Stéganographie en domaine vidéo compressé

Clément Chastagnol

▶ **To cite this version:**

Clément Chastagnol. Stéganographie en domaine vidéo compressé. Traitement du signal et de l'image [eess.SP]. 2009. dumas-00517769

HAL Id: dumas-00517769

https://dumas.ccsd.cnrs.fr/dumas-00517769v1

Submitted on 15 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École Centrale de Lyon
TFE 2009

Thales Communications
Colombes

Rapport Final de Travail de Fin d'Études

Stéganographie en domaine vidéo compressé

Clément Chastagnol

Tuteurs :	Option : Informatique et Communications
ECL : Ardabilian, Mohsen	Filière : Communications et Multimédia
Entreprise : Leny, Marc	Métier : Ingénieur Recherche et Développement

Remerciements

Je remercie Marc Leny, doctorant du laboratoire MMP et tuteur de ce stage, pour son aide, sa disponibilité et sa compétence. La confiance et l'autonomie qu'il m'a accordées ont été une source de motivation pour mon travail.

Je tiens également à remercier Erwann pour m'avoir fait partager sa connaissance profonde du format H.264, ainsi que François pour sa relecture attentive de ce rapport, ses remarques pertinentes et sa disponibilité.

Merci à Cyril et Didier pour leurs idées et les discussions qui en ont découlé.

Enfin, merci à Sébastien, Marc, Hervé, Salah ainsi qu'à tous les autres stagiaires, prestataires et titulaires du laboratoire pour leur bonne humeur et la gentillesse de leur accueil.

Résumé du rapport :

Ce document décrit les étapes de développement d'un système stéganographique complet, permettant de cacher des données dans un flux vidéo compressé. Le système s'appuie sur un schéma d'enfouissement hybride, travaillant à la fois sur les résultats de la prédiction inter et intra-image du format de compression H.264/AVC. Il est intégré au sein d'un ensemble codeur-décodeur H.264 propriétaire et réalise de meilleures performances que les techniques issues de l'état de l'art selon plusieurs critères. Un *framework* de stéganalyse a été également spécifiquement conçu et développé pour évaluer le système décrit. Des approches d'apprentissage supervisé, basées sur des concept d'exploration de données ont pour cela été utilisées.

Ce travail a été effectué dans le cadre du projet UrbanView, lui-même faisant partie du projet collaboratif Infom@gic. L'objectif du projet UrbanView est de développer une chaîne complète de traitement multimédia mettant en place des analyses de mouvement dans le domaine compressé pour des applications de vidéo-surveillance.

Mots-clés libres :

Dissimulation de données ; stéganographie ; compression vidéo ; standard H.264/AVC ; stéganalyses ; extraction de connaissances ; statistiques.

Abstract :

This document describes the development of a complete steganographic system, hiding data in a compressed video stream. The system is based on an hybrid scheme, embedding data in both intra and inter predicted frames of the video stream, using the H.264/AVC coding standard. It is fully integrated in a proprietary codec and can achieve better performances than the state-of-the-art methods regarding various criteria. The described system has been evaluated with a steganalysis benchmark specifically designed and implemented during this internship using a supervized learning approach based on data mining concepts.

This work in part of a larger project named UrbanView, inside the Infom@gic collaborative project. Its objective is to realize a full multimedia data processing chain performing compressed-domain motion analysis, for video surveillance applications.

Keywords :

Data hiding ; steganography ; video compression ; H.264/AVC ; steganalysis ; data mining ; statistics.

Table des matières

Table des figures

Liste des tableaux

Introduction

Ce rapport présente le travail effectué au cours du stage de fin d'études dans le cadre de la scolarité à l'École Centrale de Lyon (ECL) d'une part et à l'Institut National des Sciences Appliquées de Lyon (INSA Lyon) d'autre part. Le stage s'est déroulé au sein du laboratoire MMP de la société Thales Communications, sur le site de Colombes (92), du 7 avril au 30 septembre. Il a été encadré par Marc Leny, doctorant en traitement du signal. Le laboratoire MMP travaille sur l'analyse automatique de vidéos dans le domaine compressé, appliquée notamment à la détection et classification d'activité en vidéosurveillance. Les outils actuels permettent de détecter les objets mobiles présents un flux vidéo, d'effectuer des requêtes sur la présence d'un type d'objet (véhicule/piéton, couleur, etc.) et de générer des alertes. Dans ce cadre, l'objectif du stage était d'appréhender les problématiques de remontée automatique d'alarme en concevant et en implémentant un prototype logiciel d'enfouissement de données dans un flux vidéo compressé de manière peu ou pas visible et en préservant l'intégrité de régions d'intérêts, selon un brevet déposé par le laboratoire.

