmentation non pas comme une finalité mais comme une première étape vers différents services et applications qui permettent d'enrichir le flux vidéo. De ces réflexions sont nés des outils qui permettent d'accélérer les traitements vidéo au niveau du pixel, le cryptage visuel partiel d'une vidéo, la protection aux erreurs ou encore l'adaptation de débit ciblé. L'ensemble de ces algorithmes peut être utilisé de manière isolée ou combinée, offrant de nombreuses perspectives et améliorations pour les réseaux de vidéosurveillance.

Sommaire du chapitre

4.1 Introduction

Pour de nombreux domaines exploitant la vidéo, incluant la vidéosurveillance ou le langage des signes, la partie majeure de l'information visuelle peut être réduite à de petites régions au sein de l'image. On pourra ainsi se focaliser sur les véhicules ou les piétons dans un cas, ou sur les mains et le visage d'une personne qui signe. Les outils que nous avons présentés dans les chapitres précédents permettent d'identifier ces zones d'intérêt. Toutefois, les différents travaux de l'état de l'art se contentent aujourd'hui d'extraire ces zones mobiles, sans s'intéresser à l'usage qui peut en être fait *a posteriori*. Nous avons ainsi proposé dans un premier temps un outil d'aide aux investigations en fin du chapitre précédent, mais nous avons également cherché à exploiter au maximum nos outils d'identification de *RoI* bas-coût (en mémoire et temps de calcul).
Ce chapitre s'attachera donc à décrire les différentes applications qui découlent des présents travaux, depuis les considérations autour de la transmission du flux et de la protection aux erreurs, jusqu'à l'utilisation de l'analyse dans le domaine compressé comme étape de prétraitement, en passant par les aspects de cryptographies visuelles pour le respect de la vie privée, de tatouage ou stéganographie ciblée, l'adaptation automatique de débit, ou encore les applications conjointes permettant d'envisager des systèmes de vidéosurveillance au paramétrage automatique et dynamique.
L'ensemble de ces outils s'appuyant sur l'analyse dans le domaine compressé ont fait l'objet d'une famille de brevets déposés en 2008 [Leny et al., 2008c,d,e,f ; Le Barz et al., 2008a,b,c,d,e], fondés avant tout sur une étude théorique quant aux débouchés des travaux présentés précédemment, les différentes applications visées sont aujourd'hui à des niveaux différents de développement, exploitant ces inventions selon les besoins de Thales pour les futurs développements de démonstrateurs ou produits. A titre d'illustration le brevet *FR 08 .06837,* "Procédé et dispositif pour l'enfouissement d'une séquence binaire dans un flux vidéo compressé " est proposé en annexe de ce document.

4.2 Prétraitement avant analyse affinée

L'objectif initial de notre approche était de proposer une méthode de segmentation rapide et basse consommation. Le démonstrateur pouvant être utilisé à des fins d'investigation présenté au chapitre précédent a permis de prouver l'utilité de nos travaux tout en se heurtant aux limitations du procédé. La principale tient à la précision de la segmentation, à l'échelle du bloc, qui ne permet pas d'obtenir des contours précis.
Pour compenser ce problème, nous avons proposé une approche mixte, détaillée Figure 80 : dans un premier temps, les outils d'analyse dans le domaine compressé permettent d'identifier les régions qui contiennent des objets mobiles. Une fois ces régions d'intérêt isolées, il est alors possible de décompresser le flux en les ciblant, selon la structure du flux. Ainsi sur une image Intra, il est uniquement nécessaire d'effectuer les transformées inverses sur les blocs comprenant les objets mobiles.
Afin de s'assurer que l'analyse sur les informations décompressées puisse fournir les meilleurs résultats possibles, il est nécessaire que l'ensemble des objets d'intérêt soit présents dans les premiers résultats de segmentation. Ainsi, il est préférable pour ce type d'utilisation de s'assurer que les objets soient effectivement segmentés, quitte à augmenter le taux de fausses alarmes. Celles-ci seront par la suite compensées par la seconde analyse qui permet donc d'éliminer les pixels du fond et de fournir une segmentation au niveau pixellique des objets mobiles.

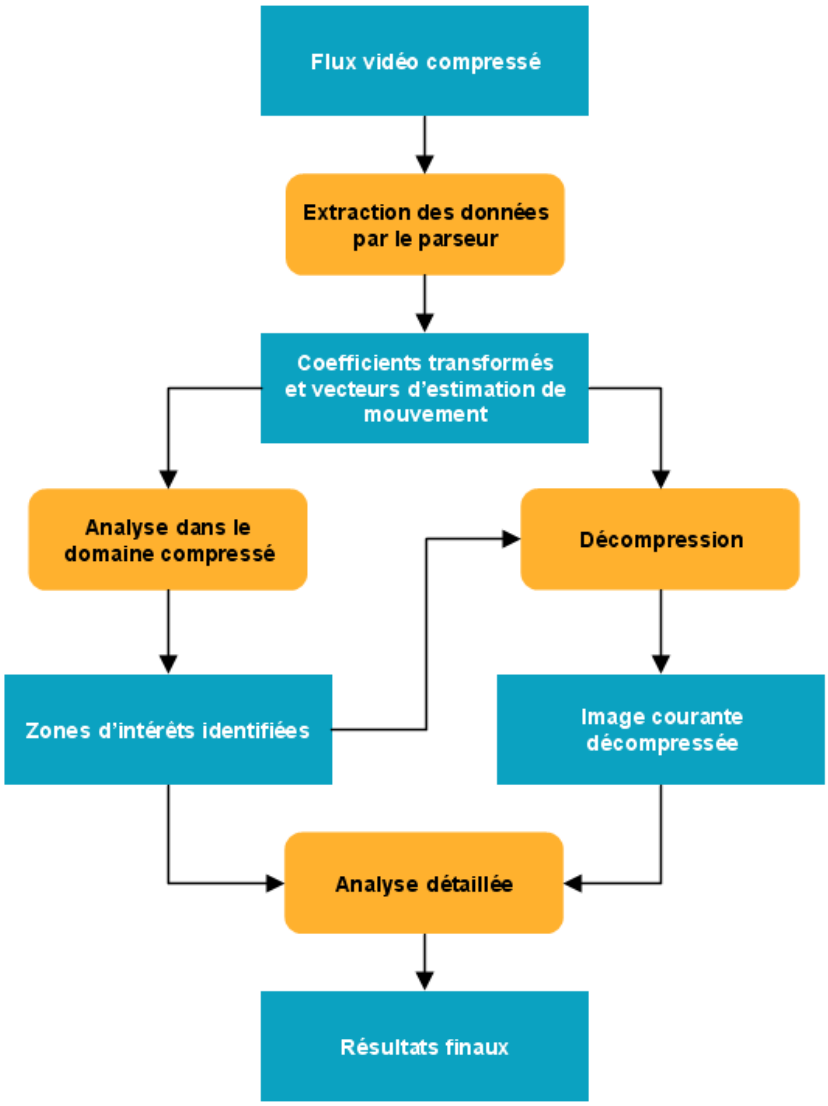


Figure 80 – Prétraitement dans le domaine compressé.

Aujourd'hui, cette approche est en cours d'implantation pour des modules de compression et d'analyse embarqués. D'un côté, une plateforme est en charge de la compression et de l'analyse conjointe (dans le domaine compressé donc). En sortie de celle-ci, un flux est transmis par réseau, comprenant à la fois le flux vidéo compressé (H.264 AVC) et les métadonnées comprenant les coordonnées des boîtes englobantes extraites par analyse dans le domaine compressé. A l'autre bout de la chaîne, un second boîtier est en charge de la décompression du flux vidéo, puis d'une analyse plus fine à l'intérieur des régions d'intérêt ayant pour objectif l'identification et la classification de piétons. Ces travaux menés conjointement entre plusieurs entités de Thales doivent faire l'objet d'une démonstration début 2011. Un premier module embarqué basse consommation fournira la compression en H.264 et l'analyse conjointe d'un flux vidéo analogique. La séquence d'entrée pourra aussi bien être dans le visible que dans l'infrarouge, couvrant des besoins opérationnels tels que la surveillance de frontière. En sortie, le flux H.264 comprendra les résultats de l'analyse dans le domaine compressé sous la forme de coordonnées de boîtes englobantes identifiant les zones mobiles de l'image qui pourront être transportées dans des NAL 30 ou 31. En bout de chaîne, un second module sera chargé de la récupération de ces métadonnées et de la décompression du flux. Il lancera alors une analyse au niveau pixel ciblant les zones précédemment identifiées, pour classer les objets et identifier les piétons.

4.3 Etude intermédiaire sur la structure des flux

Ainsi que nous l'avons précisé, diverses applications présentent un intérêt plus ou moins direct à pouvoir pointer les zones d'intérêt à l'intérieur d'un flux vidéo. Toutefois, cet accès immédiat est aujourd'hui une tâche difficile, sinon impossible, considérant les standards actuels. L'un des objectifs de MPEG-4 est d'offrir un 'accès multimédia universel' ou *Universal Media Access* [Wang et al., 2007], ce qui dans le cas qui nous intéresse signifie être en mesure de coder, décoder ou cibler par un autre outil un objet spécifiquement ciblé au sein d'une séquence vidéo. Si un tel résultat peut être obtenu *via* l'utilisation de couches de transparence comprenant chacune un objet, un unique flux vidéo ne peut encapsuler tous ces éléments.
En rendant possible le codage des objets dans une vidéo de manière indépendante, de nouvelles applications peuvent être entrevues. Dans la suite de ce chapitre, nous nous attacherons à décrire ces débouchés liés à l'analyse dans le domaine compressé. Toutefois, il est nécessaire dans un premier temps de présenter les limitations auxquelles nous nous sommes heurtés quant à l'utilisation de H.264 AVC pour une structuration sémantique du flux, ainsi qu'une proposition de structure permettant d'envisager l'implantation de modules de traitement du flux vidéo à tout endroit sur la chaîne de transmission.

4.3.1 Le découpage du flux orienté vers la sémantique objet

Nous détaillons ici les limitations rencontrées quant à l'utilisation des standards de compression actuels dans le cadre de la vidéosurveillance. Ces observations nous ont amenés à définir une nouvelle approche de la hiérarchie de l'image, s'appuyant sur des objets sémantiques visuels. Cette structure nous permet enfin de proposer de nouveaux outils, qui offrent diverses améliorations à la navigation dans un flux.

4.3.1.1 Les limites du partitionnement du flux

Les différents standards jusqu'à H.264 fournissent un cadre permettant à la fois le codage d'objets indépendants sous forme de calques et de partitionnement de l'image sous forme de s*lices* et *slice groups*. Pour autant, ces deux approches restent différentes et n'ont pas été initialement définies pour être compatibles entre elles.
Dans un premier temps, les objets indépendants tels que définis par la composition de scène (MPEG-4) sont concrètement des parties différentes du flux, comprenant des vidéos, textes, avatars, etc. Mais si les objets sont codés indépendamment les uns des autres, visualiser deux objets revient à décoder deux séquences. Il n'y a pas un flux compressé unique contenant les deux objets.
Par ailleurs, lorsque l'on utilise les *slices* et *slice groups* à l'intérieur d'un simple flux vidéo tel qu'une séquence H.264, le changement dynamique du partitionnement se révèle épineux. En considérant que l'on puisse définir pour chaque image la forme de chaque objet, H.264 ne définit pas de procédure adaptée pour les intégrer. Les décodeurs actuels s'attendront ainsi à une image codée Intra après la définition de paramètres images (*Picture Parameters Set* – ou *PPS*) modifiant la carte d'allocation des macroblocs (*MBAmap* – ou *Mabroblock Allocation Map*). Les objets peuvent être chacun défini à l'intérieur d'un *Slice Group*, grâce à une *MBAmap*. Toutefois le profil *Main* du standard n'autorise qu'un groupe, ce qui rend cette option irréaliste. Les profils *Baseline* et *Extended* peuvent en contenir jusqu'à 8. Toutefois en

vidéosurveillance, avec un groupe alloué au fond, seuls 7 objets peuvent être codés indépendamment dans une image, ce qui devient vite problématique selon l'activité du site surveillé, d'autant plus avec l'augmentation de résolution des caméras qui laissent envisager des applications complexes, comme dans les halls d'aéroports.
Au sein d'un *slice group*, un *slice* peut être défini uniquement comme une sous-portion du flux de macroblocs, ce qui implique qu'il n'est pas possible d'assigner un *slice* à une sous-partie d'objet. Cette particularité peut être utile, par exemple pour reconstruire une portion d'arrière plan derrière un véhicule qui produit une occultation partielle.

4.3.1.2 La structure du flux proposée

Avant de décrire l'approche pour découper un flux vidéo, définissons tout d'abord les termes suivants :

- Un **objet** est un ensemble connexe de pixels, blocs ou macroblocs, qui possède son propre mouvement comparé aux autres objets au sein d'une séquence vidéo. Pour la vidéosurveillance, il peut s'agir de véhicules, piétons, animaux, etc.
- Un **groupe d'objets** est alors défini par un descripteur commun choisi par l'utilisateur lors du codage du flux vidéo. Ce descripteur peut être bas voire très bas niveau (couleur, taille, direction de déplacement, etc.), ou de niveau moyen/haut (classe de l'objet - personne/véhicule/fond, localisation spatial dans la séquence, etc.). Un groupe d'objets peut ainsi comprendre les objets rouges, ceux qui se déplacent vers la gauche, ou l'ensemble des piétons à l'image.
- Un ***slice*** est défini comme une sous-portion de l'image, contrairement à une sous-portion du flux dans les standards actuels : jusque H.264, un *slice* est identifié au sein du flux par le mot VLC "first_mb_in_slice". Cette définition sous-entend qu'à partir d'un macrobloc, un *slice* occupera les lignes entières qui suivent dans l'image, à l'intérieur du *slice group* en cours, jusqu'au démarrage du prochain *slice*. Il n'est pas possible de définir n'importe quelle sous-partition d'un *slice group* comme *slice*. Pour l'approche suggérée, un *slice* peut être défini *via* la carte d'allocation des macroblocs MBAmap ou *via* les macroblocs extrêmes comme cela est fait pour les boîtes englobantes (Figure 81). Le terme *slice* est utilisé en référence / continuité à la définition MPEG, mais pour une meilleure compréhension, il faut ici le considérer comme une portion d'objet.

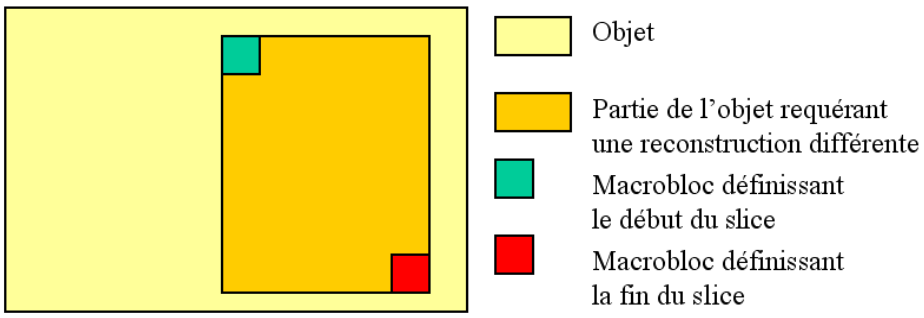


Figure 81 – Définition d'un *slice*.

Cette nouvelle définition des *slices* autorise une reconstruction partielle après une occultation par exemple. Comme illustré Figure 82, deux objets sont définis : la voiture, et le fond. La voiture est reconstruite par estimation de mouvement par un *slice*. Le fond, en revanche, est constitué de deux *slices* : la majeure partie est

directement recopiée de l'image précédente, mais la partie immédiatement à l'arrière de la voiture doit être reconstruite puisqu'elle n'était pas visible dans l'image précédente. Un *slice* est donc instancié, correspondant à cette zone, ce qui permet une identification dans le flux et une reconstruction des deux sous-parties du fond faciles et rapides.

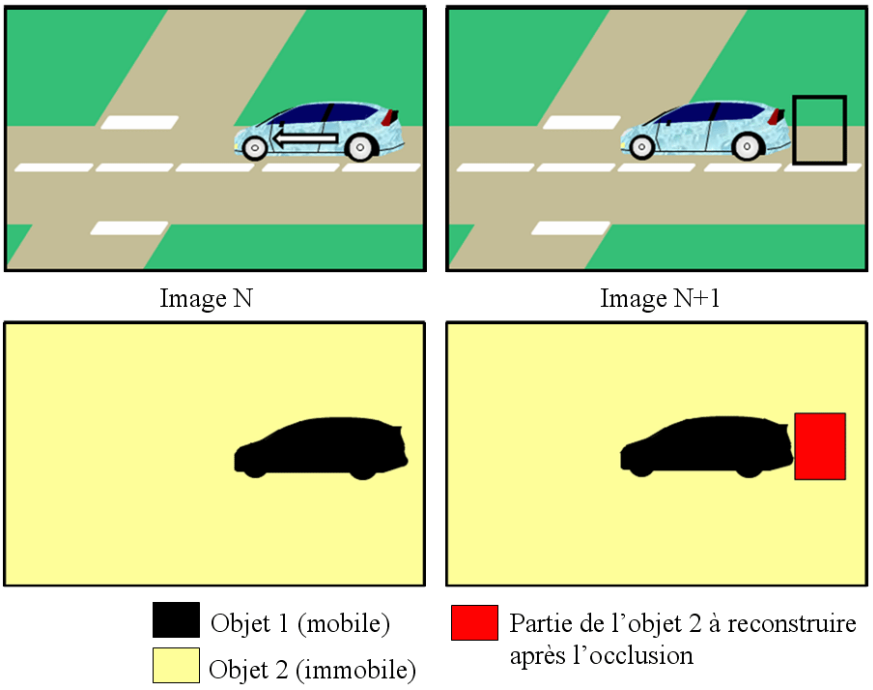


Figure 82 – Reconstruction après une occultation.

En prenant en compte ces définitions, nous pouvons proposer une nouvelle hiérachie au sein de l'image (Figure 83). Une image comporte un ou plusieurs groupes d'objets, sans limites de nombre pour prévoir les évolutions des caméras de surveillance vers la très haute définition, qui pourront ainsi filmer plusieurs dizaines d'objets en simultané. Par « sans limites de nombres », nous voulons préciser que si plusieurs profiles sont définis, même le *baseline* doit être en mesure de prendre en charge plus d'une centaine d'objets, et les profils plus hauts peuvent avoir à définir plusieurs milliers d'objets (on y tend déjà avec les caméras très hautes définitions dans les halls de gare). Chaque groupe d'objets peut contenir un ou plusieurs objets présentant un attribut commun choisi par l'utilisateur lors du codage du flux. Enfin, chaque objet peut contenir un ou plusieurs *slices*, qui sont définis par le codeur pour optimiser la reconstruction.

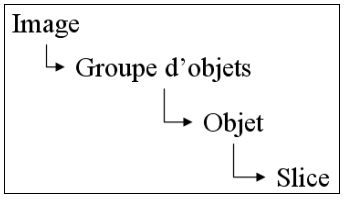


Figure 83 – Hiérarchie image.

Contrairement à l'une des limitations évoquées concernant les standards actuels, cette partition produit un flux unique, au sein duquel la séquence est structurée selon les objets sémantiques. La Figure 84 présente le flux correspondant à la scène de la séquence Figure 86,

ainsi que l'arbre hiérarchique Figure 85. Chaque élément de la hiérarchie possède son propre entête, qui contient les descripteurs utilisés lors du codage de la vidéo pour déterminer les *slices*, objets ou groupes d'objets, ce qui fournit également un outil utile pour une navigation rapide au sein du flux. Les descripteurs peuvent s'appuyer préférablement sur des normes telles que les descripteurs visuels MPEG-7 ou STANAG 4609 mais autorise aussi des définitions propriétaires ou toute autre solution pour répondre éventuellement à des applications spécifiques. La segmentation suggérée est liée au mouvement, puisque c'est ainsi que les objets ont été définis. A partir de celle-ci, plusieurs descripteurs sont extraits pour chaque objet, et seront ensuite utilisés pour le processus de classification (supervisé on non) qui détermine les groupes d'objets. L'ensemble du processus de déclaration d'objets aboutit à une indexation par le mouvement.

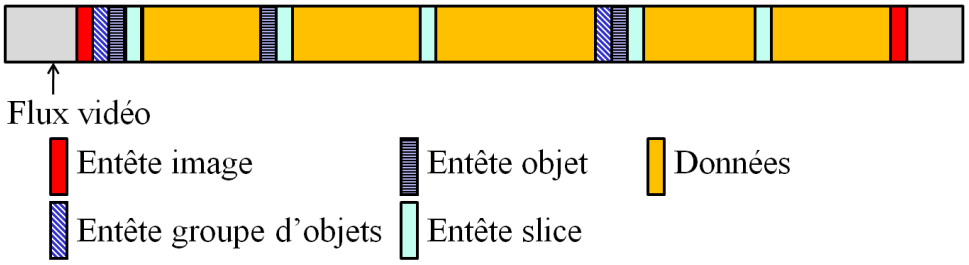


Figure 84 – Structure du flux vidéo.