Ce document se veut la synthèse de la tâche accomplie au cours du stage. Il décrit les principales étapes de la réalisation du projet, en précise l'état d'avancement et explique la méthodologie employée. L'accent a notamment été mis sur les améliorations apportées à l'état de l'art. Après une brève description du groupe Thales et du laboratoire d'accueil, le standard de compression H.264/AVC ainsi que les problématiques de dissimulation de données sont présentés. Les différentes étapes de réalisation d'un système stéganographique complet sont ensuite exposées, depuis les spécifications jusqu'à la validation expérimentale, en passant par la conception et l'implémentation. Quelques perspectives d'évolution sont ensuite discutées, le stage s'achevant dans un mois à l'heure où ces lignes sont écrites.

1 Contexte du stage

Cette partie présente tout d'abord le groupe Thales puis le laboratoire MMP qui a accueilli le stage. Le projet dans lequel il s'inscrit est ensuite détaillé, ainsi que le besoin qui l'a justifié.

1.1 Présentation de l'entreprise d'accueil

1.1.1 Le groupe Thales

Thales est un groupe international d'électronique et de systèmes mettant à profit les technologies duales au service des marchés de la défense, de l'aéronautique et de la sécurité. Il conçoit, développe et réalise des solutions de très haute technologie qui répondent aux besoins de sécurité, de communication et d'information de pays du monde entier. Ses activités couvrent trois pôles d'activités : l'Aéronautique, la Défense et les Technologies de l'Information et des Services (IT&S).

Thales est présent dans plus de 50 pays à travers le monde dont 65% de ses implantations se situent en Europe, et emploie près de 68 000 personnes en 2008 (contre 65 000 en 2002). Le chiffre d'affaire a atteint en 2003, pour l'ensemble des trois pôles d'activités plus de 12 milliards d'euros.

Le groupe est organisé en six divisions : *Aerospace*, *Air Systems*, *Land & Joint Systems*, *Naval*, *Security & Services* et *Space*.

1.1.2 Thales Communications et le laboratoire MMP

Thales Communications est l'un des cinq *Business Group* du pôle Défense. C'est un acteur majeur des grands programmes de défense dans le domaine des communications et du commandement. La société développe également son activité sur les marchés civils dans les télécommunications, l'aviation civile, le transport et la sécurité. Par ailleurs, elle contribue à de nombreux projets de recherche en télécommunications.

Thales Communications rassemble les activités du groupe dans les communications de défense. L'ensemble compte aujourd'hui environ 9000 personnes dans différentes sociétés à travers le monde. En France, Thales Communications comprend différents établissements en région parisienne (Colombes, Gennevilliers, Massy) et en province (Cholet, Laval, Brive, Marq-en-Baroeul) ainsi que deux filiales basées en France (Safare, Gerac), pour un effectif de plus de 5 000 employés.

Thales Communications est divisé en cinq *Business Lines* (BLs) :

- Systèmes Terre & Interarmées ;
- Réseaux Satcom & Sécurité ;
- Communications tactiques ;
- Communications Air & Naval ;
- Services Ssupport Clients.

Les BLs sont en charge, au plan mondial, des segments de marché, de produits ou de services. Elles sont composées de *Technical Business Units* au niveau de chaque pays. Leur organisation permet de distinguer systèmes, équipements et services. Le stage s'est effectué au sein de la TBU *Embedded Digital Systems*, située dans la branche des Opérations.

Au sein de Thales Communications, les fonctions relatives au développement des sous-ensembles cartes électroniques numériques sont confiées à la TBU EDS. Celle-ci centralise la majorité des activités connexes (traitement du signal, algorithmes des formes d'ondes,

FPGA) et est en charge au sein de TCF de la maîtrise des technologies clés de son domaine de compétence, qui couvre les segments suivants :

- Matériel numérique ;
- Systèmes sur Silicium (FPGA, ASIC) ;
- Atelier Matériel ;
- Traitement du Signal et Multimédia ;
- Logiciels propres aux formes d'ondes et protocoles pour radiocommunications.