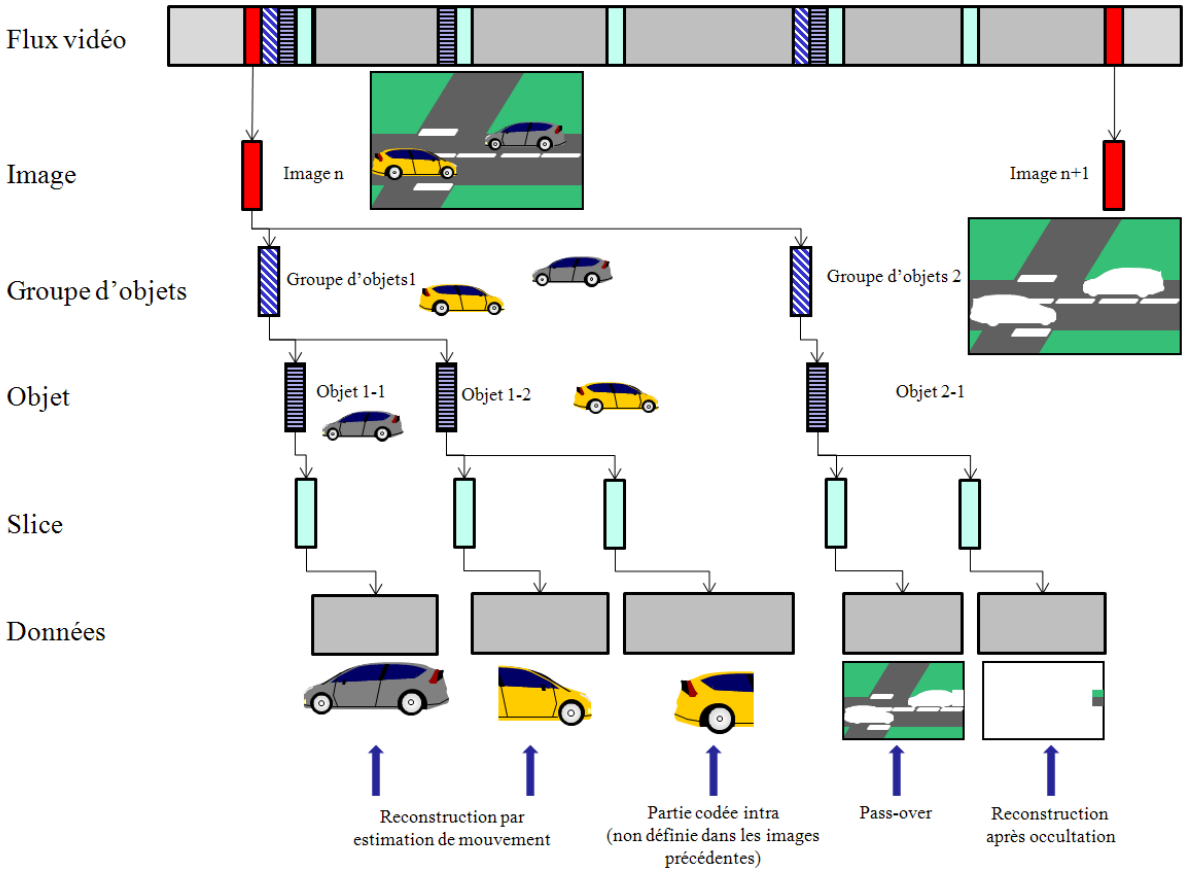


Figure 85 – Représentation en arbre de la structure.

Deux groupes d'objets, correspondant aux objets mobiles et au fond, sont représentés Figure 84 et Figure 85.Le premier comporte deux objets (deux voitures). Le premier objet est défini

par un *slice* unique, alors que le second en comporte deux (l'avant et l'arrière de la voiture). Le second groupe d'objet comporte un objet, composés de deux *slices* (le fond déjà présent dans l'image précédente et la portion qui requiert une reconstruction spécifique après une occultation).

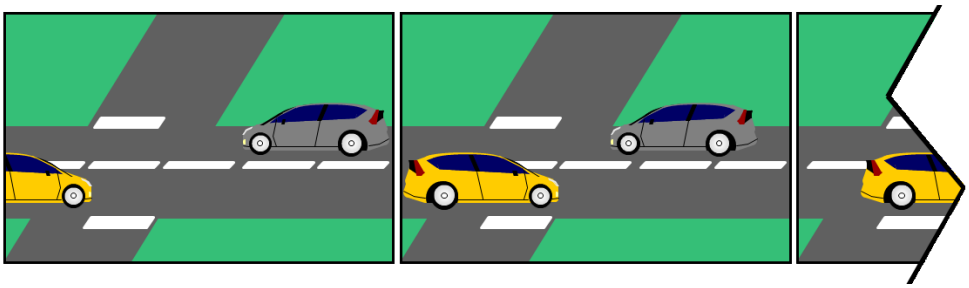


Figure 86 – Séquence d'illustration (la voiture jaune rentre dans le champ depuis la gauche, alors que la voiture grise se déplace de la droite vers la gauche).

L'arbre en Figure 85 détaille la structure de l'image du milieu de la séquence d'illustration. L'objet 1-2 comporte deux *slices* puisque l'avant de la voiture peut être reconstruit par estimation du mouvement, alors que l'arrière (non présent dans l'image précédente), sera codé Intra. Le même principe s'applique à l'objet 2-1, pour lequel la partie précédemment occultée du fond sera reconstruite différemment du reste.

4.3.1.3 Codage et compression

Pour permettre une reconstruction totalement indépendante des objets, ces derniers doivent être reconstruits sans pointer les uns sur les autres (au sens de l'estimation de mouvement). C'est pourquoi la compensation de mouvement doit pouvoir être restreinte à l'intérieur d'un objet. Si les objets sont détourés de manière suffisamment précise, l'impact sur la compression d'une telle contrainte est minimal : il n'est pas nécessaire de pointer en dehors de l'objet pour obtenir les blocs de référence. Cet aspect requiert la prédiction de la position de chaque objet avant que l'image soit compressée. Cela peut être réalisé grâce à un outil de suivi couplé à un filtrage de Kalman pour prédire la position attendue des objets détectés dans les images précédentes. Une segmentation plus fine, par une soustraction de fond par exemple, pourra améliorer les contours et identifier l'arriver de nouveaux objets dans l'image. La déclaration et la position de l'objet sont identifiées dans le flux par une carte d'allocation de macroblocs (MBAmap), similaire à celle présente dans H.264, qui sera précisée dans l'entête image. La carte en elle-même peut être compressée par estimation de mouvement, pour un impact minimal sur le ratio de compression. Le codeur/décodeur doit surtout prendre en compte les changements dynamiques de la carte MBAmap.
En utilisant cette définition des *slices*, nous proposons également la notion de *pass-over*, ou référencement direct. Avec H.264, un macrobloc *skipped* est reconstruit en considérant la prédiction de mouvement des blocs voisins. Pour recopier exactement les mêmes blocs que ceux présents dans une image de référence, un vecteur d'estimation de mouvement égal à zéro doit être codé. Le *pass-over* correspond à ce vecteur nul, offrant une syntaxe plus courte combinée à la possibilité de références multiples, que nous définissons comme *multi-reference pass-over*. Dans de nombreux domaines tels que la vidéosurveillance, les objets se déplacent devant un fond immobile. Une région qui était occultée peut maintenant être reconstruite à partir de n'importe quelle image précédente contenant la zone, comme illustré Figure 87. Lorsque plusieurs objets se déplacent en simultanément, les références multiples permettent au *pass-over* de pointer vers une image différente pour chaque *slice* occulté. Les

slices tels que définis précédemment optimise également l'approche, puisque une seule référence de *pass-over* peut être associée au *slice*. Il n'y a ainsi pas besoin de déclaration macrobloc par macrobloc comme pour H.264.

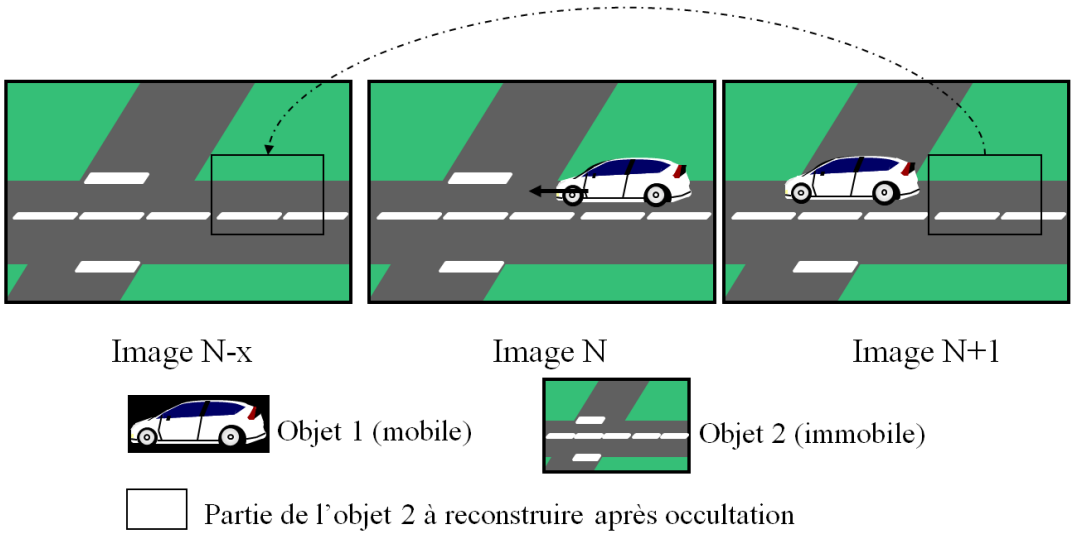


Figure 87 – *Pass-over* à références multiples.

4.3.1.4 Avantage de la structure proposée

La hiérarchie de l'image proposée autorise un décodage indépendant de chaque objet et donc son affichage sans problème de reconstruction. Les objets non décodés laissent des surfaces planes. Ces caractéristiques peuvent être optimales selon les applications visées. Ainsi seuls les véhicules sont nécessaires pour l'identification automatique de plaques d'immatriculation ; seuls le fond et les personnes présentes à l'intérieur d'une sous portion de l'image sont requis si l'utilisateur a zoomé sur une zone d'une vidéo.

La nouvelle structure a également été conçue pour offrir un nouvel accès multimédia. Il devient possible d'accéder directement à un niveau de hiérarchie donné, c'est-à-dire aux éléments d'une séquence considérés comme significatifs, en naviguant rapidement à l'intérieur des entêtes images/groupes d'objets/objets. Cette navigation peut être concrétisée *via* divers critères. Si l'utilisateur sait ce qu'il recherche (par exemple un objet bleu), il peut lancer le décodage en spécifiant un processus de décodage limité aux objets concernés. La navigation peut également être interactive : la lecture des entêtes du flux (*parsing*) extrait les descripteurs qu'ils contiennent et l'utilisateur précise ceux qu'il souhaite prendre en compte. Le décodage interactif peut également être obtenu par une sélection *point and click*, autorisant l'utilisateur à cliquer directement sur la séquence pour préciser les éléments qu'il souhaite continuer à visualiser.

La définition des descripteurs au sein des entêtes fournit aussi un outil puissant pour l'indexation, donc pour la réponse sur requêtes, à partir d'un unique flux. Les métadonnées sont codées à l'intérieur même de la vidéo compressée, et ne requièrent pas un codage et/ou une transmission dédié(e). L'implantation du processus permet d'utiliser ces données sans nécessiter d'outils supplémentaires. La structure dans sa globalité permet une navigation sémantique, s'appuyant sur la définition d'une hiérarchie ou d'une classification d'objets, au lieu des possibilités actuelles qui sont restreintes à une scalabilité spatiale ou temporelle dans une séquence.

Si une vidéo est compressée en n'employant pour chaque image qu'un groupe d'objet comprenant un objet fait d'un unique *slice*, la compression est similaire à celle obtenue *via* les standards actuels, ce qui implique des ratios de compression comparables.

4.3.2 Implantation H.264

Avec les nouvelles investigations menées aujourd'hui par le groupe d'experts MPEG, telles que le format d'application multimédia pour la surveillance (ou *SMAF* pour *Surveillance Multimedia Application Format*), un standard de compression dédié ou du moins optimisé pour la vidéosurveillance pourrait prochainement voir le jour. Dans ce cadre, nous souhaitons proposer la structure que nous venons d'exposer afin qu'elle fasse dans le futur faire l'objet d'une procédure de standardisation. Toutefois, nos besoins confrontés aux normes actuelles nous ont orientés vers un codec dédié H.264 qui s'oriente vers cette hiérarchie, sous réserve de quelques restrictions.

Remarques préliminaires
Ainsi que cela a été précisé, l'un des principaux problèmes auquel nous devons faire face est la compatibilité avec la partition dynamique de l'image. Le standard H.264 ne décrit ni ne requiert une implantation spécifique concernant ce point. Le seul point détaillé est que lors d'une mise à jour d'une carte d'allocation des macroblocs MBAmap, une description des paramètres images (*PPS unit*) doit être précisée au préalable. Habituellement les *PPS* sont employés pour les paramètres de quantification, le nombre d'images de référence, les prédictions bidirectionnelles pondérées, etc. Cela pris en compte, la majorité des décodeurs attend une image codée Intra après un *PPS*, même si le standard ne le précise pas. Notre utilisation des *PPS* et de l'allocation dynamique des macroblocs étant inhabituelle voire inattendue, il est probable que la plupart des décodeurs ne soient pas en mesure de décoder ces séquences spécifiques.
Nous restreignons la structure image détaillée dans cette section pour l'adapter aux possibilités offertes par H.264, comme le précisent les équivalences du Tableau 25. Le niveau *image* reste inchangé, et une nouvelle image est identifiée par son index *framenum*. Le niveau *groupe d'objets* n'est pas utilisé. Les *slices groups* du standard sont utilisés pour définir le niveau *objet*, au sein duquel les *slices* proposés sont remplacés par les *slices* définis par H.264.

Tableau 25 – Equivalence entre la hiérarchie proposée et la structure d'un flux H.264

Nouvelle approche	H.264
Image	Image (*access unit*)
Groupe d'objets	Aucune
Objet	Slices group
Slice / portion d'objet	Slice

Certains des outils et avantages décrits précédemment ne sont donc pas accessibles ou du moins pas aussi performants que prévu une fois restreints à H.264. Le *pass-over* sans la nouvelle définition des *slices* ne peut être utilisé et est donc remplacé par la déclaration macrobloc par macrobloc présente dans le standard. La structure hiérarchique combinée avec les descripteurs ne peut être insérées directement dans le flux vidéo compressé. S'ils sont disponibles, ces éléments pourront être précisés dans les couches d'abstraction réseau (*NAL* –

Network Abstraction Layer), préférablement dans les unités de type 30 et 31 (non définies dans le standard et non utilisées par le protocole RTP).
Pour la manipulation des *slices groups*, les restrictions en H.264 implique également d'utiliser le profil *baseline* et *extended*. En effet, le profil principal (*main*) n'autorise qu'un seul *slice group*, ce qui rend caduque la définition du flux envisagée. Les *baseline* et *extended* peuvent contenir jusqu'à huit *slices groups*, ce qui reste handicapant considérant l'usage sémantique que nous souhaitons en faire, mais peut toutefois prendre en compte jusqu'à sept objets dans la scène simultanément (le huitième *slices group* étant attribué au fond).

Processus de codage
Dans le cadre de la vidéosurveillance, il n'y a pas a proprement parler de « début de séquence », ce qui nécessite des considérations spécifiques sur l'initialisation des outils. Toutefois, un temps d'initialisation long d'un groupe d'images (*GoP*) est suffisant pour permettre les procédures suivantes :

- La détection des objets mobiles : dans le cadre des présents travaux, nous proposons évidemment d'utiliser l'analyse dans le domaine compressé pour identifier les véhicules, piétons, etc., ce qui permet d'effectuer ce découpage sémantique sans trop pénaliser le temps de codage du flux. Toutefois, tout type d'algorithme permettant d'arriver à une segmentation des objets mobiles répond aux besoins. La résolution de la segmentation à l'échelle du bloc de l'analyse dans le domaine compressé peut paraître pénalisante pour la déclaration des objets au sein du flux, mais il faut garder en mémoire que cette segmentation sera utilisée pour définir les *slices groups*, ce qui signifie qu'une résolution à l'échelle du macrobloc est suffisante.
- La partition dynamique de l'image : connaissant la position des différents objets mobiles et leur forme à l'échelle du bloc, la prochaine étape consiste à définir les *slices groups* et la carte d'allocation des macroblocs MBAmap. Pour être aussi proches que possible de la nouvelle hiérarchie de l'image, nous associons un *slices group* à un objet. Si plus de sept objets sont détectés, il est possible d'attribuer un *slices group* à plusieurs objets. Dans ce cas, les descripteurs seront utilisés pour regrouper préférentiellement les objets les plus similaires (par exemple les voitures circulant dans la même direction). Les unités NAL additionnelles peuvent être employées pour stocker directement les descripteurs obtenus à partir de l'analyse dans le domaine compressé.
- La compression de l'image : après les étapes précédentes, elle est réalisée presque « normalement » pour chaque objet, selon le standard H.264. Il faut toutefois restreindre l'estimation de mouvement à l'intérieur de l'objet, en prenant en compte les images précédentes et l'algorithme d'appariement qui est utilisé pour prédire la position de chaque objet. Une fois de plus, cette considération n'a que peu d'impact sur le taux de compression, puisqu'un objet peu *a priori* être prédit à partir de lui-même sur une image de référence. Ces considérations permettront un éventuel décodage indépendant des différents objets, l'un des objectifs de la nouvelle hiérarchie image.

Le flux vidéo obtenu
La vidéo compressée obtenue est compatible H.264, bien que l'utilisation de changement dynamique des *slices* ait une grande probabilité de ne pas être pris en charge par n'importe quel décodeur. Toutefois les principaux objectifs sont atteints : le fond est codé indépendamment des objets. Ceux-ci peuvent également être décodés séparément les uns des autres (à condition qu'il y en ait moins de sept dans l'image courante). Si huit objets ou plus

sont présents, ils peuvent être décodés par groupe, fournissant des ensembles présentant les similarités les plus fortes obtenues lors du processus de compression.

Ainsi que nous l'avions suggéré lors de la définition de la structure hiérarchique de l'image, cette nouvelle approche peut être employée pour décoder indépendamment les différents objets dans un flux de vidéosurveillance, mais autorise également de nouvelles applications. En effet, l'augmentation du volume de données multimédia échangées soulève une fois de plus différentes problématiques telles que la protection de la vie privée ou la confidentialité des données, la gestion intelligente des débits sur les réseaux et la prise en compte d'erreurs de transmission, etc. Les paragraphes suivants détaillent certaines solutions que nous avons entrevues pour répondre à ces problématiques.

4.4 Cryptographie visuelle

Pour la vidéosurveillance, l'un des problèmes récurrents est lié aux problématiques de protection de la vie privée. Les flux peuvent être interceptés, les personnes attendant une carte d'accès à un site à l'accueil sécurité peuvent observer les moniteurs, etc. Par ailleurs le nombre croissant de caméras implique *a priori* le recrutement de nouveaux agents de sécurité. Toutefois, les habilitations ne sont pas toujours faciles à obtenir. En cryptant précisément les véhicules et piétons dans une scène, il reste possible d'interpréter une scène tout en préservant les notions de vie privé et d'identification (les visages, vêtements, plaques d'immatriculation ou véhicules ne sont plus directement visibles), ainsi qu'illustré Figure 88. Le flux d'origine n'est alors disponible qu'après la validation de la clef de cryptage appropriée.

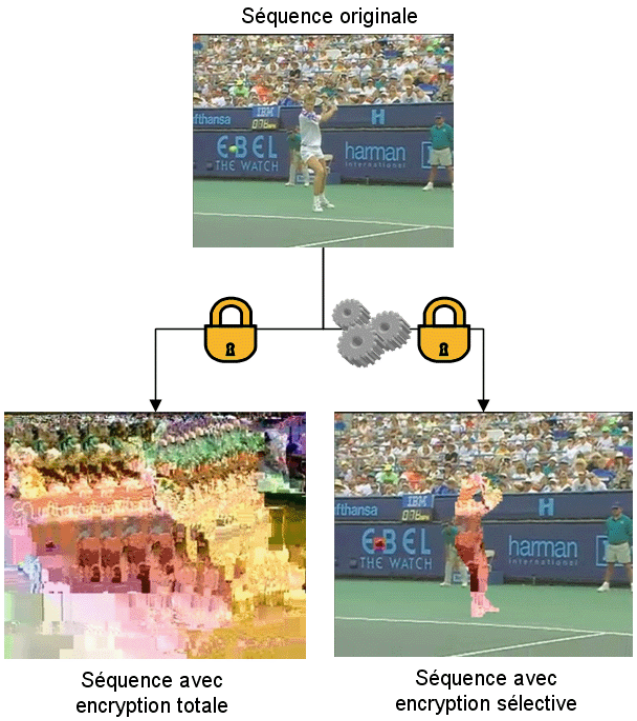


Figure 88 – Encryption sélective.

L'objectif de la méthode suivante est donc de fournir un outil capable de créer une vidéo cryptée visuellement compatible avec la norme H.264. Une étude précédemment menée à Thales [Bergeron et Lamy-Bergot, 2005] décrit une procédure s'appuyant sur AES (*Advance*

Encryption Standard [AES, 2001]) pour obtenir un tel résultat en analysant les bits des mots VLC (*Variable Length Code*) au sein d'un flux H.264 qui peuvent être altérés sans dénaturer la compatibilité avec le standard. En connaissant ces bits, une clef de cryptage peut être appliquée en les ciblant, ce qui aboutit à la séquence cryptée attendue.
Dans notre cas, l'objectif est de cibler les objets mobiles dans le flux vidéo, ce qui permet de comprendre la scène sans la clef de décryptage et de maintenir les aspects de vie privée sur les personnes et les véhicules. Sans l'implantation d'un codeur H.264 dédié, il serait possible d'identifier les objets mobiles et de les crypter sans plus de précaution. Le problème dans ce cas provient de la prédiction qui n'est pas contrainte à l'intérieur d'un objet. Les images codées Intra afficheront plus ou moins les résultats que nous recherchons, mais plus les autres images seront éloignées d'une Intra, plus le cryptage se sera étendu, jusqu'à altéré l'ensemble du champ de vision.
Toutefois, si nous considérons la structure hiérarchique décrite en 4.3.1, nous pouvons appliquer un cryptage H.264 en ciblant les *slices groups* dédiés aux objets mobiles. Ces objets pouvant être codés indépendamment dans le flux, le phénomène d'extension de la zone cryptée est résolu. Les unités NAL 30 et 31, comprenant *a priori* les descripteurs, sont également employées pour identifier les *slices groups* qui ont été cryptés, de manière que le décodeur puisse sélectionner les portions de l'image sur lesquelles appliquer l'algorithme de décryptage. L'ensemble de la chaîne de traitement permettant d'obtenir ce cryptage partiel est détaillé Figure 90.
Le schéma proposé fonctionne tout aussi bien sur des séquences ou les objets sont petits (caméra grand angle en vidéosurveillance) que pour celle où ils occupent la plus grande partie du champ de vision. Dans ce dernier cas, le résultat est proche de celui consistant à crypter tout la séquence. Toutefois, pour exploiter au mieux les possibilités offertes par le cryptage sélectif, de petits objets rendent possible l'identification contextuelle de la scène, comme le montre la Figure 89.

Figure 89 – Illustration en vidéosurveillance (en haut : Speedway, en bas : corpus Caretaker).

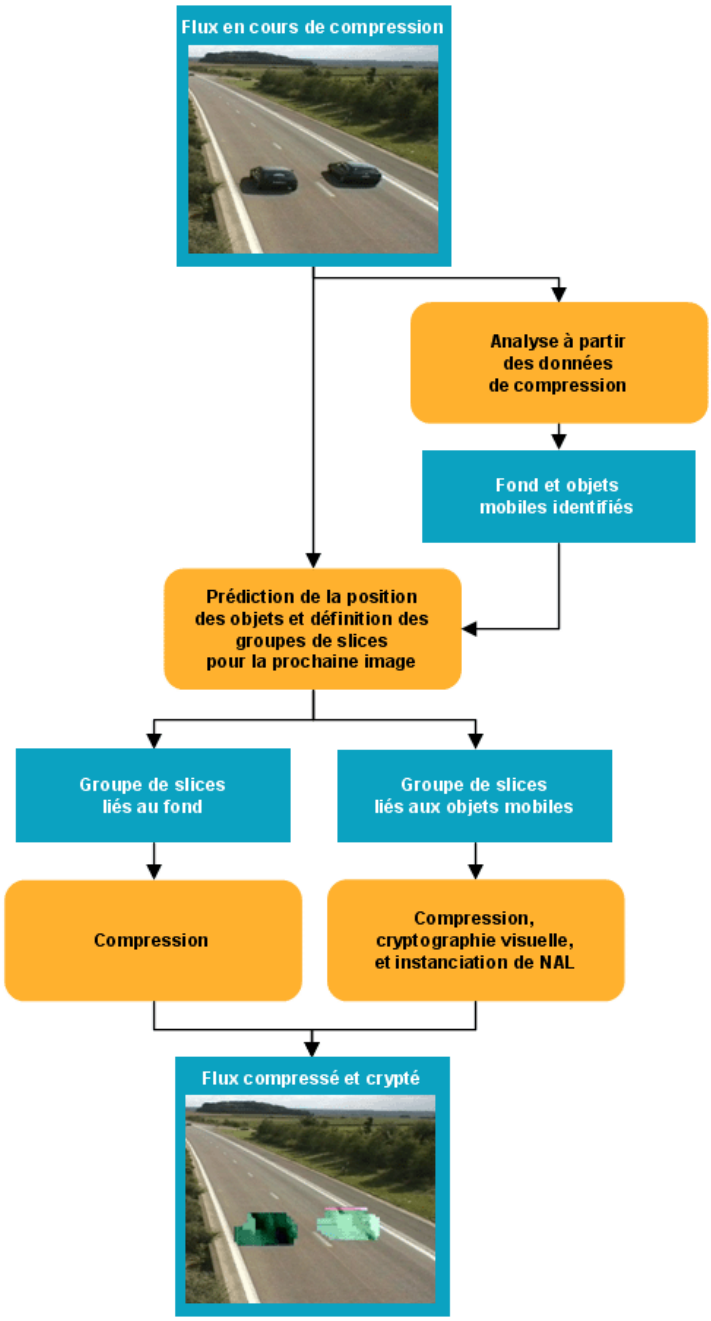


Figure 90 – Algorithme de cryptage visuel partiel.

Aujourd'hui, les outils de cryptage visuel sélectif ne sont pas disponibles. Si l'étude théorique a été menée et un brevet déposé, les contraintes de temps ont rendu impossible la finalisation de l'ensemble des algorithmes imaginés. Cette contribution est toutefois prévue pour un futur projet collaboratif sur lequel le laboratoire MMP de Thales est impliqué.

4.5 Adaptation de débit

En pouvant adresser indépendamment les blocs liés aux objets mobiles de ceux de l'arrière plan, il devient possible de faire une compression intelligente du flux vidéo, en adaptant les paramètres aux zones d'intérêt comme peut le faire JPEG2000. Ainsi, si le débit cible implique des matrices ou un pas de quantification (selon les standards) élevé(es), risquant donc de détériorer la qualité et les détails sur les visages ou autres éléments importants de la surveillance, il devient possible de traiter différemment ces deux types de zones. Une faible quantification sera appliquée sur les objets mobiles, alors que le fond sera plus fortement compressé.

La Figure 91 illustre l'algorithme permettant d'obtenir ces résultats avec un outil de compression – analyse conjointes. Il faut alors prédire la position des objets dans l'image courante à partir de l'historique de la séquence. Toutefois, cette approche peut également être utilisée pour un flux vidéo déjà compressé, et sera alors équivalente à un transcodage. Si l'on ne peut pas améliorer la qualité sur les régions d'intérêt, il sera possible d'appliquer une nouvelle quantification, plus forte, sur le fond, pour réduire le débit global du flux vidéo.
Cette approche est compatible avec la structure hiérarchique proposée en 4.3, mais cette dernière n'est pas indispensable puisqu'il est toujours possible de définir plusieurs pas de quantification selon les slices à l'intérieur du flux vidéo.

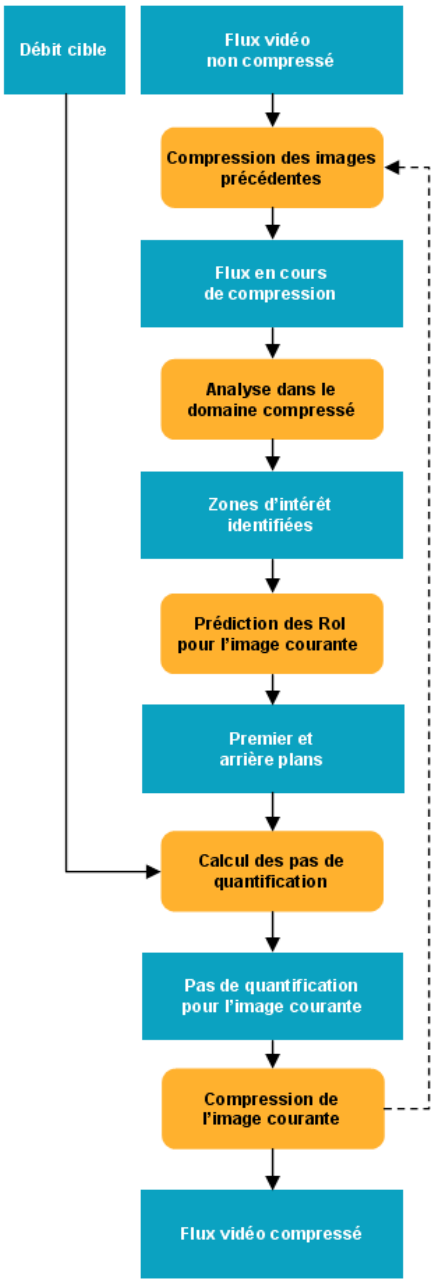


Figure 91 – Algorithme d'adaptation de débit

4.6 Protection aux erreurs

Un raisonnement équivalent à ceux présentés en 4.4 et 4.5 peut être conduit quant à la protection aux erreurs. Différents outils existent selon les standards, tels que les vecteurs de reconstruction sur les blocs Intra (*Concealment Vectors*) en H.264, mais ceux-ci s'appliquent au flux dans sa globalité, et ne permet donc pas de cibler une région plutôt qu'une autre dans l'image. En identifiant les zones liées aux objets mobiles, il devient possible de ne protéger que ces zones, plus importantes en termes d'usage de la vidéo.

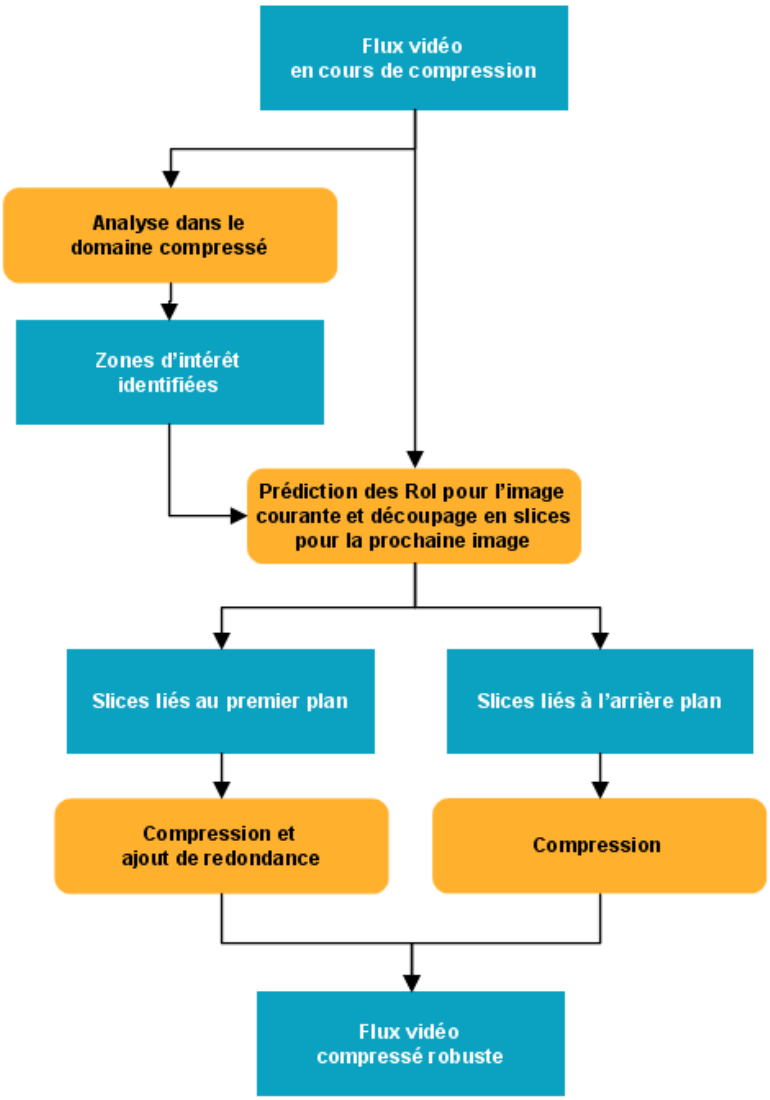


Figure 92 – Algorithme de protection aux erreurs

Ici encore, la structure de l'image (4.3) n'est pas indispensable pour le bon fonctionnement de la protection. Elle facilite toutefois la mise en place de la redondance, et permet surtout d'insérer cet outil de protection à tout moment sur le réseau. Ainsi, il peut ne pas être nécessaire d'ajouter de redondance sur une partie du canal de transmission, celui-ci étant très peu sujet aux erreurs, en revanche un lien sans fil pourra par exemple motiver la mise en place d'un module tel que celui de la Figure 92 entre les deux réseaux hétérogènes.

4.7 Tatouage et stéganographie

L'un des objectifs en ciblant les régions d'intérêt pour tous les algorithmes présentés ici est de pouvoir modifier un flux vidéo tout en conservant une qualité maximale sur les détails importants. D'une manière générale, l'image et la vidéo sont aujourd'hui de plus en plus utilisées comme preuves, et il devient donc primordial de pouvoir garantir l'authenticité et la qualité de ces dernières. Ces considérations passent par des métriques de qualité par exemple, qui assure que le flux issu d'une caméra fraîchement mise en place pourra servir à identifier un délinquant si un délit se produit, mais également par la garantie de la non-altération de la

vidéo entre sa captation et son rendu. Dans cette optique, nous proposons d'utiliser l'analyse dans le domaine compressé couplée aux outils de tatouage (Figure 93). Ainsi la première étape consiste-t-elle à concaténer les régions d'intérêt pour calculer le condensé, signature de son contenu. Pour préserver une qualité maximale sur les régions d'intérêt, et donc ne pas les altérer, le condensé crypté est tatoué uniquement sur les parties de l'image correspondant au fond. Avec une détermination de condensé par une fonction de hachage (*HASH*) type SHA-1, seuls 160 bits sont nécessaires pour valider l'intégrité du flux. Toutefois, en cas d'objets nombreux et/ou de taille importante, il se peut que la carte de tatouage intersectée avec l'arrière plan n'offre pas de capacité suffisante. Dans ce cas, seuls les premiers bits seront enfouis, présentant donc une sécurité plus faible que celle du condensé complet.
Au niveau du décodeur, le même algorithme d'analyse dans le domaine compressé est utilisé, garantissant la même identification des régions d'intérêt. Il est alors possible de recalculer le condenser sur ces régions, et en parallèle d'extraire celui enfoui dans le fond. Une comparaison entre les deux condensés est effectuée pour valider l'intégrité du flux depuis le tatouage jusqu'au décodage.

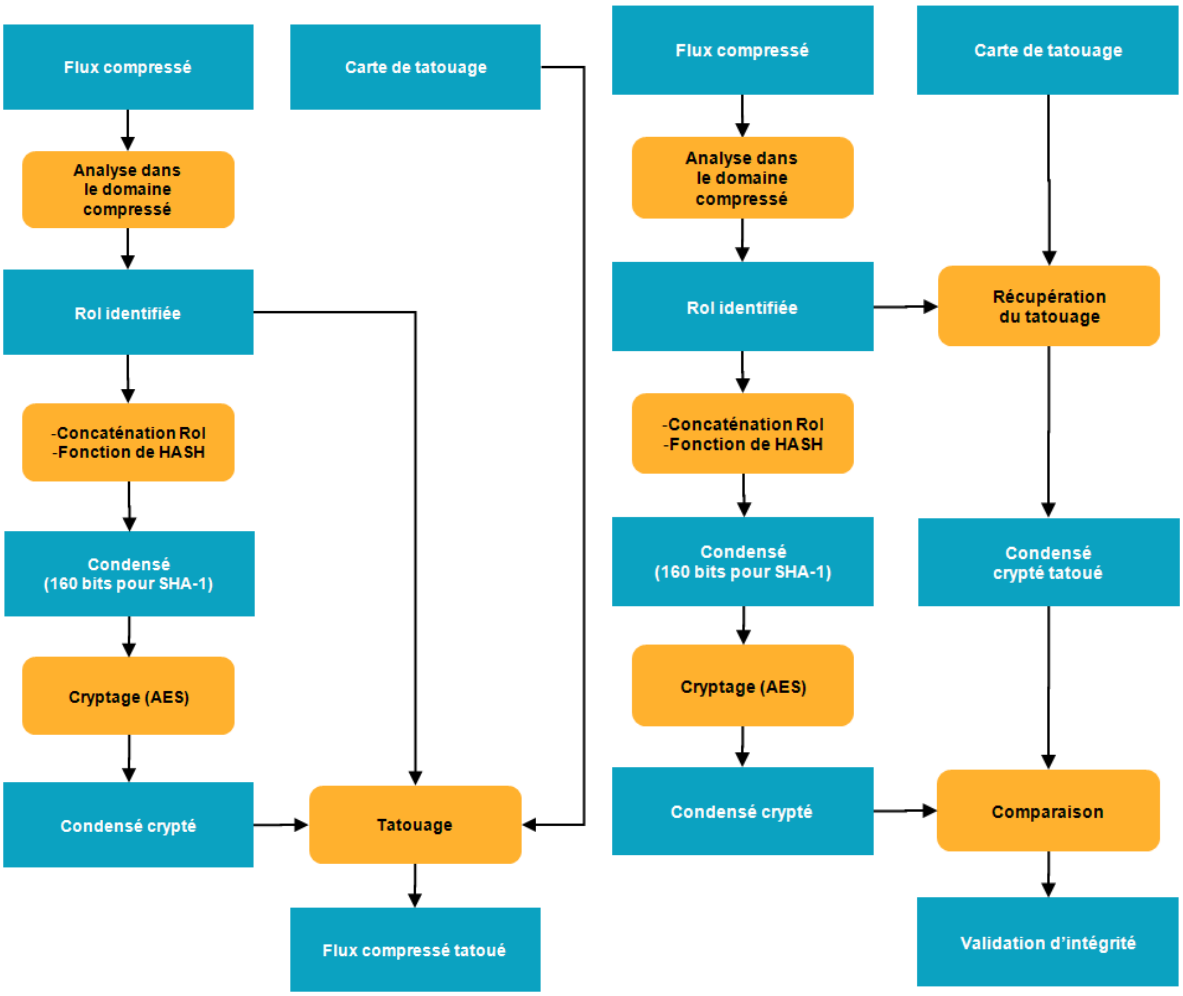


Figure 93 – Algorithme de tatouage (à gauche : côté codeur) et vérification d'intégrité (à droite : côté décodeur).

Cette application n'a pas besoin de la nouvelle structure d'image proposée. Elle n'a pas été implantée en mode « tatouage », mais l'étude a été enrichie par des outils de stéganographie sur flux H.264. Nous avons développé un algorithme permettant d'enfouir jusqu'à 4 kb/s

d'information dans un flux à 4 Mb/s, dans les coefficients issus de la transformée entière, mais également dans les vecteurs d'estimation de mouvement. Cela permet par exemple d'ajouter un canal audio stéganographié au sein d'une vidéo, et ainsi de transmettre le signal issu d'un micro couplé à une caméra, sans changer le réseau (tables de routage, débit, nombre de flux, etc.). Les métadonnées peuvent également être cachées dans le flux, ou tout autre type de message. L'algorithme est similaire à celui détaillé pour le tatouage, avec des cartes d'enfouissement calculé dynamiquement selon le contenu de la séquence (pour masquer davantage la modification du flux).

4.8 Conclusion : Système de vidéosurveillance

L'ensemble des outils proposés dans ce chapitre repose sur un facteur commun : la connaissance *a priori* des zones d'intérêts de l'image. Cet avantage sémantique permet d'entrevoir de nombreuses améliorations ou applications qui permettent de prétraiter, d'enrichir, de crypter ou de protéger le flux vidéo.
Une étude théorique conséquente a été menée, entre autre vis-à-vis des limitations des standards actuels face aux besoins émergeants liés à la vidéosurveillance. Nous avons proposé une nouvelle structuration du flux vidéo compressé, qui permet de faire intervenir la sémantique dans la structure même du flux vidéo compressé. Cet aspect, outre le fait de rendre possible des algorithmes qu'il était jusqu'alors compliqué de mettre en place efficacement, permettrait également de mettre en exergue les spécificités et les forces d'un codeur par rapport à un autre selon le domaine d'application. Ainsi les descripteurs permettant de différencier objets et groupe d'objets ne seront pas les mêmes selon que l'on souhaite compresser un journal TV ou une séquence de surveillance, et dans le domaine même de la surveillance, on choisira différents critères selon que l'on filme un couloir avec des passants ou une autoroute, etc.