L'unité emploie environ 300 personnes et se compose de quatre services :

- le service Logiciel et Architecture (SWA - 80 personnes) ;
- le service Traitement du signal et Multimédia (SPM - 75 personnes) ;
- le service Matériel Numérique (DHD - 70 personnes) ;
- le service Composants programmables (SIE - 30 personnes).

Le service SPM a en charge le développement des algorithmes de Traitement du Signal & Multimédia, la production des logiciels associés et le développement de produits pour l'ensemble des secteurs suivants :

Radiocommunications : codage canal et modem dans toutes les gammes de fréquence (de la VLF à l'EHF) et pour tous les débits utiles (de 50 b/s en VLF à plusieurs Mb/s en satellite) ;

Traitement capteurs : traitement d'antennes de type goniométrie, filtrage d'antennes, formation de faisceaux, antennes intelligentes ; traitement de guerre électronique de type détection ;

Traitement du signal multimédia (parole et image) : synthèse et reconnaissance vocale, compression de type vocodeur, compression d'images et vidéos, gestion de la qualité de service associé ;

Support TS : mise en place et entretien de l'atelier de production des logiciels de Traitement de Signal.

Le laboratoire *MultiMedia Processing* (MMP), entité au sein de laquelle le stage a été effectué, a pour principale activité la compression du signal audio à bas et très bas débit, le *watermarking* et le codage des images (JPEG 2000, H.264/MPEG 4-AVC). Il est composé de 11 ingénieurs Thales, 3 prestataires et 1 thésard ; parmi ces 15 personnes, 9 travaillent sur des problématiques image ou vidéo et 6 sur des problématiques audios.

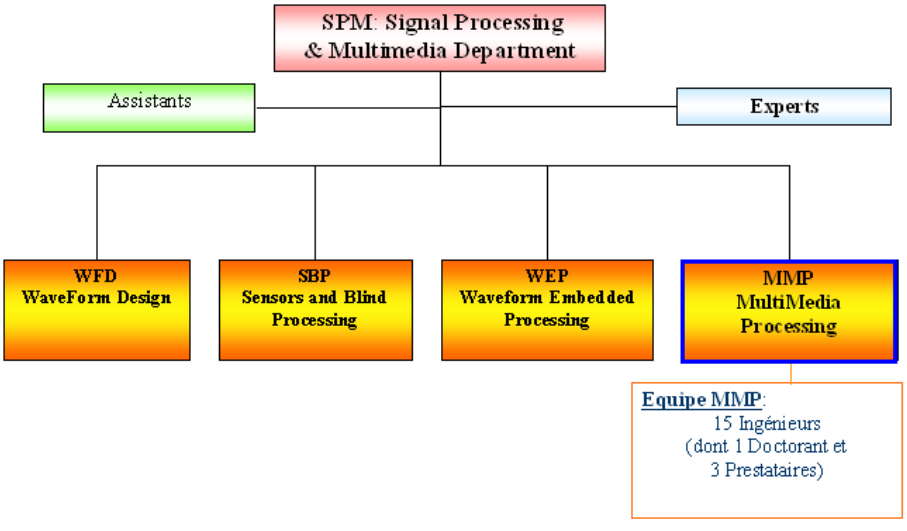

FIGURE 1 – Organigramme du service SPM (source : Thales).

1.2 Le projet Infom@gic

Le stage effectué s'inscrit au sein d'Infom@gic. Il s'agit d'un projet collaboratif régional coordonné par la société Thales et regroupant plus de 20 partenaires, aussi bien privés (EADS, Pertimm, Xerox, Arisem, TEMIS...) que publics (CEA, INA, Laboratoires de recherche d'universités françaises...) et soutenu par le pôle de compétitivité francilien Cap Digital « Image, Multimédia et Vie Numérique » (IMVN), un groupement de 30 grandes entreprises et plus de 200 PME centrées sur les nouveaux marchés des contenus numériques. Il vise à sélectionner, tester, intégrer et valider des méthodes opérationnelles dans le domaine de la recherche, de l'indexation et de l'extraction de connaissances ainsi que de la fusion d'informations multimédia (image, son, texte et données structurées), pour des domaines d'application aussi variés la finance, la santé, le numérique, la défense ou la sécurité, avec l'ambition de capter 20% du marché européen et 10% du marché mondial et de développer l'emploi et la formation dans les NTAI (Nouvelles Technologies de l'Analyse de l'Information). Le projet a démarré en 2006 et s'achève en 2009.