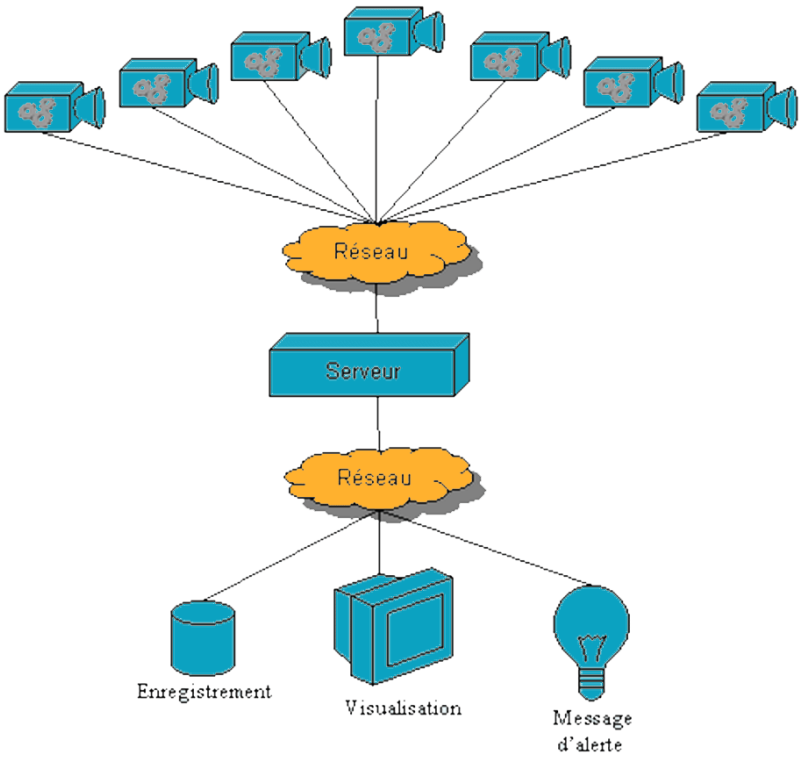


Figure 94 – Réseau de vidéosurveillance avec caméras intelligentes

En combinant l'ensemble des outils décrits dans ce chapitre, il devient possible d'imaginer un réseau de vidéosurveillance capable de se reconfigurer dynamiquement. Le débit global disponible est déterminé par l'infrastructure réseau elle-même, et chaque caméra est une « caméra intelligente », capable de déterminer un niveau de priorité du flux qu'elle produit sur le réseau. Cela peut être mis à l'œuvre suite à un apprentissage, qui déterminera par exemple qu'il est normal d'avoir une alternance de foule puis de quelques personnes sur un quai de métro, mais pas d'avoir un individu sur les rails. Un tel réseau est présenté Figure 94 : chaque caméra intelligente analyse en continu la scène qu'elle filme, et détermine un niveau de priorité du flux qu'elle génère (Figure 95). Le serveur prend en compte l'ensemble des niveaux de priorité pour fournir un ordre de débit aux caméras. L'adaptation de débit est alors faite en conservant autant de qualité que possible sur les zones d'intérêt. Si nécessaire chaque flux peut être crypté, et de la redondance peut être ajoutée si certaines parties du réseau sont plus que d'autres sujettes aux erreurs de transmission.

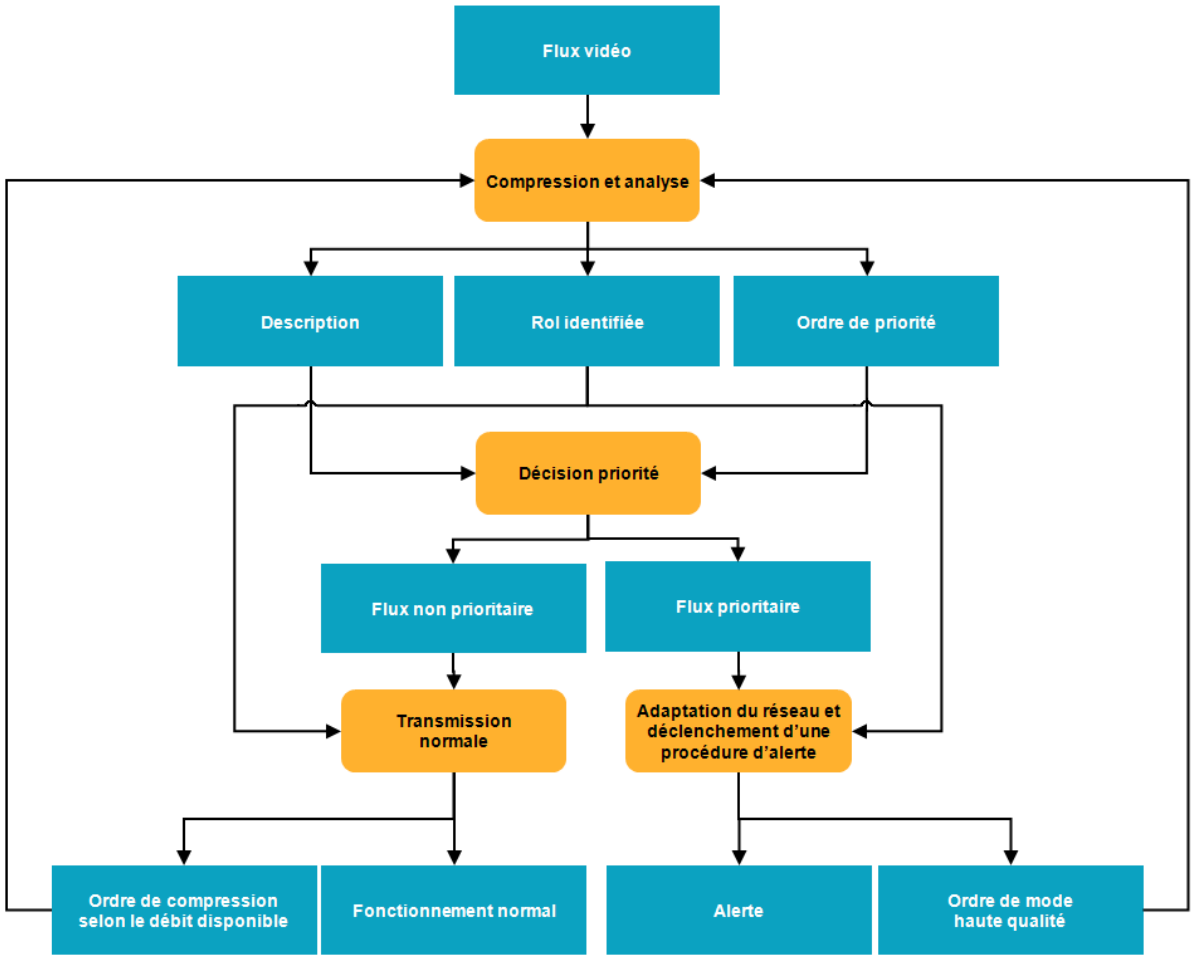


Figure 95 – Gestion de priorité sur un réseau muni de caméras intelligentes

L'ensemble des algorithmes présentés dans ce chapitre ont fait l'objet d'une famille de brevets déposés en juin et décembre 2008 (les références sont précisées dans la liste des publications, brevets et communications associées). Ces applications sont aujourd'hui développées selon les besoins et les offres du laboratoire MMP de Thales, ce qui explique les niveaux de maturité variés de ces implantations. L'objectif est à terme de disposer de

l'ensemble des outils pour pouvoir offrir un large éventail de solutions et pouvant répondre à des besoins clients différents.
Pour pallier les limitations des standards actuels aux spécificités de certains algorithmes, nous avons été amenés à suggérer une nouvelle structure du flux vidéo, ayant comme principale nouveauté une définition s'appuyant avant tout sur la sémantique des objets qu'elle contient. Pour rester compatible avec les normes utilisées en vidéosurveillance, nous avons également envisagé l'adaptation de cet outil construit sans contrainte de standardisation pour obtenir un compromis avec le standard H.264. Cette solution intermédiaire nous permet déjà d'enrichir le flux et de proposer de nouveaux services, mais il reste envisageable de suggérer notre approche lors de prochains comités de normalisation, puisqu'à la fois le département ARTEMIS de TELECOM SudParis et le laboratoire MMP de Thales sont impliqués dans les activités de standardisation, notamment autour de la compression et la vidéosurveillance.

Chapitre 5

Conclusion

Les travaux de cette thèse ont porté sur l'analyse et l'enrichissement de flux vidéo compressés, dans un contexte de vidéosurveillance.

Nous avons étudié et développé des outils de segmentation d'objets mobiles et d'analyse capables de traiter aussi bien des flux codés en MPEG-2 qu'en MPEG-4 Part 2 ou MPEG-4 Part 10 / H.264 AVC. Selon la syntaxe et le codage entropique du standard, l'étape consistant à parser le flux est plus ou moins complexe, et représente la principale source de l'écart en temps de calcul persistant entre les différentes vidéos prises en compte. Nous sommes en mesure d'analyser en temps réel, pour des vidéos 720x576 à 25 images par seconde, 14 flux MPEG-2, 8 flux MPEG-4 Part 2 ou 3 flux MPEG-4 Part 10 / H.264 AVC sur un cœur à 2.66GHz. Nous nous sommes ensuite interrogés sur l'usage qui peut être fait de notre approche pour aller au delà d'une simple segmentation. D'un côté, un démonstrateur d'aide à l'investigation permet de retrouver dans un corpus des véhicules répondant à un signalement. De l'autre, une phase de dépôts de brevets nous a permis de proposer des outils d'enrichissement de flux, depuis la protection inégale aux erreurs, la cryptographie visuelle ciblant les objets mobiles, la vérification d'intégrité par tatouage ou l'enfouissement de métadonnées dans le flux lui-même par stéganographie, jusqu'à une solution de surveillance globale intégrant l'ensemble des algorithmes conçus.

Le premier chapitre de ce mémoire de thèse présente ainsi les enjeux et techniques de la compression vidéo, ainsi que les normes de codage les plus utilisées en vidéosurveillance. Les différentes approches qui permettent d'utiliser les informations du domaine compressé sont également décrites dans l'état de l'art proposé.

La chaîne de traitements unifiée proposée au chapitre 2 constitue notre contribution majeure par rapport à l'état de l'art, puisqu'elle permet d'utiliser les mêmes outils quel que soit le type de flux en entrée. En reconstruisant des séquences basses résolutions et en calculant des vecteurs d'estimation de mouvement pour les blocs Intra des séquences, nous rendons possibles deux segmentations qui s'avèrent complémentaires : la première exploitant une soustraction de fond à l'échelle des blocs fournit de meilleurs résultats sur les images Intra, alors que la seconde qui s'appuie sur les vecteurs donne de meilleures performances sur les images prédites. La fusion qui suit permet, grâce à la dualité de ces deux approches, de compenser les taux de fausses alarmes ou de non détections souvent inconciliables avec l'analyse dans le domaine compressé. L'objectif visant à proposer une approche hybride capable de traiter des flux en résolution standard au-delà du temps réel est atteint, puisque l'analyse la plus lente, sur une vidéo H.264, atteint les 78 images par seconde. Une solution de compression/analyse conjointe a été mise au point à partir des résultats de l'estimation de mouvement effectuée par le codeur du laboratoire MMP. Cette solution hybride est pour sa part utilisable sur un ATOM N270 à 1.6GHz, ce qui constitue une preuve de concept quant aux futures solutions embarquées que Thales souhaite pouvoir offrir.

La validation à grande échelle du chapitre 3 permet de confirmer que cette approche aboutit à des résultats suffisants pour de nombreuses applications. Notre implication directe dans les projets collaboratifs français Infom@gic et européens Caretaker puis Vanaheim, en plus des projets internes à Thales, nous a fourni des corpus réalistes, variés et importants en volume de vidéos, mais aussi permis d'interagir avec différents partenaires travaillant sur des problématiques connexes. La précision des contours extraits est directement liée à la taille des blocs mise en œuvre dans la norme considérée, soit 16x16 pour nos outils initiaux en MPEG-2, et 4x4 pour les derniers corpus compressés en H.264. Le prototype réalisé démontre l'efficacité d'une approche de détection dans le domaine compressé. Les ressources

minimales nécessaires pour cet outil d'aide à l'investigation constituent l'avantage principal pour Thales pour développer des applications embarquées sur diverses plateformes.

Nous avons présenté un éventail d'applications qui seraient rendues possibles ou facilitées par l'analyse dans le domaine compressé, faisant l'objet du quatrième chapitre qui constitue une autre de nos principales contributions par rapport à l'état de l'art. En effet jusqu'alors, les différents outils s'appuyant sur l'analyse dans le domaine compressé se contentaient d'offrir une segmentation, autorisant éventuellement une levée d'alarme de bas niveau type détection d'activité, ou quelques explorations en termes de classification de piétons ou de groupes de personnes image par image. De notre point de vue, le caractère « bas coût » de nos traitements laisse entrevoir des produits de codage ou transcodage intégrant la segmentation des zones d'intérêt dans le flux. Dès lors, il devient possible de proposer des solutions de protection inégale aux erreurs, en ajoutant de la redondance pour les objets mobiles et non pour le fond. L'adaptation de débit d'un flux lors de sa compression ou en tant que transcodage est enrichie par l'analyse dans le domaine compressé, qui permet de conserver un maximum de détails, par exemple sur les visages ou les plaques d'immatriculation, tout en compressant davantage la végétation ou les bâtiments environnants, le tout *via* l'application de quantifications différentes sur ces deux types de zone pour chaque image. L'extraction d'une signature à partir des objets mobiles présents dans la séquence est aussi possible, et son enfouissement dans l'image qui a servi à son calcul, à l'intérieur de l'arrière plan, permet par tatouage d'offrir une authentification du contenu d'une vidéo. Une approche similaire permet de faire de la stéganographie avec une capacité d'enfouissement importante (4kb/s sur une vidéo à 4Mb/s) sans dégrader la qualité des régions d'intérêt (moins de 0.1 dB PSNR de perte). Enfin, l'utilisation de la segmentation proposée comme prétraitement avec l'identification des zones d'intérêt permet d'accélérer les algorithmes existant qui travaillent au niveau du pixel et d'obtenir des résultats identiques tout en accélérant les temps de traitement.

Vis-à-vis du choix de l'analyse dans le domaine compressé face aux outils plus habituels traitant des pixels, un expert étatique de la vidéo nous a fait part de sa vision sur l'automatisation des traitements : « il faut avant tout se soucier des résultats des algorithmes, et ensuite il faut trouver une solution pour les rendre temps réel ». Le problème est qu'aujourd'hui le nombre de caméra augmente plus vite que les ressources informatiques disponibles pour analyser les flux générés. De nombreux algorithmes sont considérés comme fournissant d'excellents résultats, mais il faut un processeur et une carte graphique de dernière génération, couplés à une grande capacité de mémoire vive pour les faire tourner en temps réel. Si l'on ne souhaite pas attendre cinq à dix ans que les progrès matériels nous autorisent effectivement à généraliser ce type d'approche, il est important d'explorer les ruptures technologiques. Cherchant à la fois des solutions de traitements massifs de corpus importants ou des outils qui puissent être portés sur des produits basse consommation, l'analyse dans le domaine compressé nous a permis de répondre à des besoins concrets en termes d'algorithmes contraints aux ressources disponibles.

D'un point de vue critique, plus la compression progresse, plus les données du domaine compressé sont décorellées du monde réel. Donc l'analyse dans le domaine compressé est *a priori* amenée à fournir des résultats se dégradant à mesure que les codecs évolueront. Toutefois, de nouvelles approches germent, parmi lesquelles l'analyse entropique du flux, qui permet déjà d'estimer la quantité de mouvement présent dans une image, ou même l'analyse sur les données liées à la transformée en ondelettes. Dans le cadre des travaux sur la vidéosurveillance, le temps d'adoption d'un nouveau standard et la limitation des paramètres de compression nous laisse entrevoir une utilité certaine des considérations abordées pour 10

à 15 ans dans le cadre de plateformes d'analyse classiques. Toutefois, ces outils garderont tout leur sens pour des solutions très basse consommation, qui profiteront à la fois des avancées technologiques et matérielles. Pour des serveurs de calcul, de nombreuses équipes de recherche ou d'industriels s'orientent aujourd'hui vers des solutions utilisant des calculs déportés sur carte graphique. Nous étudions également ces méthodes, qui combinées à l'analyse dans le domaine compressé pourraient par exemple autoriser le traitement d'une centaine de flux sur un unique ordinateur. A mi-chemin entre ces deux problématiques, nous commençons également l'étude aujourd'hui de solutions embarquées avec déport de calculs sur la puce de traitement graphique (chipset ION de NVIDIA). Ces outils laissent présager d'outils plus complexes intégrés dans des solutions basse-consommation (inférieure à 20W).

La principale limitation persistante est le manque de prise en charge de caméras mobiles. Cet aspect constitue la principale évolution algorithmique de bas niveau à ajouter à notre solution d'analyse. Les études préalables que nous avons menées en prenant en compte la bibliographie et l'expertise de membres du laboratoire MMP nous orientent vers un algorithme de classification des vecteurs bruts : la classe majoritaire sera alors *a priori* celle du mouvement de la caméra, dont la valeur moyenne pourra être soustraite à l'ensemble des vecteurs pour se replacer dans les conditions expérimentales qui furent l'objet de nos travaux. Le risque est alors de retrouver les inconvénients critiqués dans l'état de l'art, avec le problème d'incohérence si un objet occupe plus de la moitié du champ de la caméra. Cet obstacle pourra être contourné par le lissage temporel de l'estimation de mouvement globale (MEG), favorisant l'attribution du déplacement d'une petite zone si elle est cohérente avec le mouvement précédemment attribué à la prise de vue. Un tel cas de figure reste pour le moment isolé au sein des applicatifs envisagés pour l'industrialisation de nos outils.

Les autres perspectives de cette thèse sont étroitement liées aux orientations produits et services autour de l'analyse dans le domaine compressé chez Thales. Début 2011, un démonstrateur doit intégrer les aspects de prétraitement, avec un premier module proposant une compression et analyse conjointe en H.264 permettant d'identifier les zones d'intérêt. A l'autre bout de la chaîne de transmission, un second module sera en charge de la décompression du flux et de l'analyse au niveau pixel à l'intérieur des zones d'intérêt pour identifier les piétons dans une scène. Les autres services présentés font également l'objet de divers projets ou études futures pour le laboratoire MMP, ce qui nous permet d'envisager aujourd'hui pouvoir à termes proposer l'ensemble de la palette d'outils que nous avons imaginé.

ANNEXE : Texte du brevet FR 08.06837, 05/12/08
M. Leny, C. Le barz

PROCEDE ET DISPOSITIF POUR L'ENFOUISSEMENT D'UNE SEQUENCE BINAIRE DANS UN FLUX VIDEO COMPRESSE

L'invention concerne un procédé et un dispositif permettant d'enfouir un ou plusieurs types d'information représentées par une séquence binaire dans un flux vidéo déjà compressé avant sa transmission. Elle peut être utilisée, notamment, pour vérifier l'intégrité partielle d'un flux vidéo et a pour objectif de certifier que les zones d'intérêt d'une image dudit flux vidéo n'ont pas été modifiées lors de la transmission. L'invention s'applique, par exemple, dans un contexte de transmission numérique de vidéos dont on cherche à garantir que le contenu et en particulier certaines zones d'une image plus critiques en terme d'importance pour l'utilisateur final n'ont pas été modifiées par un intermédiaire malveillant. Ces zones peuvent correspondre, par exemple, à des objets mobiles. Un autre cas d'application de l'invention consiste à enfouir dans le flux vidéo compressé, un message de haut niveau fourni par une étape d'analyse dudit flux vidéo compressé. Ce type d'application permet par exemple à l'utilisateur final d'obtenir des informations sur le contenu de la séquence vidéo sans avoir à décompresser le flux vidéo au préalable.

L'invention peut, entre autre, être utilisée dans des applications mettant en œuvre la norme définie en commun par l'ISO MPEG et le groupe video coding de l'ITU-T dite H.264 ou MPEG-4 AVC (advanced video coding) qui est une norme vidéo fournissant une compression plus efficace que les normes vidéo précédentes tout en présentant une complexité de mise en œuvre raisonnable et orientée vers les applications réseau.

Dans la suite du document, le terme « premier plan » désigne le ou les objets mobiles dans une séquence vidéo, par exemple, un piéton, un véhicule, une molécule en imagerie médicale. A contrario, la désignation « arrière plan » est utilisée en référence à l'environnement ainsi qu'aux objets fixes. Ceci comprend, par exemple, le sol, les bâtiments, les arbres qui ne sont pas parfaitement immobiles ou encore les voitures stationnées.

Dans la description, l'expression « flux vidéo compressé » et l'expression « séquence vidéo compressée » font référence au même objet, à savoir un flux de données en sortie d'un module de compression vidéo dont l'entrée est une vidéo capturée en temps réel par une caméra ou encore une vidéo pré-enregistrée dans un fichier. L'expression « marqueurs par tatouage » fait référence, dans la suite de la description, à une information enfouie au sein d'un flux image ou vidéo via un procédé de tatouage.

Les systèmes de vidéo surveillance sont de plus en plus répandus. Ces derniers utilisent pour transmettre les informations vidéos ou autres des systèmes de diffusion composés de réseaux hétérogènes filaires ou sans fils dont l'architecture peut être complexe. A ce sujet, l'un des problèmes qui se pose est l'obtention d'informations sur le contenu de la séquence vidéo transmise sans décompresser au préalable le flux vidéo compressé reçu. Le type d'informations visé peut, par exemple, servir à garantir l'intégrité du contenu de séquences vidéo lors de leur diffusion dans un contexte où elles peuvent être interceptées et modifiées par un tiers malveillant. En particulier, certaines zones d'une séquence vidéo peuvent être d'un intérêt plus important pour l'utilisateur, par exemple les zones identifiant des objets mobiles, par opposition à des zones de moindre intérêt, par exemple des zones comme le sol ou le ciel pour lesquelles la garantie de l'intégrité est moins cruciale. Un autre type d'informations utile à l'utilisateur peut porter, par exemple, sur les caractéristiques des zones d'intérêts d'une image, en particulier des informations sur la taille ou la couleur des dites zones.