Dans ce cadre, le laboratoire MMP a contribué à l'un des sous projets, UrbanView, qui avait pour but de proposer un démonstrateur de recherche par requête sur un corpus de vidéosurveillance en milieu urbain. Dirigée par EADS, cette section comprenait également TELECOM ParisTech et l'ONERA. Le travail du laboratoire s'est centré sur l'analyse de vidéos de circulation routière, permettant le suivi et l'indexation automatique de véhicules au travers de leurs trajectoire, couleurs, dimensions... Le cœur du système développé consiste en un outil d'analyse de mouvement en temps réel dans le domaine vidéo compressé : il est capable de détecter les zones ou objets en mouvement dans le flux vidéo et de les distinguer de l'arrière-plan, qui est fixe dans le cas de la vidéosurveillance. Cette analyse est effectuée directement après la compression vidéo : le système utilise l'estimation de mouvement calculée par l'encodeur ainsi qu'un apprentissage statistique pour segmenter dans la frame courante les objets mouvants. Cette approche, en rupture avec les algorithmes d'estimation de mouvement qui travaillent traditionnellement au niveau pixelique, permet de réduire significativement les temps de calculs (d'un facteur 50 environ) et ouvre la voie au traitement du flux en temps-réel en haute définition ou sur des systèmes embarqués. Des rapports d'analyse sont générés au niveau des caméras équipées du système et doivent être transmis ainsi que d'autres informations à un serveur central pour un traitement plus fin (partie déléguée à EADS dans le démonstrateur final).

Deux besoins apparaissent alors :

- un canal de communication synchronisé avec la vidéo pour transmettre les méta-données calculées par le système d'analyse de mouvement ;
- une sécurisation des données, de manière à ce que celles-ci ne soient pas accessibles sans la « clé » adéquate.

L'idée d'enfouir les informations directement dans le flux vidéo selon les techniques de stéganographie présente alors des avantages certains, puisque le même canal servira au transport de la vidéo et des méta-données associées, et ceci de manière discrète et sécurisée. On aura donc automatiquement synchronisation des données et du flux auquel elles se rapportent. De plus, une gestion hiérarchisée des droits d'accès pourra être mise en place, le niveau d'accès le plus bas permettant simplement de lire la vidéo et le plus élevé de pouvoir lire les méta-données issues de l'analyse de mouvement.

Enfin, les techniques de dissimulation de données dans les médias numériques sont assez peu connues au sein de Thales. Tout en ayant une direction précise dans le cadre d'un projet existant, ce stage est donc l'occasion d'explorer ce domaine.

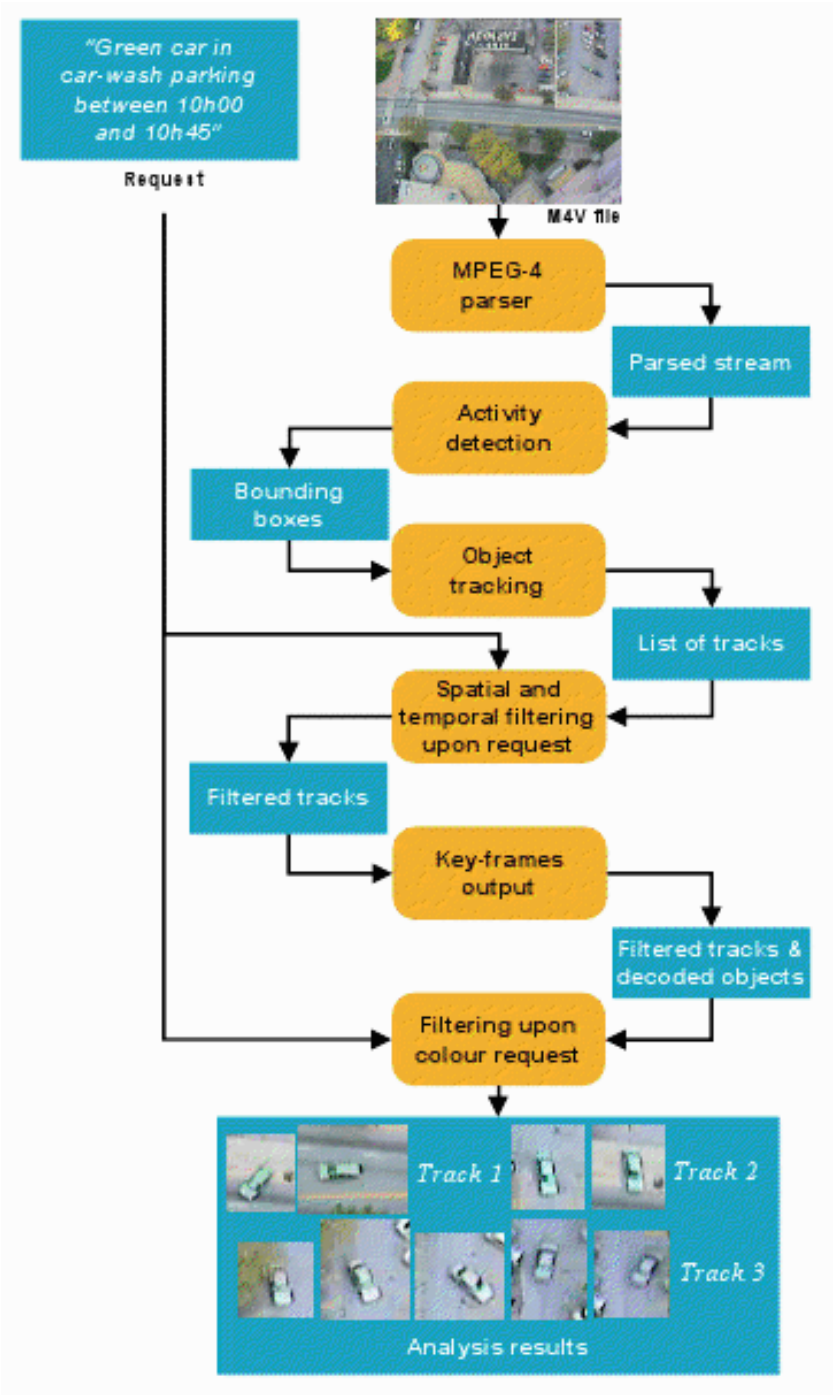

FIGURE 2 – Description de l'algorithme d'analyse de mouvement dans le domaine compressé (source : laboratoire MMP).

2 Objectifs et planification

Le but du stage est la conception d'un module de dissimulation de données relatives à des ROI détectées dans un flux vidéo compressé. Ce module doit venir s'insérer dans un système complet d'acquisition, de traitement et de transmission de vidéo pour des applications de vidéosurveillance.

2.1 Cahier des charges

2.1.1 Spécifications fonctionnelles

Les contraintes du système sont présentées ci-dessous par ordre de priorité.

Non-perturbation des ROI : on souhaite préserver les ROI avec le minimum de dégradations visuelles possible. Les éléments de syntaxe du flux compressé devront également être le moins perturbés possibles car l'analyse de mouvement se base sur l'estimation de mouvement du codeur.

Capacité d'enfouissement du signal-hôte : le schéma d'enfouissement de données choisi devra permettre la transmission des rapports en terme de volume. La charge (payload) représentée par les rapports et la capacité d'enfouissement du signal-hôte devront donc être évalués. Un coefficient de sécurité sera appliqué pour prévoir d'éventuelles augmentations de charge (ajout de redondance du à l'utilisation de codes correcteurs d'erreurs par exemple).

Qualité de la transmission : les informations doivent être transmises sans erreurs.

Complexité : le temps d'exécution des algorithmes d'enfouissement mis en place devra rester inférieur aux limites imposées par le fonctionnement temps-réel du système complet.

Indétectabilité : cette contrainte rejoint la contrainte de non-perturbation des ROI. Une indétectabilité à l'examen visuel est demandée, mais l'indétectabilité à un examen statistique poussé n'est pas exigée.