L'art antérieur comprend diverses méthodes permettant d'enfouir une information au sein d'un flux vidéo, en particulier, les techniques de tatouage numérique de flux multimédia comme celle décrite dans la référence suivante « Combining low-frequency and spread spectrum watermarking », 1999, Jiri Fridrich. Ce procédé consiste à tatouer l'intégralité d'une image et s'effectue sur des flux vidéo non compressés, ce qui présente comme inconvénient d'être complexe à mettre en œuvre sur des processeurs à ressources limitées.
D'autres techniques permettant d'enfouir une information par tatouage dans un flux vidéo déjà compressé existent mais elles ne permettent pas d'identifier et de traiter

uniquement certaines zones de plus grande importance au sein d'une image et non l'intégralité du contenu de l'image.

Par exemple, la demande de brevet français 2896938 décrit un procédé de tatouage de données numériques utilisant les coefficients de transformée en cosinus discrète plus connus sous l'appellation anglo-saxonne Discrete Cosine Transform (DCT) pour enfouir une signature au sein d'une vidéo. L'ensemble de la vidéo est considéré ici sans effectuer au préalable une analyse permettant de déterminer les zones les plus critiques, du point de vue de l'utilisateur.

D'autres problèmes ne sont pas résolus par l'art antérieur tels que :

- Le tatouage d'une image sans modification aucune des zones d'intérêt et sans décompression de la séquence vidéo.
- La génération d'informations pertinentes concernant les dites zones d'intérêt et leur enfouissement dans le flux vidéo compressé, toujours sans décompression de la séquence vidéo,
- L'insertion de marqueurs par tatouage permettant de vérifier uniquement l'intégrité de certaines zones pertinentes au sein de la séquence vidéo et non l'ensemble de la séquence comme cela est le cas plus traditionnellement.

Un des objets de la présente invention est d'offrir un procédé d'enfouissement d'une information sous forme de séquence binaire dans un flux vidéo compressé. Cette information concerne certaines zones d'intérêts de la séquence vidéo et a pour objet, par exemple, une vérification de l'intégrité des dites zones ou une alerte sur des éléments caractéristiques des dites zones telles que la taille ou la couleur. Un autre objet de l'invention est de permettre l'insertion de marqueurs par tatouage sans modifier les zones d'intérêt de la séquence. A cet effet, l'invention a pour objet un procédé d'enfouissement d'une séquence binaire dans une séquence vidéo ou un flux vidéo compressé, ledit flux pouvant être décomposé en plusieurs types d'objets, le procédé s'appliquant sur au moins une image contenue dans ladite séquence vidéo caractérisé en ce qu'il comporte au moins les étapes suivantes :

a) Analyser la séquence vidéo dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt ,

b) Générer une carte de tatouage définissant l'ensemble des blocs de pixels de la séquence éligibles pour l'opération d'enfouissement, ainsi que deux coefficients C_1 et C_2 issus d'une transformée en fréquence pour chacun des dits blocs. Lesdits coefficients C_1 et C_2 sont tirés aléatoirement parmi l'ensemble des coefficients obtenus par application d'une transformée en fréquence sur un bloc de pixels donné.

c) Pour l'image compressée courante, exclure de la carte de tatouage les blocs associés à la zone d'intérêt, ainsi que ceux dont les coefficients C_1 et C_2 définis par la carte de tatouage ne répondent pas au critère suivant :

   $abs(abs(C_1)-abs(C_2)) < S$ où S est un seuil prédéterminé et abs() la fonction valeur absolue d'un entier.

d) Appliquer une fonction de tatouage numérique sur chaque bloc disponible afin d'obtenir un flux compressé tatoué par une séquence binaire de la façon suivante :

   - Pour insérer un bit « 1 » de ladite séquence binaire,
     - Si $abs(C_1) > abs(C_2)$, on ne change rien
     - Si $abs(C_1) \leq abs(C_2)$, on calcule $\varepsilon = abs(C_2) - abs(C_1)$ et on modifie la valeur de C_1, $C_1 = C_1 + \varepsilon + 1$ si $C_1>0$, $C_1= C_1 - \varepsilon - 1$ sinon.
   - Pour insérer un bit « 0 » de ladite séquence binaire,
     - Si $abs(C_1) < abs(C_2)$, on ne change rien
     - Si $abs(C_1) \geq abs(C_2)$, on calcule $\varepsilon = abs(C_1) - abs(C_2)$ et on modifie la valeur de C_2, $C_2 = C_2 + \varepsilon + 1$ si $C_2>0$, $C_2= C_2 - \varepsilon - 1$ sinon.

Selon un mode de réalisation, la séquence binaire à enfouir est un condensé de l'image obtenu via l'étape suivante :

- Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt et appliquer au résultat une fonction de hachage visuel générant en sortie un condensé de l'image.

Selon un mode de réalisation, le condensé de l'image est crypté via une fonction de cryptage.

Selon un mode de réalisation, le flux compressé tatoué est traité afin de vérifier l'intégrité de la séquence vidéo selon les étapes suivantes :

- Analyser le flux compressé tatoué dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt ,
- Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt et appliquer au résultat une fonction de hachage visuel générant en sortie un condensé de l'image ,
- Récupérer le condensé tatoué dans le flux compressé tatoué à partir de la carte de tatouage
- Effectuer une comparaison dudit condensé tatoué avec le condensé.

Selon un mode de réalisation, le flux compressé tatoué est traité afin de vérifier l'intégrité de la séquence vidéo selon les étapes suivantes :

- Analyser le flux compressé tatoué dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt ,
- Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt et appliquer au résultat une fonction de hachage visuel générant en sortie un condensé de l'image ,
- Appliquer, une fonction de cryptage au condensé précédemment obtenu afin d'obtenir un condensé crypté,
- Récupérer le condensé tatoué dans le flux compressé tatoué à partir de la carte de tatouage,
- Effectuer une comparaison dudit condensé tatoué avec le condensé crypté.

Selon un mode de réalisation, la fonction de cryptage met en œuvre un algorithme de chiffrement asymétrique ou un algorithme de chiffrement AES (Advanced Encryption Standard).

Selon un mode de réalisation, la fonction de hachage visuel met en œuvre l'algorithme SHA-1 et le condensé crypté a une taille de 160 bits.

Selon un mode de réalisation, la séquence binaire à enfouir est un message comportant une indication sur les caractéristiques des zones d'intérêt et est fourni par une étape d'analyse préalable, par exemple dans le domaine compressé

Selon un mode de réalisation, la séquence vidéo est produite par un standard MPEG ou un standard H.264.

L'invention a également pour objet un dispositif pour tatouer numériquement au moins une partie d'un flux vidéo ou d'une séquence vidéo compressée comportant un émetteur et un récepteur caractérisé en ce que :

- ledit émetteur comporte au moins les éléments suivants : un module d'analyse dans le domaine compressé, un module de hachage visuel, un module de tatouage numérique et un module de transmission du flux tatoué,
- ledit récepteur comporte au moins les éléments suivants : un module de transmission, un module d'analyse, un module de hachage visuel, un module de validation d'intégrité.

D'autres caractéristiques et avantages du procédé et du dispositif selon l'invention apparaîtront mieux à la lecture de la description qui suit d'un exemple de réalisation donné à titre illustratif et nullement limitatif annexé des figures qui représentent :

- Les figures 1 à 4, les résultats obtenus par une analyse dans le domaine compressé,
- La figure 5, un exemple de procédé de tatouage selon l'invention appliquée à un flux vidéo en cours de compression,
- La figure 6, un exemple de procédé de vérification d'intégrité sur un flux compressé tatoué via le procédé selon l'invention,
- La figure 7, un exemple de schéma pour un émetteur vidéo adapté à mettre en œuvre le procédé de tatouage selon l'invention.
- La figure 8, un exemple de schéma pour un récepteur vidéo adapté à mettre en œuvre le procédé de tatouage selon l'invention et permettant la vérification de l'intégrité de la séquence vidéo.

Afin de mieux faire comprendre le fonctionnement du procédé selon l'invention, la description comprend un rappel sur la manière d'effectuer une analyse dans le domaine compressé, tel qu'il est décrit par exemple dans la demande de brevet US

2006 188013 en référence aux figures 1, 2, 3 et 4 et aussi dans les deux références suivantes :

Leny, Nicholson, Prêteux, "De l'estimation de mouvement pour l'analyse temps réel de vidéos dans le domaine compressé", GRETSI, 2007.

Leny, Prêteux, Nicholson, "Statistical motion vector analysis for object tracking in compressed video streams", SPIE Electronic Imaging, San Jose, 2008.

Dans la suite du texte le terme « bloc » fait référence à un ensemble de pixels d'une image formant ensemble une matrice et le terme « bloc transformé» identifie le résultat obtenu via une fonction de transformée permettant un passage dans le domaine fréquentiel, appliqué audit bloc. Par exemple, un bloc de 8x8 pixels représenté par une matrice sera transformé en une matrice à 8 lignes et 8 colonnes contenant 64 coefficients C_i..

En résumé certaines techniques utilisées dans les standards MPEG et exposées dans ces articles consistent à diviser la compression vidéo en deux étapes. La première étape vise à compresser une image fixe. L'image est tout d'abord divisée en blocs de pixels (de 4x4 à 16x16 selon les standards MPEG-1/2/4), qui subissent, par la suite, une transformée permettant un passage dans le domaine fréquentiel telle que la transformée en cosinus discrète (DCT) ou la transformée entière, puis une quantification permet d'approximer ou de supprimer les hautes fréquences auxquelles l'œil est moins sensible. Enfin les données quantifiées sont codées entropiquement. A cet effet, la quantification permet de supprimer ou atténuer les hautes fréquences moins sensibles à l'œil et ainsi de réduire la quantité d'informations. La seconde étape a notamment pour objectif de réduire la redondance temporelle. Elle permet de prédire une image à partir d'une ou plusieurs autres image(s) précédemment décodée(s) au sein de la même séquence (prédiction de mouvement). Pour cela, le processus recherche dans ces images de référence le bloc qui correspond le mieux à la prédiction souhaitée. Seul un vecteur (Vecteur Estimation de Mouvement, également connu sous l'appellation anglo-saxonne Motion Vector), correspondant au déplacement du bloc entre les deux images ainsi qu'une erreur résiduelle permettant de raffiner le rendu visuel sont conservés.

Ces vecteurs ne correspondent toutefois pas nécessairement à un mouvement réel d'un objet dans la séquence vidéo mais peuvent s'apparenter à du bruit. Différentes étapes sont donc nécessaires pour utiliser ces informations afin d'identifier les objets mobiles. Les travaux décrits dans la publication précitée de Leny et al, « De

l'estimation de mouvement pour l'analyse temps réel de vidéos dans le domaine compressé », et dans la demande de brevet US 2006 18 8013 précitée ont permis de délimiter cinq fonctions rendant l'analyse dans le domaine compressé possible, ces fonctions et les modules les mettant en œuvre sont représentées à la figure 1 :

1) un décodeur basse résolution (LRD – Low-Res Decoder) permet de reconstruire l'intégralité d'une séquence à la résolution du bloc, supprimant à cette échelle la prédiction de mouvement ;

2) un générateur de vecteurs estimation de mouvement (MEG – Motion Estimation Generator) détermine quant à lui des vecteurs pour l'ensemble des blocs que le codeur a codé en mode "Intra" (au sein d'images Intra ou prédites) ;

3) un module de segmentation basse résolution d'objets (LROS – Low-Res Object Segmentation) s'appuie pour sa part sur une estimation du fond dans le domaine compressé grâce aux séquences reconstruites par le LRD et donne donc une première estimation des objets mobiles ;

4) un filtre d'objets basé sur le mouvement (OMF – Object Motion Filtering) utilise les vecteurs en sortie du MEG pour déterminer les zones mobiles à partir de l'estimation de mouvement ;

5) enfin un module permettant d'établir une décision coopérative (CD – Cooperative Decision) à partir de ces deux segmentations, prend en compte les spécificités de chaque module selon le type d'image analysée (Intra ou prédite).

L'intérêt principal de l'analyse dans le domaine compressé porte sur les temps de calcul et les besoins en mémoire qui sont considérablement réduits par rapport aux outils d'analyse classiques. En s'appuyant sur le travail effectué au moment de la compression vidéo, les temps d'analyse sont actuellement de 10 à 20 fois le temps réel (250 à 500 images traitées par seconde) pour des images 720x576 4:2:0.

Un des inconvénients de l'analyse dans le domaine compressé telle que décrite dans les documents précités est que le travail est effectué sur l'équivalent d'images basse résolution en manipulant des blocs composés de groupes de pixels. Il en résulte que l'image est analysée avec moins de précision qu'en mettant en œuvre les algorithmes usuels utilisés dans le domaine non compressé. De plus, les objets trop petits par rapport au découpage en blocs peuvent passer inaperçus.

Les résultats obtenus par l'analyse dans le domaine compressé sont illustrés par la figure 2 qui montrent l'identification de zones contenant des objets mobiles. La figure 3 schématise l'extraction de données spécifiques telles que les vecteurs estimation

de mouvement représentés sur la gauche de la figure et la figure 4 des cartes de confiance basse résolution obtenues correspondant aux contours de l'image, également représentées sur la gauche de la figure.

La figure 5 schématise un exemple de réalisation du procédé selon l'invention dans lequel, un condensé des zones d'intérêt d'une image est calculé par exemple via une fonction de signature souple, plus connue de l'Homme du métier sous le terme anglo-saxon « visual hash ». Ce condensé est ensuite enfoui dans le flux vidéo compressé à transmettre.

Le flux vidéo compressé 10 est transmis à une première étape d'analyse 12 dans le domaine compressé connue de l'Homme du métier ayant pour fonction d'extraire les zones d'intérêt définies par l'utilisateur. Ainsi, le procédé dispose par exemple, d'une séquence de masques comprenant des blobs (régions ayant reçues un label identique) liés aux objets mobiles. Les masques peuvent être des masques binaires.

Cette analyse dans le domaine compressé a permis de définir pour chaque image ou pour un groupe d'images défini GoP, d'une part différentes zones Z1i appartenant à un premier plan P1 et d'autres zones Z2i appartenant à un deuxième plan P2 d'une des images de la séquence vidéo. L'analyse peut être effectuée en mettant en œuvre le procédé décrit dans la demande de brevet US 2006 18 8013 précitée. Toutefois, tout procédé permettant d'obtenir une sortie de l'étape d'analyse se présentant sous forme de masques par image, ou tout autre format ou paramètres associés à la séquence vidéo compressée analysée pourra aussi être mis en œuvre en sortie de l'étape d'analyse dans le domaine compressé. A l'issue de l'étape d'analyse, le procédé a permis d'isoler les zones d'intérêt,13, Z1i dont on souhaite protéger l'intégrité du contenu. Dans un autre mode de réalisation, les zones d'intérêt Z1i peuvent être à contrario des objets fixes dont on cherche à identifier sans ambigüité le contenu, par exemple des panneaux de signalisation. Dans un cas d'application générale, le procédé selon l'invention s'applique à tout type de zone d'intérêt que l'étape d'analyse 12 a permis d'isoler.

Afin de permettre une authentification fiable des zones d'intérêts au sein d'une image, il est nécessaire de générer une signature qui soit directement liée au contenu desdites zones. Pour se faire, le procédé utilise par exemple une technique connue de l'Homme du métier sous la dénomination anglo-saxonne « visual hash » ou fonction de hachage visuel en français. On appelle fonction de hachage visuel un procédé permettant de calculer un condensé d'une image (ou de tout type de

données multimédia). Contrairement aux fonctions de hachage cryptographique, une telle fonction peut permettre de générer un condensé identique pour deux images différentes, sous réserve que celles-ci soient suffisamment proches du point de vue perceptuel. Le condensé résultant de cette opération est aussi appelé signature souple, le qualificatif « souple » étant employé par opposition au qualificatif « strict ». Par contre, le condensé doit être différent dès lors que l'image subit une altération de sa sémantique, par exemple l'ajout d'un personnage ou d'un objet dans une scène, la modification d'images ou de texte. De ce fait, la problématique de définition formelle de la sémantique d'une image rend la conception de fonctions de hachage visuel délicate.

Néanmoins, il est possible de définir un critère de localité, qui peut être considéré comme valide dans la grande majorité des cas. Une altération contingente, c'est-à-dire due à une compression du flux vidéo par exemple, va donner lieu à des modifications du signal de faible amplitude, mais réparties de manière relativement uniforme sur l'ensemble de l'image. Par contre, une altération de la sémantique résultera en une modification forte mais localisée des données. On cherchera donc à utiliser des fonctions de hachage à seuil, qui tolèrent des modifications inférieures à une certaine valeur mais réagissent à des variations localisées trop fortes.

Cette fonctionnalité est mise en œuvre lors de l'étape 14. Le procédé sélectionne les blocs de transformée en cosinus discrète (blocs DCT) appartenant aux zones d'intérêt au sein du flux vidéo compressé et les concatène pour obtenir un message. Ce message est fourni en entrée à un algorithme de hachage visuel, 14, qui permet de calculer un condensé 15 de la partie de l'image constituée par lesdites zones d'intérêt. Dans le cas d'une utilisation du procédé couplée au standard de compression vidéo H.264, les blocs à considérer pour l'obtention du message sont des blocs de transformée entière. De façon générale, toute transformée équivalente qui pourrait être définie dans d'éventuels futurs ou anciens standards de compression vidéo reste compatible du procédé selon l'invention. L'algorithme de hachage visuel utilisé peut être, par exemple, un algorithme de type SHA-1 tel que décris par le document référencé « FIPS PUB 180-2, Secure Hash Standard » disponible à l'adresse internet http://csrc.nist.gov/encryption/tkhash.html. Dans ce cas, le message d'entrée dudit algorithme doit avoir une longueur de 2^{64} bits, ledit message est éventuellement complété pour atteindre la longueur requise. Cette éventualité est prévue dans le standard SHA-1 qui reste donc compatible dans ce

cas. Le condensé 15 a une longueur de 160 bits dans ce mode de réalisation. Tout autre algorithme permettant le calcul d'un condensé d'une image peut être utilisé.
L'étape 16 du procédé met ensuite en œuvre une fonction de cryptage ou chiffrement du condensé 15 obtenu via l'étape précédente 14. L'algorithme de chiffrement utilisé peut être, par exemple, un algorithme de chiffrement asymétrique ou un algorithme AES (Advanced Encryption Standard). Un condensé crypté 17 est obtenu après cette étape. Une fois ce résultat obtenu, la suite du procédé consiste à enfouir ledit condensé crypté 17 dans les zones de moindre importance de l'image par le biais d'une étape de tatouage numérique 18.
Au préalable, une carte de tatouage 11 est définie avant de commencer à traiter la séquence et cela sans connaissance a priori du flux vidéo compressé 10. Cette carte détermine tous les blocs de l'image qui peuvent contenir l'information à enfouir. Il peut arriver que des blocs associés aux zones d'intérêt Z1i déterminées à l'issu de l'étape d'analyse dans le domaine compressé 12 appartiennent à cette carte de tatouage, leur position n'étant pas connue au préalable. Comme il n'est pas souhaitable d'altérer le rendu visuel des dites zones, même de façon peu perceptible, on se contentera de vérifier avant le tatouage de chaque bloc son appartenance à une zone d'intérêt : si c'est le cas, le processus de tatouage passe automatiquement au bloc suivant, sinon le bloc courant est éligible pour enfouir la portion du condensé courante. Dans un mode de réalisation mettant en œuvre une compression vidéo utilisant la transformée en cosinus discrète DCT (Discrete Cosine Transform), chaque bloc correspond à 8x8 pixels. Pour une résolution standard de 720x576 pixels, il y a donc 90x72 blocs pour une image, soit 6480 blocs potentiels auxquels il convient d'enlever les blocs appartenant aux zones d'intérêt Z1i. Dans l'exemple de mise en œuvre de l'étape 16, le condensé crypté 17 a une longueur de 160 bits, il convient dans ce cas de sélectionner 160 blocs parmi ceux disponibles pour permettre d'enfouir la totalité du condensé crypté dans le flux compressé 10. La carte de tatouage 11 étant générée au démarrage du procédé, à ce stade il n'est pas possible de connaitre le nombre exact de blocs disponibles pour le tatouage car l'étape d'analyse 12 permettant de déterminer les zones d'intérêt 13 et donc le nombre de blocs associés n'a pas encore été effectuée. Lors de la génération de la carte de tatouage, il n'est donc pas possible de savoir si un nombre de blocs suffisant existe pour couvrir la longueur totale du condensé crypté 17, dans notre exemple 160 bits. Pour résoudre ce problème, une solution consiste à déterminer

arbitrairement un nombre maximum de blocs à sélectionner, d'effectuer le tatouage de ces blocs tant qu'ils ne font pas partie des zones d'intérêt et tant que la fin de l'image n'est pas atteinte. Si à la fin de ce processus, les 160 bits n'ont pas tous été utilisés, l'opération est tout de même arrêtée. Le condensé crypté tatoué au sein du flux compressé 10 aura dans ce cas une longueur inférieure à celui calculé lors de l'étape 16.