2.1.2 Quantification des objectifs et données techniques

Le système sera basé sur le format de compression vidéo H.264/AVC, imposé par la chaîne de traitement amont. La résolution de travail se situera entre le VGA et la SD, à 25 images/seconde. Les débits d'usage pour de la vidéosurveillance sont compris entre 200 Ko/s et 4 Mo/s. La quantité d'informations à enfouir a été évaluée à environ 2500 bits/seconde ; avec un coefficient de sécurité important [1], la capacité minimale est donc portée à 5000 bits/seconde, soit 200 bits/frame à 25 images/seconde.

Remarque En l'état, le système d'analyse génère 1 Go de données texte au format XML par heure. Cependant une grande partie de ces informations n'est pas critique (histogrammes de couleur complet des ROI pour chaque frame par exemple) et le format XML est particulièrement inefficace en terme de codage (pour une heure de vidéo, le fichier XML généré pèse 10 Mo après compression ZIP). Après sélection, codage efficace et compression, la charge utile se réduit finalement à environ 1 Mo par heure.

1. On a choisi un cœfficient de sécurité de 2 pour plusieurs raisons :
   - l'utilisation éventuelle de codes correcteurs d'erreurs qui gonflera la charge à enfouir,
   - la prévision des évolutions du système d'analyse, qui pourra générer plus de données.

2.2 Planning

Un diagramme GANTT a été établi pour planifier l'activité du stage et la suivre.