L'étape 18 de tatouage numérique permettant l'enfouissement du condensé crypté 17 dans le flux compressé 10, à partir de la connaissance des zones d'intérêt 13 d'une image et de la carte de tatouage 11 peut, par exemple, être réalisée de la façon décrite dans la demande de brevet français 2896938.

A l'établissement de la carte de tatouage, une fois les blocs sélectionnés, pour chacun d'eux, deux coefficients transformés C_1 et C_2 sont tirés aléatoirement. Lesdits coefficients sont testés de la façon suivante : si abs(abs(C_1)- abs(C_2)) , où la notation abs() correspond à la fonction valeur absolue d'un nombre, est inférieur à un seuil prédéterminé, le tatouage peut avoir lieu sur le bloc correspondant car le rendu visuel sera imperceptible. Dans le cas contraire, on passe au bloc suivant.

La relation d'ordre entre abs(C_1) et abs(C_2) est alors testée et ces 2 coefficients sont modifiés si besoin afin qu'ils reflètent la valeur du bit du condensé crypté 17 à enfouir « 0 » ou « 1 ». L'algorithme suivant est mis en œuvre :

- Pour insérer un bit « 1 »,
    - Si abs(C_1) > abs(C_2), on ne change rien
    - Si abs(C_1) $\leq$ abs(C_2), on calcule ε = abs(C_2) – abs(C_1) et on modifie la valeur de C_1, $C_1 = C_1 + \varepsilon + 1$ si $C_1>0$, $C_1= C_1 - \varepsilon - 1$ sinon.
- Pour insérer un bit « 0 »,
    - Si abs(C_1) < abs(C_2), on ne change rien
    - Si abs(C_1) $\geq$ abs(C_2), on calcule ε = abs(C_1) – abs(C_2) et on modifie la valeur de C_2, $C_2 = C_2 + \varepsilon + 1$ si $C_2>0$, $C_2= C_2 - \varepsilon - 1$ sinon.

Dans une variante de réalisation, l'information tatouée au sein du flux compressé peut consister en une autre donnée telle que, par exemple, une alarme déclenchée suite à une opération d'analyse d'activité effectuée sur la séquence vidéo en lieu et place du condensé permettant la vérification d'intégrité. L'analyse dans le domaine compressée peut en effet aboutir à la génération de messages de plus haut niveau, tels que, par exemple la présence d'un véhicule mobile dans une zone donnée ou

une information sur les caractéristiques d'un objet telles que sa couleur ou sa taille, ou bien encore tout simplement une alarme indiquant qu'un objet d'une taille spécifiée a été détecté dans une zone de l'image. Ces messages peuvent également être enfouis dans la séquence via le procédé d'enfouissement selon l'invention. L'étape d'analyse 12 permet dans ce cas de produire une information pertinente et exploitable directement par l'étape de tatouage 18.

La figure 6 illustre une variante de réalisation dans laquelle le flux compressé tatoué 19 obtenu via le procédé selon l'invention décrit précédemment est utilisé afin de vérifier l'intégrité des zones d'intérêt de la séquence vidéo transmise. Le flux compressé tatoué 19 est soumis aux mêmes étapes 12, 14 et 16 que précédemment afin d'obtenir un condensé crypté 17 identique à celui décrit sur la figure 5. En parallèle une étape 20 permet de récupérer le condensé crypté tatoué 21 enfouit au sein du flux compressé tatoué 19. Cette étape est réalisée en utilisant la carte de tatouage 11 précédemment décrite. Une comparaison 22 des deux condensés 17 et 21 permet d'obtenir une information de validation de l'intégrité des zones d'intérêts de la séquence 23. Cette comparaison est faite sur la longueur du condensé crypté tatoué 21 qui peut être plus court que celui 17 généré via les étapes de hachage 14 et cryptage 16, comme expliqué précédemment. Dans ce cas la comparaison est faite uniquement sur la partie commune des deux condensés 21 et 17, validant seulement en partie l'intégrité. Une alarme mineure précisant que l'image considérée n'est validée que partiellement peut être renvoyée à l'utilisateur final dans ce cas.

La figure 7 représente un schéma bloc d'un dispositif selon l'invention représentant un émetteur vidéo 30 adapté pour mettre en œuvre les étapes décrites avec la figure 5. L'émetteur vidéo 30 comprend un module d'analyse vidéo 31 recevant le flux vidéo compressé F et adapté à déterminer les différentes zones d'intérêt Z1i, un module 32 réalisant une fonction de hachage puis cryptage des coefficients transformés des zones Z1i et fournissant à sa sortie un condensé crypté de l'image, un module de tatouage numérique 33 adapté à insérer ledit condensé au sein du flux compressé sans altérer le rendu visuel de la séquence vidéo et enfin un module de communication 34 permettant au dispositif de transmettre à la fois le flux vidéo compressé tatoué F_t et une carte de tatouage générée en début de processus par le module 33.

La figure 8 représente un schéma bloc d'un dispositif selon l'invention représentant un récepteur vidéo 40 adapté pour mettre en œuvre les étapes décrites à la figure 6.

Le récepteur vidéo 40 comprend un module de réception 41 permettant au dispositif de recevoir à la fois un flux vidéo compressé tatoué F_t et une carte de tatouage associée générés tous deux par le procédé selon l'invention décris par la figure 5. Le récepteur vidéo comprend également un module 42 qui effectue une analyse dans le domaine compressé du flux F_t et permet d'identifier les différentes zones d'intérêt Z1i, un module 43 réalisant une fonction de hachage puis cryptage des coefficients transformés des zones Z1i et fournissant à sa sortie un condensé crypté de l'image, un module 44 réalisant la récupération dans le flux F_t d'un condensé crypté tatoué et la comparaison avec le condensé crypté délivré par le module 43. Le module 44 produit en sortie une décision V de validation d'intégrité des zones d'intérêt de la séquence.

Le procédé et le système selon l'invention présentent plusieurs avantages notamment de garantir que certaines zones d'intérêts de l'image n'ont pas été modifiées. Le fait d'utiliser l'analyse dans le domaine compressé permet d'effectuer l'ensemble des traitements sans décompresser le flux vidéo. Les ressources matérielles sont réduites de ce fait comparativement aux méthodes de l'art antérieur et permettent l'utilisation de systèmes embarqués. La signature, ou condensé crypté, obtenue via le procédé selon l'invention permet de cibler uniquement les zones d'intérêt et son enfouissement par tatouage sur le reste de l'image permet une vérification indépendante de chaque image tout en préservant lesdites zones d'intérêt de toute modification. Un autre avantage réside dans le fait de pouvoir enfouir au sein du flux vidéo compressé des messages fournissant une information sur les caractéristiques d'un objet présent dans la séquence vidéo, par exemple sa taille, sa couleur, ou même sa présence. Les informations enfouies dans le flux vidéo compressé via le procédé selon l'invention peuvent être indépendantes d'une image à l'autre. Les caractéristiques d'une image donnée au sein du flux vidéo peuvent être prises en compte pour déterminer le type d'information à enfouir. Par exemple, une signature permettant la vérification de l'intégrité du contenu sera préférablement enfouie dans une image fixe encodée dans son intégralité, dite image "intra", plutôt que dans une image prédite à partir d'une autre.

REVENDICATIONS

1. Procédé d'enfouissement d'une séquence binaire (17) dans une séquence vidéo ou un flux vidéo compressé (10), ledit flux pouvant être décomposé en plusieurs types d'objets, le procédé s'appliquant sur au moins une image contenue dans ladite séquence vidéo caractérisé en ce qu'il comporte au moins les étapes suivantes :
    e) Analyser la séquence vidéo dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt (13),
    f) Générer une carte de tatouage (11) définissant l'ensemble des blocs de pixels de la séquence éligibles pour l'opération d'enfouissement, ainsi que deux coefficients C_1 et C_2 issus d'une transformée en fréquence pour chacun des dits blocs. Lesdits coefficients C_1 et C_2 sont tirés aléatoirement parmi l'ensemble des coefficients transformés.
    g) Pour l'image compressée courante, exclure de la carte de tatouage les blocs associés à la zone d'intérêt (13), ainsi que ceux dont les coefficients C_1 et C_2 définis par la carte de tatouage ne répondent pas au critère suivant :

        $abs(abs(C_1)-abs(C_2)) < S$ où S est un seuil prédéterminé et abs() la fonction valeur absolue d'un entier.

    h) Appliquer une fonction de tatouage numérique (18) sur chaque bloc disponible afin d'obtenir un flux compressé tatoué (19) par une séquence binaire (17) de la façon suivante :
        - Pour insérer un bit « 1 » de ladite séquence binaire (17),
            - Si $abs(C_1) > abs(C_2)$, on ne change rien
            - Si $abs(C_1) \leq abs(C_2)$, on calcule $\varepsilon = abs(C_2) - abs(C_1)$ et on modifie la valeur de C_1, $C_1 = C_1 + \varepsilon + 1$ si $C_1>0$, $C_1= C_1 - \varepsilon - 1$ sinon.
        - Pour insérer un bit « 0 » de ladite séquence binaire (17),
            - Si $abs(C_1) < abs(C_2)$, on ne change rien

- Si $abs(C_1) \geq abs(C_2)$, on calcule $\varepsilon = abs(C_1) - abs(C_2)$ et on modifie la valeur de C_2, $C_2 = C_2 + \varepsilon + 1$ si $C_2>0$, $C_2= C_2 - \varepsilon - 1$ sinon.
-

2. Procédé selon la revendication 1, caractérisé en ce que la séquence binaire (17) est un condensé de l'image obtenu via l'étape suivante :
   - Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt (13) et appliquer au résultat une fonction de hachage visuel (14) générant en sortie un condensé de l'image (15)

3. Procédé selon la revendication 2 caractérisé en ce que le condensé de l'image (15) est crypté via une fonction de cryptage (16)

4. Procédé selon la revendication 2 caractérisé en ce que le flux compressé tatoué (19) est traité afin de vérifier l'intégrité de la séquence vidéo selon les étapes suivantes :
   - Analyser le flux compressé tatoué (19) dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt (13),
   - Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt (13) et appliquer au résultat une fonction de hachage visuel (14) générant en sortie un condensé de l'image (15),
   - Récupérer le condensé tatoué (21) dans le flux compressé tatoué (19) à partir de la carte de tatouage (11)
   - Effectuer une comparaison dudit condensé tatoué (21) avec le condensé (15)

5. Procédé selon la revendication 3 caractérisé en ce que le flux compressé tatoué (19) est traité afin de vérifier l'intégrité de la séquence vidéo selon les étapes suivantes :
   - Analyser le flux compressé tatoué (19) dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt (13),
   - Concaténer les coefficients issus de la transformée en fréquence des blocs appartenant aux dits objets ou groupes d'objets obtenus définissant une zone d'intérêt (13) et appliquer au résultat une fonction de hachage visuel (14) générant en sortie un condensé de l'image (15),
   - Appliquer, une fonction de cryptage (16) au condensé précédemment obtenu afin d'obtenir un condensé crypté (17),
   - Récupérer le condensé tatoué (21) dans le flux compressé tatoué (19) à partir de la carte de tatouage (11)
   - Effectuer une comparaison dudit condensé tatoué (21) avec le condensé crypté (17)

6. Procédé selon les revendications 3 ou 5 caractérisé en ce que la fonction de cryptage (16) met en œuvre un algorithme de chiffrement asymétrique ou un algorithme de chiffrement AES (Advanced Encryption Standard).

7. Procédé selon l'une des revendications précédentes caractérisé en ce que la fonction de hachage visuel (14) met en œuvre l'algorithme SHA-1 et que le condensé crypté (17) ait une taille de 160 bits.

8. Procédé selon la revendication 1 caractérisé en ce que la séquence binaire (17) est un message comportant une indication sur les caractéristiques des zones d'intérêt (13) et est fourni par l'étape d'analyse dans le domaine compressé (12).

9. Procédé selon l'une des revendications précédentes caractérisé en ce que la séquence vidéo est produite par un standard MPEG ou un standard ITU.

10. Dispositif pour tatouer numériquement au moins une partie d'un flux vidéo ou d'une séquence vidéo compressée comportant un émetteur (30) et un récepteur (40) caractérisé en ce que :
    - ledit émetteur (30) comporte au moins les éléments suivants adaptés à exécuter les étapes du procédé selon l'une des revendications 1 à 3 et 6 à 9 : un module d'analyse dans le domaine compressé (31), un module de hachage visuel (32), un module de tatouage numérique (33) et un module de transmission (34) du flux tatoué,
    - ledit récepteur (40) comporte au moins les éléments suivants adaptés à exécuter les étapes du procédé selon l'une des revendications 4 à 9 : un module de transmission (41), un module d'analyse (42), un module de hachage visuel (43), un module de validation d'intégrité (44).

ABREGE

PROCEDE ET DISPOSITIF POUR L'ENFOUISSEMENT D'UNE SEQUENCE BINAIRE DANS UN FLUX VIDEO COMPRESSE

Procédé et dispositif pour l'enfouissement d'une séquence binaire (17) dans une séquence vidéo ou un flux vidéo compressé (10), ledit flux pouvant être décomposé en plusieurs types d'objets, le procédé s'appliquant sur au moins une image contenue dans ladite séquence vidéo caractérisé en ce qu'il comporte au moins les étapes suivantes :

i) Analyser la séquence vidéo dans le domaine compressé afin de définir pour une image compressée donnée au moins un premier type d'objets ou groupe d'objets à traiter définissant une zone d'intérêt (13),

j) Générer une carte de tatouage (11) définissant l'ensemble des blocs de pixels de la séquence éligibles pour l'opération d'enfouissement, ainsi que deux coefficients C_1 et C_2 issus d'une transformée en fréquence pour chacun des dits blocs. Lesdits coefficients C_1 et C_2 sont tirés aléatoirement parmi l'ensemble des coefficients obtenus par application d'une transformée en fréquence sur un bloc de pixels donné.

k) Pour l'image compressée courante, exclure de la carte de tatouage les blocs associés à la zone d'intérêt (13), ainsi que ceux dont les coefficients C_1 et C_2 définis par la carte de tatouage ne répondent pas au critère suivant :

   $abs(abs(C_1)-abs(C_2)) < S$ où S est un seuil prédéterminé et abs() la fonction valeur absolue d'un entier.

l) Appliquer une fonction de tatouage numérique (18) sur chaque bloc disponible afin d'obtenir un flux compressé tatoué (19) par une séquence binaire (17) de la façon suivante :

   - Pour insérer un bit « 1 » de ladite séquence binaire (17),
     - Si $abs(C_1) > abs(C_2)$, on ne change rien
     - Si $abs(C_1) \leq abs(C_2)$, on calcule $\varepsilon = abs(C_2) - abs(C_1)$ et on modifie la valeur de C_1, $C_1 = C_1 + \varepsilon + 1$ si $C_1>0$, $C_1= C_1 - \varepsilon - 1$ sinon.

- Pour insérer un bit « 0 » de ladite séquence binaire (17),
  - Si $abs(C_1) < abs(C_2)$, on ne change rien
  - - Si $abs(C_1) \geq abs(C_2)$, on calcule $\varepsilon = abs(C_1) - abs(C_2)$ et on modifie la valeur de C_2, $C_2 = C_2 + \varepsilon + 1$ si $C_2>0$, $C_2= C_2 - \varepsilon - 1$ sinon.
  -

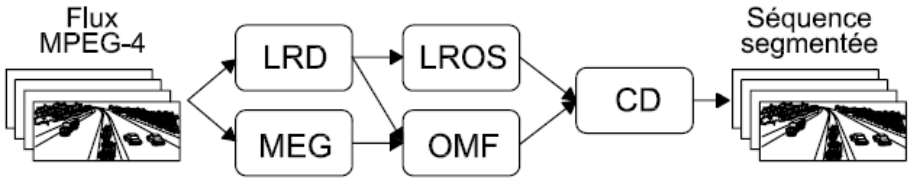


FIG.1

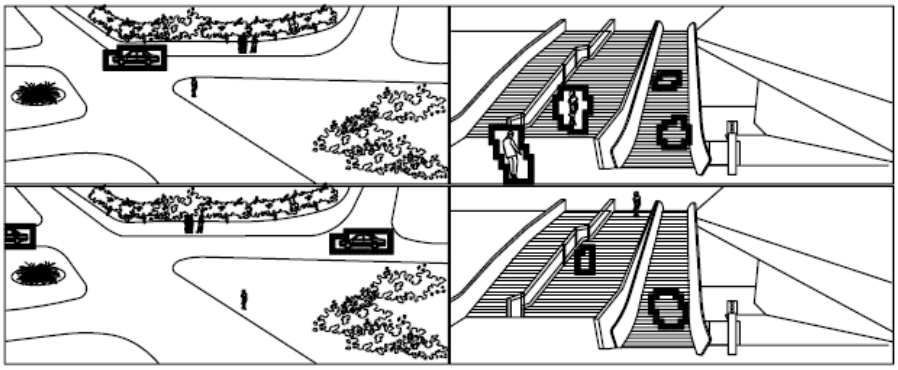


FIG.2

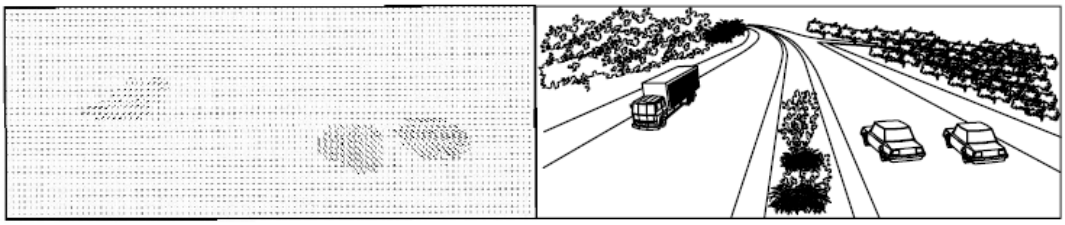


FIG.3

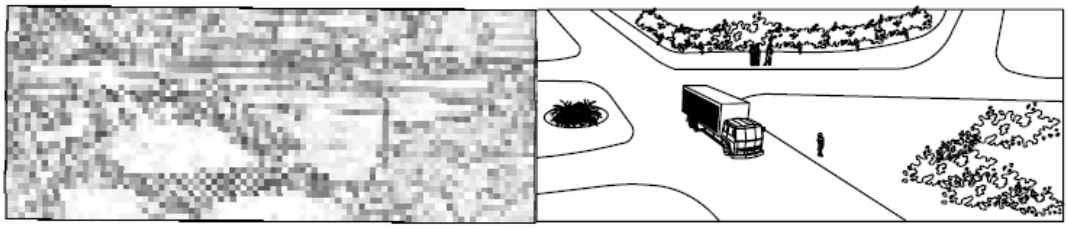


FIG.4

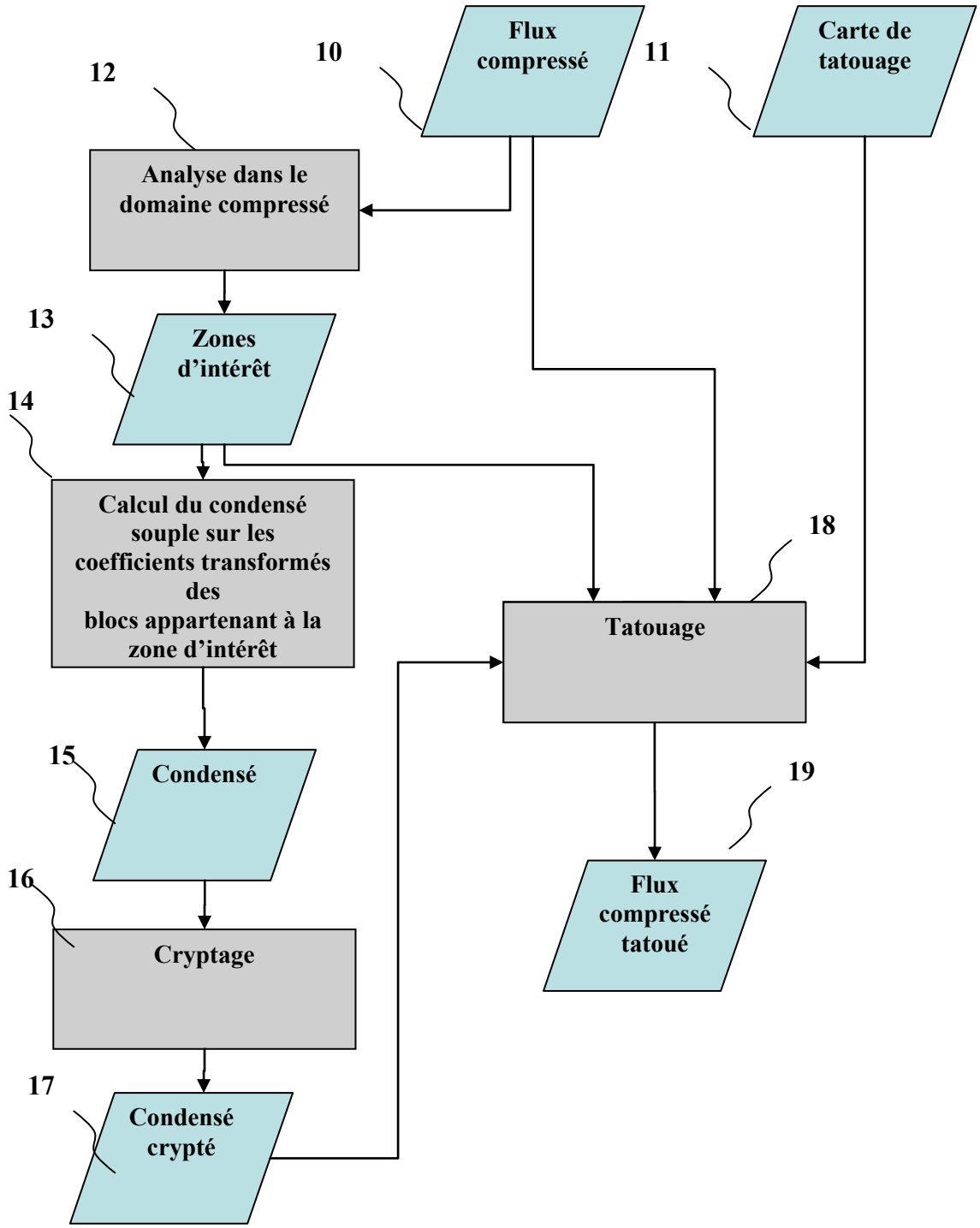

10
Flux compressé
11
Carte de tatouage
12
Analyse dans le domaine compressé
13
Zones d'intérêt
14
Calcul du condensé souple sur les coefficients transformés des blocs appartenant à la zone d'intérêt
15
Condensé
16
Cryptage
17
Condensé crypté
18
Tatouage
19
Flux compressé tatoué

FIG.5

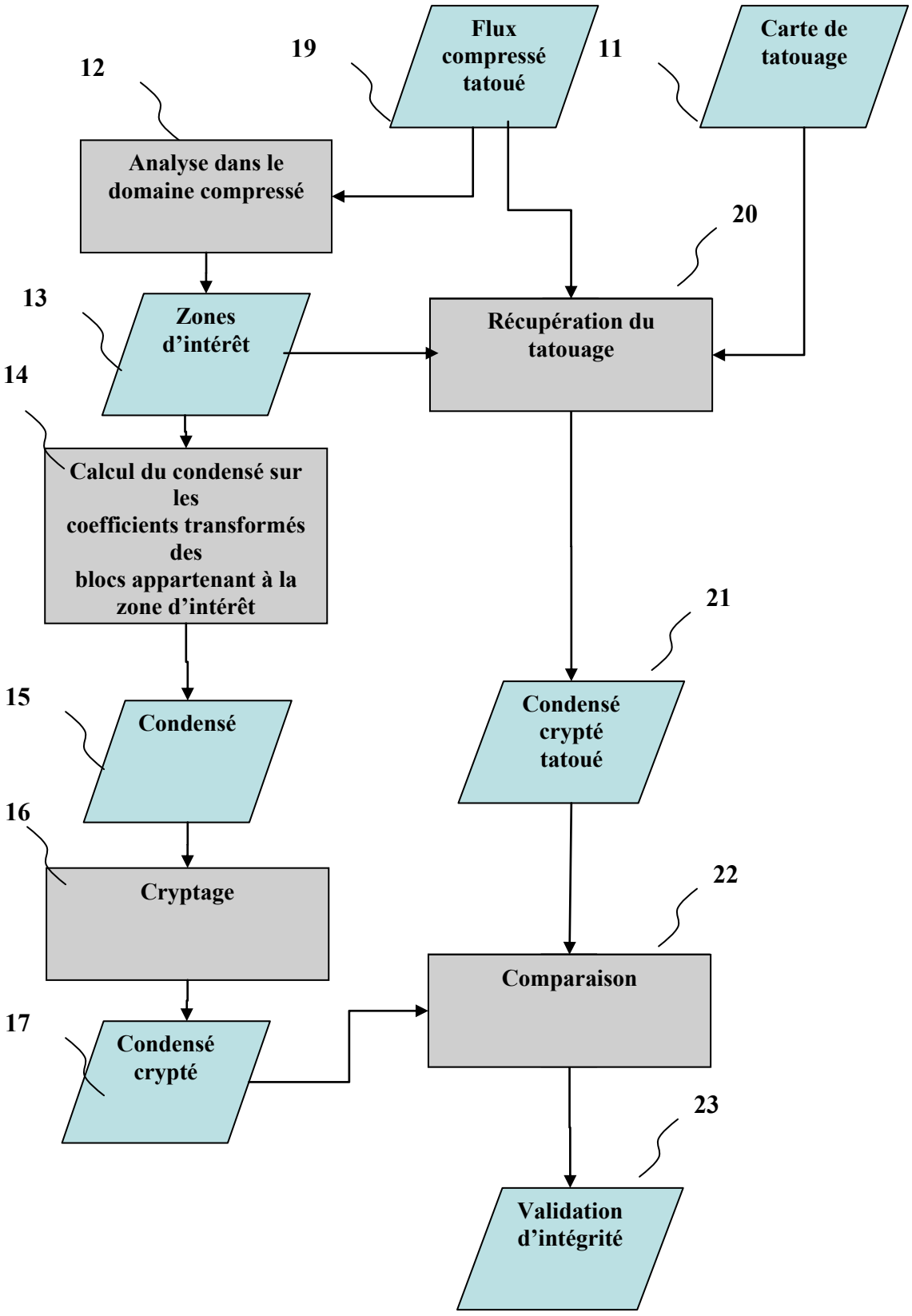

Flux compressé tatoué
19
11
Carte de tatouage
12
Analyse dans le domaine compressé
20
13
Zones d'intérêt
Récupération du tatouage
14
Calcul du condensé sur les coefficients transformés des blocs appartenant à la zone d'intérêt
21
15
Condensé
Condensé crypté tatoué
16
Cryptage
22
Comparaison
17
Condensé crypté
23
Validation d'intégrité

FIG.6

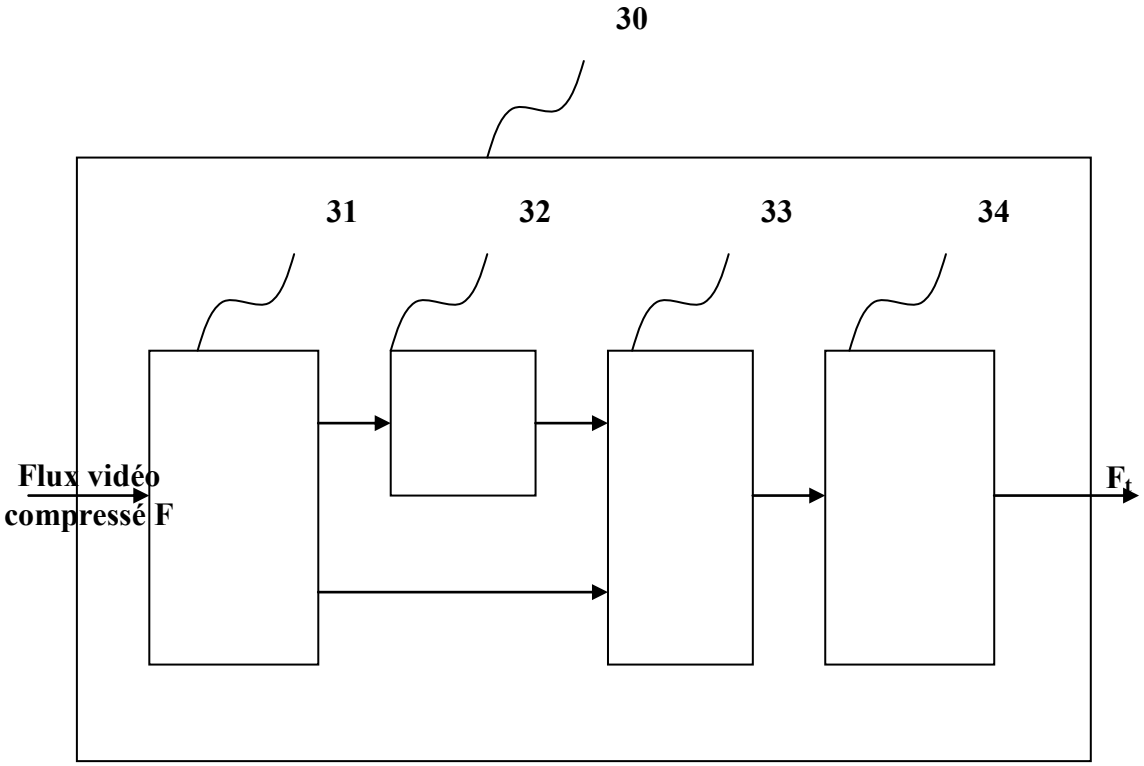


FIG.7

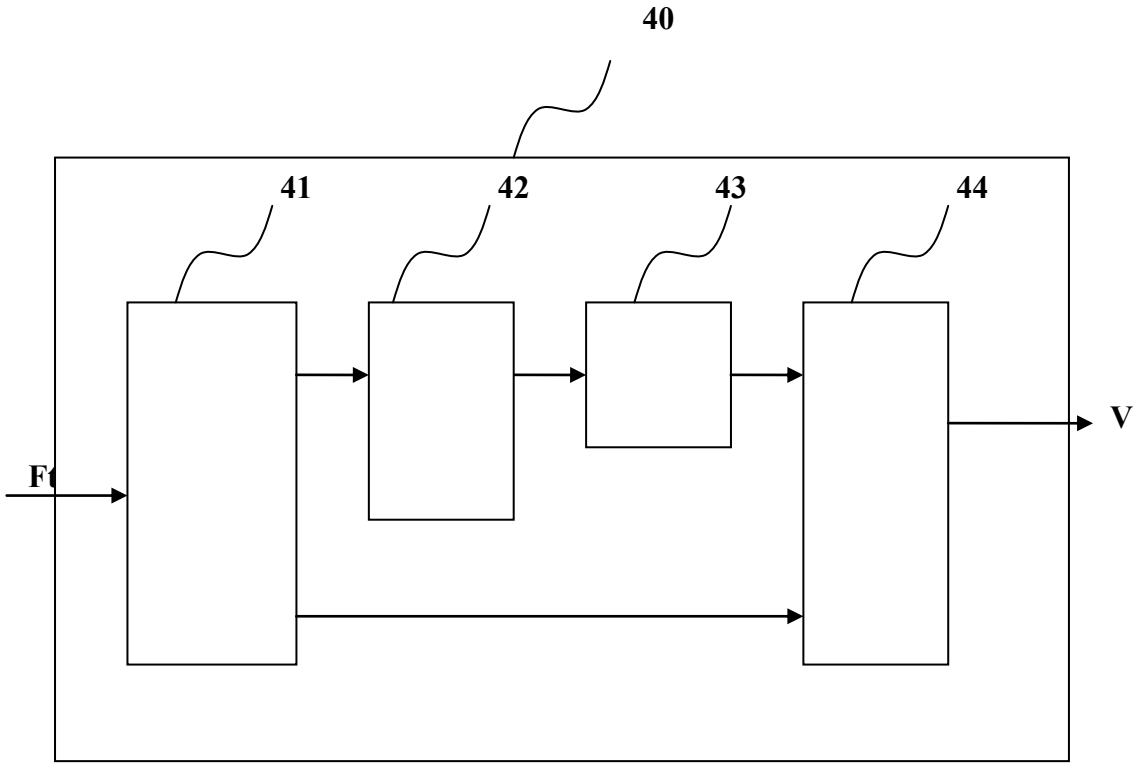


FIG.8

Liste des publications et brevets associées

Articles

[Leny et al., 2007] M. Leny, D. Nicholson, F. Prêteux, "De l'estimation de mouvement pour l'analyse temps réel de vidéos dans le domaine compressé", *21ème colloque du Groupe d'Etudes du Traitement du Signal et des Images (GRETSI 2007)*, pp. 1173-1176, Troyes, France, Septembre 2007.

[Leny et al., 2008a] M. Leny, F. Prêteux, D. Nicholson, "Statistical motion vector analysis for object tracking in compressed video streams", *SPIE (Society of Photographic Instrumentation Engineers) Electronic Imaging, Image Processing: Algorithms and Systems VI*, Volume 6812, pp. 68120Y.1-68120Y.12, San Jose, USA, Janvier 2008.

[Leny et al., 2008b] M. Leny, F. Prêteux, C. Le Barz, "Motion Aware Slicing for H.264 Selective Visual Encryption", *Conference on Design and Architectures for Signal and Image Processing (DASIP 2008)*, Bruxelles, Belgique, Novembre 2008.

[Leny et al., 2009] M. Leny, C. Le Barz, D. Nicholson, F. Prêteux, "A Fast Vehicle Retrieval Demonstrator Based on Compressed Video Stream Analysis", *Seventh International Workshop on Content-Based Multimedia Indexing (CBMI 2009)*, Chania, Crête, Juin 2009.

Chapitre

INFOM@GIC, projet structurant du pôle de compétitivité Cap Digital, 2011
D. Marraud, M. Leny, S. Herbin, M. Campedel : Chapitre 8 : Urbanview – Indexation sémantique de vidéos de surveillance urbaine pour l'aide à l'investigation

Brevets

[Leny et al., 2008c] FR 08.03053, 03/06/08 – M. Leny, C. Le Barz, D. Nicholson et F. Prêteux : "Procédé d'optimisation de l'analyse d'un flux vidéo par prétraitement dans le domaine compressé et système mettant en oeuvre le procédé".

[Leny et al., 2008d] FR 08.03061, 03/06/08 – M. Leny, C. Le Barz et E. Renan : "Procédé et système permettant de protéger dès la compression la confidentialité des données d'un flux vidéo lors de sa transmission".

[Leny et al., 2008e] FR 08.03065, 03/06/08 – M. Leny, C. Le Barz, E. Renan et F. Prêteux : "Construction d'une nouvelle structure de représentation de l'image s'appuyant sur des objets sémantiques au sein d'un flux vidéo compressé, procédé et dispositif associés".

[Le Barz et al., 2008a] FR 08.03063, 03/06/08 – C. Le Barz, M. Leny et E. Renan : "Procédé et système permettant de crypter visuellement les objets mobiles au sein d'un flux vidéo compressé".

[Le Barz et al., 2008b] FR 08.03060, 03/06/08 – C. Le Barz, M. Leny et D. Nicholson : "Procédé et système permettant de protéger un flux vidéo en cours de compression contre les erreurs survenant lors d'une transmission".

[Le Barz et al., 2008c] FR 08.03064, 03/06/08 – C. Le Barz, M. Leny et D. Nicholson :"Procédé et système permettant de protéger un flux vidéo compressé contre les erreurs survenant lors d'une transmission".

[Le Barz et al., 2008d] FR 08.03052, 03/06/08 – C. Le Barz, M. Leny et D. Nicholson :" Système de vidéosurveillance intelligent reconfigurable dynamiquement".

[Le Barz et al., 2008e] FR 08.03054, 03/06/08 – C. Le Barz, M. Leny et D. Nicholson :" Procédé d'adaptation du débit de transmission de flux vidéo par prétraitement dans le domaine compressé et système mettant en oeuvre le procédé".

[Leny et al., 2008f] FR 08.06837, 05/12/08 – M. Leny, C. Le barz : "Procédé et dispositif pour l'enfouissement d'une séquence binaire dans un flux vidéo compressé ".

[Leny, 2008] FR 08.07065, 16/12/08 – M. Leny : "Gant intégrant un stylet pour interaction avec une interface tactile".

Bibliographie

[AES, 2001] Advanced Encryption Standard, *National Institute of Standards and Technology*, FIPS-197, November 2001.

[Alexiadis et Sergiadis, 2008] D.S. Alexiadis, G.D. Sergiadis, "Estimation of Multiple, Time-Varying Motions Using Time-Frequency Representations and Moving-Objects Segmentation", *IEEE Transactions on Image Processing*, Volume 17, Issue 6, pp. 982-990, Juin 2008.

[Arulampalam et al., 2002] S. Arulampalam, S. Maskell, N. Gordon et T. Clapp, “Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking”, *IEEE Transactions on Signal Processing, Voume 50, No. 2*, Février 2002.

[Asefi et Dabbagh, 2006] M. Asefi , M.-Y. Dabbagh, "Adaptive Video Motion Estimation Algorithm via Estimation of Motion Length Distribution and Bayesian Classification", *IEEE International Symposium on Signal Processing and Information Technology*, pp. 807-810, Août 2006.

[Aventura] Aventura Technologies, H.264 SVC IP Box Camera, 640 x 480, Real-time 30FPS, http://www.aventuratechnologies.com/products/product_detail.asp?clProdID=992.

[Babu et al., 2004] R.V. Babu, K.R. Ramakrishnan, S.H. Srinivasan, "Video object segmentation: a compressed domain approach", *IEEE Transactions on Circuits and Systems for Video Technology*, Volume 14, Issue 4, pp. 462-474, Avril 2004.

[Babu et Ramakrishnan, 2002] R.V. Babu, K.R. Ramakrishnan, "Compressed domain motion segmentation for video object extraction", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2002)*, Volume 4, pp. 3788-3791, Mai 2002.

[Battiato et al., 2007] S. Battiato, G. Gallo, G. Puglisi, S. Scellato, "SIFT Features Tracking for Video Stabilization", *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, pp. 825-830, Septembre 2007.

[Bergeron et Lamy-Bergot, 2005] C. Bergeron, C. Lamy-Bergot, "Compliant selective encryption for H.264/AVC video streams", *Proc. of the International Workshop on Multimedia Processing (MMSP`05)*, pp. 477-480, Shanghai, Chine, Octobre-Novembre 2005.

[Bevilacqua et Azzari , 2006] A. Bevilacqua, P. Azzari, "High-Quality Real Time Motion Detection Using PTZ Cameras", *IEEE International Conference on Video and Signal Based Surveillance (AVSS 2006)*, pp. 23-27, Novembre 2006.

[Bo et al., 2006] L. Bo, C. Qimei, G. Fan, "Freeway Auto-surveillance From Traffic Video", *6th International Conference on ITS Telecommunications*, pp. 167-170, 2006.

[Boxin et al., 2008] S. Boxin, L. Lin; X. Chao, "Comparison between JPEG2000 and H.264 for digital cinema", *IEEE International Conference on Multimedia and Expo (ICME 2008)*, 2008.

[Christopoulos et al., 2000] C. Christopoulos, A. Skodras, et T. Ebrahimi, "The JPEG2000 still image coding system: an overview", *IEEE Transactions on Consumer Electronics*, Vol. 46, No. 4, pp. 1103-1127, Novembre 2000.

[Coimbra et Davies, 2004] M. Coimbra, and M. Davies, "Segmentation of moving pedestrians within the compressed domain", *IEEE International Conference on Acoustics, Speech, and Signal Processing 04*, Montréal, Canada, 2004.

[Desurmont et al., 2005] X. Desurmont, B. Lienard, J.Meessen and J.F. Delaigle. Real-time optimizations for integrated smart network camera. *Conf. on Real-Time Imaging IX, part of the IS&T SPIE Symposium on Electronic Imaging*, Janvier 2005.

[Ebrahimi et Horne, 2000] T. Ebrahimi, C. Horne, "MPEG-4 Natural Video Coding – An Overview", *Signal Processing*, VOL. 15, No 4, pp 365 385, 2000.

[Eng et Ma, 2000] H.-L. Eng; K.-K. Ma, "Spatiotemporal segmentation of moving video objects over MPEG compressed domain", *IEEE International Conference on Multimedia and Expo (ICME 2000)*, Volume 3, pp. 1531-1534, Juillet 2000.

[Ewerth et al., 2007] R. Ewerth, M. Schwalb, P. Tessmann, B. Freisleben, "Segmenting Moving Objects in MPEG Videos in the Presence of Camera Motion", *14th International Conference on Image Analysis and Processing (ICIAP 2007),* pp. 819-824, Septembre 2007.

[Fan et al., 2008] X. Fan, L. Xu, X. Zhang, L. Chen, "The Research and Application of Human Detection Based on Support Vector Machine Using in Intelligent Video Surveillance System", *Fourth International Conference on Natural Computation. (ICNC 2008)*, Volume 2, pp. 139-143, Octobre 2008.

[Flierl et Girod , 2004] M. Flierl, B. Girod, " Video Coding with Superimposed Motion-Compensated Signals: Applications to H.264 and Beyond", *Springer*, 2004.

[Frank 2004] A. Frank, "On Kuhn's Hungarian Method – A tribute from Hungary", *Technical reports, TR-2004-14. Published by the Egrervary Research Group, Pazmany P. setany 1/C, H–1117*, Budapest, Hungary, 2004.

[Goldman, 2005] M. Goldman, "A Comparison of MPEG-2 Video, MPEG-4 AVC, and SMPTE VC-1 (Windows Media 9 Video)", Tandberg Television, http://video.ldc.lu.se/pict/WM9V-MP4AVC-MP2V_comparison-Goldman.pdf, 2005.

[Haque et al., 2008] M. Haque, M. Murshed, M. Paul, "On Stable Dynamic Background Generation Technique Using Gaussian Mixture Models for Robust Object Detection", *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance (AVSS 2008)*, pp. 41-48, Septembre 2008.

[Haritaoglu et al., 1998] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: A real-time system for detecting and tracking people in 2D", *European Conference on Computer Vision*, Freiburg, Germany, 1998.

[He et Liou., 1997] Z.L. He et M.L. Liou, "A high performance fast search algorithm for block matching motion estimation", *IEEE Trans. on Circuits and Systems for Video Technology*, vol.7, no.5, pp.826-8, Octobre 1997.

[Hoi-Ming et al., 2005] W. Hoi-Ming, O.C. Au, H. Chi-Wang et Y. Shu-Kei, "Enhanced predictive motion vector field adaptive search technique (E-PMVFAST)-based on future MV prediction", *IEEE International Conference on Multimedia and Expo, ICME 2005*, Juillet 2005.

[Hsieh et al., 2008] C.C. Hsieh, W.R. Lai, A. Chiang, "A Real Time Spatial/Temporal/Motion Integrated Surveillance System in Compressed Domain", Eighth International Conference on Intelligent Systems Design and Applications (ISDA 2008.), Volume 3, pp. 658-665, Novembre 2008.

[Huang et al., 2009] T. Huang, J. Qiu, T. Sakayori, S. Goto, T. Ikenaga, "Motion Detection Based on Background Modeling and Performance Analysis for Outdoor Surveillance", *International Conference on Computer Modeling and Simulation (ICCMS 2009)*, pp. 38-42, Février 2009.

[ISO/TC 223/AH 3, 2009] ISO/TC 223/AH 3, "Societal Security - Videosurveillance format for interoperability", Document en cours de rédaction par l'AFNOR, 2009.

[Ji et Park, 1999] S. Ji, H.W. Park, "Region-based video segmentation using DCT coefficients", *International Conference on Image Processing (ICIP 99)*, Volume 2, pp. 150-154, Octobre 1999.

[JO, 2007a] Journal Officiel de la République Française, MINISTÈRE DE L'INTÉRIEUR, DE L'OUTRE-MER ET DES COLLECTIVITÉS TERRITORIALES, "Arrêté du 3 août 2007 portant définition des normes techniques des systèmes de vidéosurveillance", NOR : IOCD0762353A, 21 août 2007.

[JO, 2007b] Journal Officiel de la République Française, MINISTÈRE DE L'INTÉRIEUR, DE L'OUTRE-MER ET DES COLLECTIVITÉS TERRITORIALES, "Arrêté du 3 août 2007 portant définition des normes techniques des systèmes de vidéosurveillance (rectificatif)", NOR : IOCD0762353Z, 25 août 2007.

[Jodoin et al., 2006] P.-M. Jodoin, M. Mignotte, J. Konrad, "Light and Fast Statistical Motion Detection Method Based on Ergodic Model", *IEEE International Conference on Image Processing (ICIP 2006)*, pp. 1053-1056, Octobre 2006.

[Kas et Nicolas, 2009] C. Kas, H. Nicolas, "H.264/SVC scene motion analysis", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pp. 957-960, Avril 2009.

[Kas et al., 2009] C. Kas, M. Brulin, H. Nicolas, C. Maillet, "Compressed domain aided analysis of traffic surveillance videos", *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC 2009)*, pp. 1-8, Août 2009.

[Kim, 2007], Y. M. Kim, "Object Tracking in a Video Sequence", *Stanford Project Report,* 2007.

[Koenen, 2003] R. Koenen, "MPEG-4 Demystified", *Apple Worldwide Developers Conference*, http://www.m4if.org/resources/MPEG-4-WWDC.pdf , Juin 2003.

[Kompatsiaris et al., 2000]I. Kompatsiaris, G. Mantzaras, M.G. Strintzis, "Spatiotemporal segmentation and tracking of objects in color image sequences", *IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, Volume 5, pp. 29-32, May 2000.

[Lan et al., 2006] D. Lan, I. Zoghlami, S.C. Schwartz, "Object Tracking in Compressed Video with Confidence Measures", *IEEE International Conference on Multimedia and Expo (ICME 2006)*, pp. 753-756, Juillet 2006.

[Lay et Guan, 1999] J.A. Lay, L. Guan, "Image retrieval based on energy histograms of the low frequency DCT coefficients", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1999)*, Volume 6, pp. 3009-3012, Mars 1999.

[Lee, 2004] D.M. Lee, "Television Technical Theory Unplugged", Version 5.0, http://www.danalee.ca/ttt/index.htm, 2004.

[Li et al., 1994] R. Li, B. Zeng, et M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438-42, Août 1994.

[Ma et Hosur, 2000] K.K. Ma et P.I. Hosur, "Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST)," *ISO/IEC JTC1/SC29/WG11 MPEG99/m5851*, Noordwijkerhout, NL, Mars 2000.

[Manerba et al., 2008] F. Manerba, J. Benois-Pineau, R. Leonardi, B. Mansencal, "Multiple Moving Object Detection for Fast Video Content Description in Compressed Domain", *EURASIP Journal on Advances in Signal Processing*, Volume 2008, No. 1, pp. 1-15, 2008.

[Mehaoua, 2006] A. Mehaoua, "Normes de Compression audio-vidéo", Support de cours Université de Paris 5, 2006.

[Meng et Li, 2009] M. Lei et L. Hang , "Motion Estimation Algorithm Based on Motion Characteristics", *WASE International Conference on Information Engineering*, pp. 91-94 , Juillet 2009.

[MPEG-2 RS] Reference software MPEG-2 ISO/IEC 13818-5.

[NGSIM] Next Generation SIMulation (NGSIM) - http://www.webs1.uidaho.edu/ngsim Corpus Peachtree : http://www.webs1.uidaho.edu/ngsim/ATL/ATL.htm

[Ouaret et al., 2006], M. Ouaret, F. Dufaux, T. Ebrahimi, "On Comparing JPEG 2000 and Intraframe AVC", SPIE *Applications of Digital Image Processing XXIX*, 2006.

[Pearson et Gill, 2005] G. Pearson et M. Gill "An Evaluation of Motion JPEG 2000 for Video Archiving", *Proceedings Archiving 2005, IS & T*, pp. 237-243, Avril 2005.

[Pereira et Ebrahimi, 2002] F. Pereira, T. Ebrahimi, "The MPEG-4 book", IMSC Press Multimedia Series, Prentice Hall PTR, 2002.

[Pereira et Nunes, 2001] F. Pereira and P. Nunes, "Levels for MPEG-4 Visual Profiles", MPEG Industry Forum, http://www.m4if.org/resources/profiles/index.php, 2001.

[PETS] *Eleventh IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, http://www.pets2009.net .

[Piciarelli et Foresti, 2007] C. Piciarelli, G.L. Foresti, "Anomalous trajectory detection using support vector machines", ", *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSBS 2007)*, pp. 153-158, Septembre 2007.

[Rong et al., 2002] S. Rong , L. Xiaofeng, L. Zaiming, "Efficient spatiotemporal segmentation and video object generation for highway surveillance video", *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions*, Volume 1, pp. 580-584, Juin 2002.

[Rovio, 2008] Robot de surveillance grand public de WowWee, commercialisé en 2008, http://www.wowwee.com/en/products/tech/telepresence/rovio:rovio.

[Santiago-Mozos et al., 2003] R. Santiago-Mozos, J.M. Leiva-Murillo, F. Perez-Cruz, A. Artes-Rodriguez, "Supervised-PCA and SVM classifiers for object detection in infrared images", *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSBS 2003)*, pp. 122-127, Juillet 2003.

[Schwarz et al., 2007] H. Schwarz, D. Marpe, et T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", *IEEE Transactions on circuits and systems for video technology*, Vol.17, No.9, Septembre 2007.

[Shneier et Mottaleb, 1996] M. Shneier and M. A. Mottaleb, "Exploiting the JPEG compression scheme for image retrieval", *IEEE Trans on Pattern Analysis and Machine Intelligence*, Volume 18, No. 8, pp. 849-853, Août 1996.

[Shen et Sethi, 1996] B. Shen, I. Sethi, “Direct feature extraction from compressed images”, *SPIE Storage and Retrieval for Image and Databases IV 2670*, 1996.

[Sheng et al., 2009] H. Sheng, C. Li, Q. Wei, Z. Xiong, "An Approach to Motion Vehicle Detection in Complex Factors over Highway Surveillance Video", *International Joint*

Conference on Computational Sciences and Optimization (CSO 2009), Volume 1, pp. 520-523, Avril 2009.

[Sofka, 2008] M. Sofka, "Commentary Paper 2 on "On Stable Dynamic Background Generation Technique Using Gaussian Mixture Models for Robust Object Detection"", *IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance (AVSS 2008)*, pp. 41-48, Septembre 2008.

[Spykee, 2007] Robot ludique de Meccano, commercialisé en septembre 2007, http://www.spykeeworld.com/spykee/FR/index.html.

[Stauffer et Grimson, 1998] C. Stauffer et W.E.L Grimson, "Adaptive background mixture models for real-time tracking", *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[Topiwala et al., 2006] P. TOPIWALA, T. TRAN, D. WEI, "Performance comparison of JPEG2000 and H.264/AVC high profile intra-frame coding on HD video sequences", *Applications of digital image processing XXIX*, Août 2006.

[Tourapis et al., 2000] A. M. Tourapis, O.C. Au, M.L. Liou, G. Shen, et I. Ahmad, "Optimizing the Mpeg-4 Encoder – Advanced Diamond Zonal Search", *IEEE International Symposium on Circuits and Systems*, Geneva, Suisse, Mai 2000.

[Toyama et al., 1999] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance", *IEEE International Conference on Computer Vision*, Corfu, Grèce, 1999.

[TRECVID] TREC Video Retrieval Evaluation, Text REtrieval Conference, http://www-nlpir.nist.gov/projects/trecvid .

[Verdant et al., 2007] A. Verdant, P. Villart, A. Dupret, H. Mathias, "Détection de mouvement adaptative bas niveau", *GRETSI*, Volume 1, pp. 341-344, Troyes, France, Septembre 2007

[Wang et al., 2007] Y. Wang; J.G. Kim; S.F. Chang; H.M. Kim, "Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding", *IEEE Transactions on Multimedia*, Volume 9, Issue 2, pp. 213–220, Février 2007.

[Watkinson, 2004] J. Watkinson, " The MPEG handbook: MPEG-1, MPEG-2, MPEG-4", *Edition 2, Focal Press*, 2004.

[Weerakkody et al., 2007] W. Weerakkody, W.A.C. Fernando, A.B.B. Adikari, "Unidirectional Distributed Video Coding for Low Cost Video Encoding", IEEE *Transactions on Consumer Electronics*, Volume 53 Issue: 2, pp 788-795, Mai 2007.

[WG11, 2002a] MPEG Working Group 11, " Overview of the MPEG-4 Standard", http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm, Mars 2002.

[WG11, 2002b] MPEG Working Group 11, "MPEG-4 – The Media Standard", http://www.m4if.org/public/documents/vault/m4-out-20027.pdf, Novembre 2002.

[Wiegand et al., 2003] T.Wiegand, G. J. Sullivan, G. Bjøntegaard, et A. Luthra, "Overview of the H.264/AVC Video Coding Standard", *IEEE Transactions on circuits and systems for video technology*, Vol.13, No.7, pp. 560-576, Juillet 2003.

[Wiegand et al., 2008] T. Wiegand, H. Schwarz, "Multi-frame Motion-Compensated Prediction", Fraunhofer Heinrich Hertz Institute, http://www.hhi.fraunhofer.de/en/departments/image-processing/image-communication/multi-frame-motion-compensated-prediction, 2008.

[Wren et al., 1997] C. Wren, et al., “Pfinder: Real-time tracking of the human body”, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 780-785, 1997.

[Yokoyama et al., 2009] T. Yokoyama, T. Iwasaki, T. Watanabe, "Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain", *Seventh International Workshop on Content-Based Multimedia Indexing (CBMI 2009)*, pp. 201-206, Juin 2009.

[Zhang et al., 1995] H. Zhang, C.Y. Low et S.W. Smoliar, "Video Parsing and Browsing Using Compressed Data", *Multimedia Tools and Applications*, Volume 1, No. 1, pp. 89-111, 1995.

[Zhang et Chen, 2007] J. Zhang; C.H. Chen, "Moving Objects Detection and Segmentation In Dynamic Video Backgrounds", *IEEE Conference on Technologies for Homeland Security (THS 2007)*, pp. 64-69, Mai 2007.

[Zhou et al., 2003] X. Zhou, R. Collins, T. Kanade, "A master-slave system to acquire biometric imagery of humans at distance", *ACM International Workshop on Video Surveillance*, 2003.

Acronymes

4CIF	Résolution quadruple du CIF : 704x576.
AC	Emploi de l'acronyme de « *alternative current* », ou courant alternatif, utilisé en image pour désigner les coefficients d'un bloc transformé, sauf le premier (voir DC).
AES	*Advanced Encryption Standard*. Standard de cryptage avancé.
AFF	*Adaptative Field Coding*. Basculement automatique d'images progressives vers entrelacées et inversement (selon les contenus par exemple).
AVC	*Advanced Video Coding*. Standard de compression vidéo également appelé MPEG-4 Part 10 AVC ou H.264.
B	Pour *Bidirectionnelle*. Image du flux vidéo qui nécessite deux images références (une avant et une après) pour être décodée.
CABAC	*Context-adaptive binary arithmetic coding*. Codage entropique proposé dans H.264.
CAVLC	*Context-adaptive variable-length coding*. Codage entropique proposé dans H.264.
Cb	Coefficient de chrominance différentielle bleue, pour le format YCbCr.
CIF	*Common Intermediate Format*. Taille standardisée de séquences YCbCr à 352x288 pixels.
Cr	Coefficient de chrominance différentielle rouge, pour le format YCbCr.
DC	Emploi de l'acronyme de *direct current*, ou courant continu, utilisé en image pour désigner le premier coefficient d'un bloc transformé, correspondant à la valeur moyenne du signal sur le bloc considéré.
DCT	*Discrete Cosine Transform*, ou transformé en cosinus discrète est une transformation proche de la transformée de Fourier discrète (DFT).
FMO	*Flexible Macrobloc Ordering*. Méthode de partitionnement d'une image.
Full HD	*Full High Definition*. Format d'image standardise, entre autre pour la télévision, comprenant 1080 ligne (1920 colonnes en 16/9^{e} , 1440 en 4/3).
GMC	*Global Motion Compensation*. Technique améliorant la compression lors de travellings.
GME	*Global Motion Estimation.* Estimation du mouvement global de la caméra, utilisé pour le compenser lors de l'analyse.
GMM	*Gaussian Mixture Model*. Outil de modélisation statistique par plaquage de gaussienne(s) sur une distribution.
GoP ou GOP	*Group of Pictures*. Portion du flux vidéo entre deux images Intra qui peut être décodée indépendamment du reste de la séquence.
HD	*High Definition*. Format d'image comptant au moins 720 lignes.
I	Pour *Intra*. Image du flux vidéo codée intrinsèquement.
IDR	*Instantaneous Decoder Refresh.* Image de H.264 SVC
IEC	*International Electrotechnical Commission.* Organisation internationale non gouvernementale de standardisation.
ISO	*International Standardisation Organisation*. Organisation internationale non gouvernementale de standardisation.
J2K	Abréviation alternative pour JPEG 2000.
JPEG	*Joint Photographic Experts Group*. Comité conjoint à ISO/IEC JTC1 et ITU-T à l'origine des standards JPEG et JPEG-2000.
MAD	*Mean Absolute Difference*. Méthode de calcul d'erreur.
MBA	*MacroBlock Allocation Map*. Table d'allocation des macroblocs pour la création

Map	de slices de formes arbitraires (en H.264).
MPEG	*Moving Picture Experts Group*. Ou ISO/IEC JTC1/SC29 WG11. Comité de standardisation audio et video créé en 1988 par l'ISO.
MSE	*Mean Square Error*. Méthode de calcul d'erreur.
MV	*Motion Vector*. Abréviation anglaise de vecteur d'estimation de mouvement.
NAL	*Network Abstratcion Layer*. Couche d'abstraction réseau.
NTSC	*National Television System Committee*. Standard de codage video analogique d'initiative états-unienne en 1953.
P	Pour *Prédite*. Image du flux vidéo qui nécessite une image référence la précédant pour être décodée.
PPS	*Picture Parameters Set*. Déclaration des paramètres de l'image.
PTZ	*Pan, Tilt, Zoom*. Type de caméra de surveillance motorisée avec une mobilité verticale (*Tilt*), horizontale (*zoom*) et une fonction de zoom.
QCIF	*Quarter CIF*. CIF à une résolution ¼, soit 176x144.
RLE	*Run-Lengh Encoding*, ou codage par plage. Algorithme de compression de données.
RoI ou ROI	*Region of Interest*. Région d'intérêt.
SAD	*Sum of Absolute Differences*. Méthode de calcul d'erreur.
SD	*Standard Definition*. Format d'image standardisé entre autre pour la télévision, d'une résolution de 720x576 (voire 704x525 pour des formats NTSC), de 25 à 30 images par seconde.
SHA	*SecureHash Algorithm*. Algorithme de hachage sécurisé. SHA-1 est une fonction de hachage 160 bits développée par la *National Security Agency* (NSA).
SNR	*Signal Noise Ratio*. Rapport signal sur bruit.
SSE	*Sum of Square Error*. Méthode de calcul d'erreur.
STANAG	*STANdardisation Agreement*. Accords de normalisation édités par l'OTAN.
SVC	*Scalable Video Coding*. Extension G de MPEG-4 Part 10 AVC / H.264.
SVM	*Support Vector Machine*. Techniques d'apprentissage supervisé permettant de résoudre des problèmes de discrimination et de régression.
VLC	*Variable Length Coding*. Codage par agrégation des coefficients de même valeur.
VS	Vidéosurveillance.
Y	Coefficient lié à la luminance, entre autre pour le format YCbCr.

FORMATION DOCTORALE

Dans le cadre de l'Ecole Doctorale S&I
Télécom & Management SudParis en co-accréditation avec
l'Université d'Evry–Val d'Essonne

Résumé

Le développement de réseaux de vidéosurveillance, civils ou militaires, pose des défis scientifiques et technologiques en termes d'analyse et de reconnaissance des contenus des flux compressés. Dans ce contexte, les contributions de cette thèse portent sur :

- une méthode de segmentation automatique des objets mobiles (piétons, véhicules, animaux …) dans le domaine compressé,
- la prise en compte des différents standards de compression les plus couramment utilisés en surveillance (MPEG-2, MPEG-4 Part 2 et MPEG-4 Part 10 / H.264 AVC),
- une chaîne de traitement multi-flux optimisée depuis la segmentation des objets jusqu'à leur suivi et description.

Le démonstrateur réalisé a permis d'évaluer les performances des approches méthodologiques développées dans le cadre d'un outil d'aide à l'investigation, identifiant les véhicules répondant à un signalement dans des bases de données de plusieurs dizaines d'heures. En outre, appliqué à des corpus représentatifs des différentes situations de vidéosurveillance (stations de métro, carrefours, surveillance de zones en milieu rural ou de frontières ...), le système a permis d'obtenir les résultats suivants :

- analyse de 14 flux MPEG-2, 8 flux MPEG-4 Part 2 ou 3 flux AVC en temps réel sur un cœur à 2.66 GHZ (vidéo 720x576, 25 images par seconde),
- taux de détection des véhicules de 100% sur la durée des séquences de surveillance de trafic, avec un taux de détection image par image proche des 95%,
- segmentation de chaque objet sur 80 à 150% de sa surface (sous ou sur-segmentation liée au domaine compressé).

Ces recherches ont fait l'objet du dépôt de 9 brevets liés à des nouveaux services et applications rendus opérationnels grâce aux approches mises en œuvre. Citons entre autres des outils pour la protection inégale aux erreurs, la cryptographie visuelle, la vérification d'intégrité par tatouage ou l'enfouissement par stéganographie.

Summary

The increasing deployment of civil and military videosurveillance networks brings both scientific and technological challenges regarding analysis and content recognition over compressed streams. In this context, the contributions of this thesis focus on:

- an autonomous method to segment in the compressed domain mobile objects (pedestrians, vehicles, animals …),
- the coverage of the various compression standards commonly used in surveillance (MPEG-2, MPEG-4 Part 2, MPEG-4 Part 10 / H.264 AVC),
- an optimised multi-stream processing chain from the objects segmentation up to their tracking and description.

The developed demonstrator made it possible to bench the performances of the methodological approaches chosen for a tool dedicated to help investigations. It identifies vehicles from a witness description in databases of tens of hours of video. Moreover, while dealing with corpus covering the different kind of content expected from surveillance (subway stations, crossroads, areas in countryside or border surveillance …), the system provided the following results:

- simultaneous real time analysis of up to 14 MPEG-2 streams, 8 MPEG-4 Part 2 streams or 3 AVC streams on a single core (2.66 GHz; 720x576 video, 25 fps),
- 100% vehicles detected over the length of traffic surveillance footages, with a image per image detection near 95%,
- a segmentation spreading over 80 to 150% of the object area (under or over-segmentation linked with the compressed domain).

These researches led to 9 patents linked with new services and applications that were made possible thanks to the suggested approaches. Among these lie tools for Unequal Error Protection, Visual Cryptography, Watermarking or Steganography.