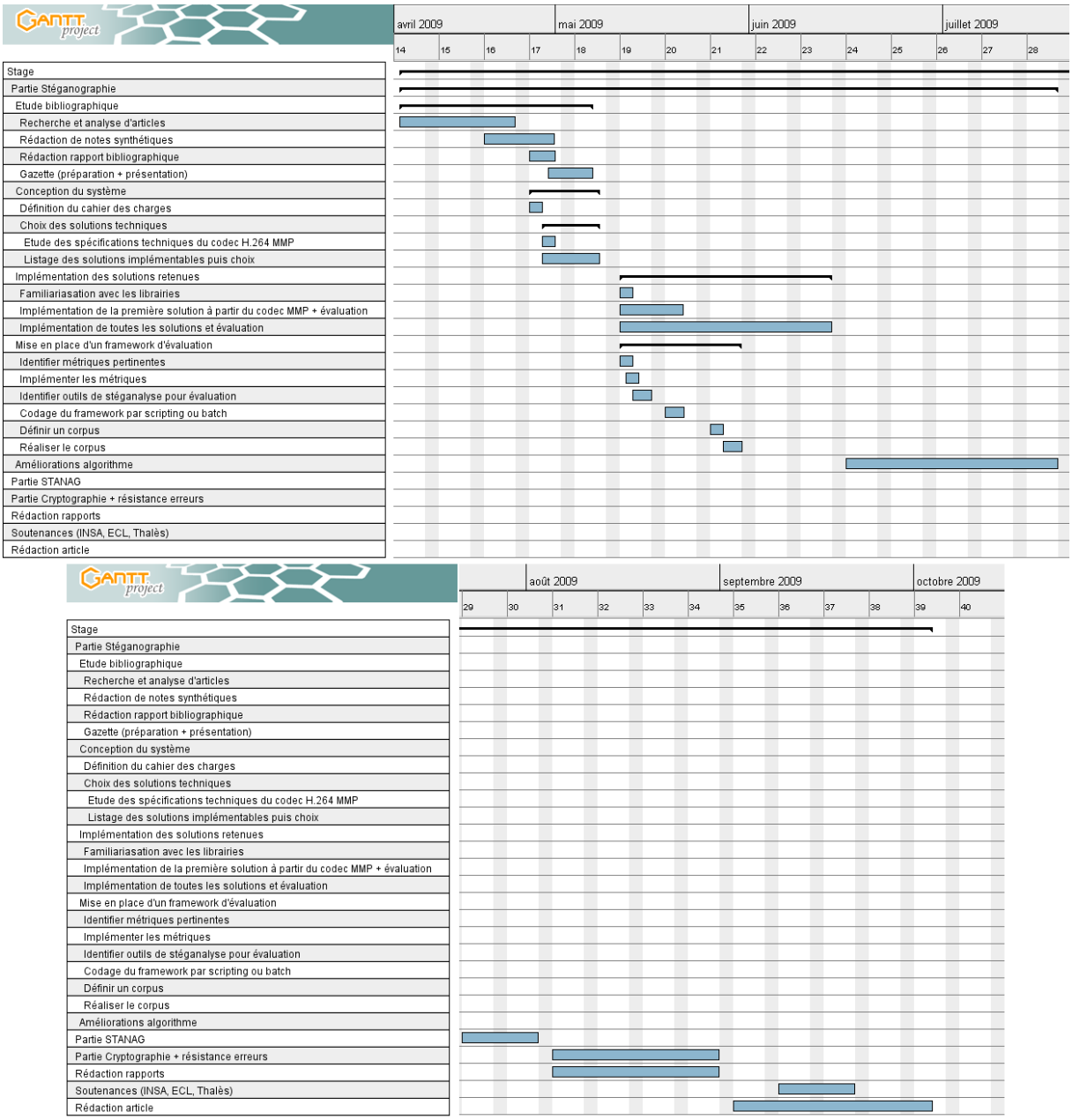

FIGURE 3 – Diagramme GANTT prévisionnel pour l'avancement du stage.

3 Éléments bibliographiques

Cette partie comprend tout d'abord une introduction au format de compression vidéo H.264/AVC, indispensable à la compréhension de la suite du rapport car au coeur du travail effectué au cours du stage. Les principes et enjeux de la stéganographie et de la stéganalyse sont ensuite présentés, ainsi que quelques techniques usuelles. De plus amples détails sont présents dans le rapport bibliographique [1].

3.1 Le standard H.264/AVC

On est resté ici relativement superficiel sur la présentation du standard H.264/AVC ; les détails techniques ont volontairement été omis pour des raisons de concision. Cependant, le lecteur ressentira certainement le besoin d'approfondir le sujet ; il est invité à se reporter à l'article [2] pour plus d'informations.

3.1.1 Présentation générale

La conception du nouveau standard de compression vidéo H.264/AVC a été initiée par la *Joint Video Team* (JVT), formée par le regroupement d'entités du *Video Coding Experts Group* (VCEG. ITU-T SG16 Q.6) et du *Moving Pictures Expert Group* (MPEG. ISO/IEC JTC1/SC29/WG11) en décembre 2001. Son objectif principal est le codage efficace et robuste de frames vidéo rectangulaires et leur transport sur un réseau ; l'efficacité de compression est doublée par rapport à toutes les autres normes vidéo existantes.

Le document « *Recommendation H.264 : Advanced Video Coding* », publié en 2003, ne définit pas un codec (association encodeur/décodeur) à proprement parler, mais plutôt une syntaxe pour le flux vidéo compressé et une méthode pour décoder cette syntaxe et produire une séquence vidéo affichable. De cette manière, tout décodeur se conformant aux recommandations du standard produira la même sortie, étant donné un flux encodé selon le standard. Seul le décodeur est donc véritablement standardisé, le document fondateur ne spécifiant pas comment encoder la vidéo, mais simplement quelle forme devra avoir le flux encodé. La méthode de codage est ainsi laissée au choix des développeurs de logiciels, permettant une grande flexibilité pour optimiser les implémentations pour des applications spécifiques, même si en pratique les modules du codeur sont souvent les symétriques de ceux du décodeur.

Le standard H.264/AVC regroupe une couche vidéo (*Video Coding Layer* — VCL), qui représente le contenu vidéo, et une couche réseau (*Network Abstraction Layer* — NAL), qui comprend les en-têtes appropriées pour le transport de l'information par des couches transports ou des supports de stockages particuliers. La conception du VCL suit l'approche de codage vidéo par blocs, comme dans toutes les normes antérieures (MPEG et ITU).

L'algorithme de base du codage-source est constitué d'une prédiction d'images (inter-picture prediction) pour exploiter la corrélation statistique temporelle, et d'une transformation orthogonale permettant la décorrélation des dépendances statistiques spatiales. Il n'y a en fait pas véritablement un seul nouvel élément de codage vidéo dans le VCL qui fournit à lui seul une amélioration importante de l'efficacité de compression par rapport aux normes vidéo antérieures. C'est plutôt la pluralité de différentes petites améliorations qui s'ajoutent jusqu'à un gain significatif.

3.1.2 Vue d'ensemble du codec

Un codeur H.264 va mettre en œuvre séquentiellement des traitement de base (prédiction, transformation, codage) pour produire un flux binaire compressé. Un décodeur H.264 va effectuer les opérations inverses (décodage, transformation inverse et reconstruction) pour produire une séquence vidéo décodée.

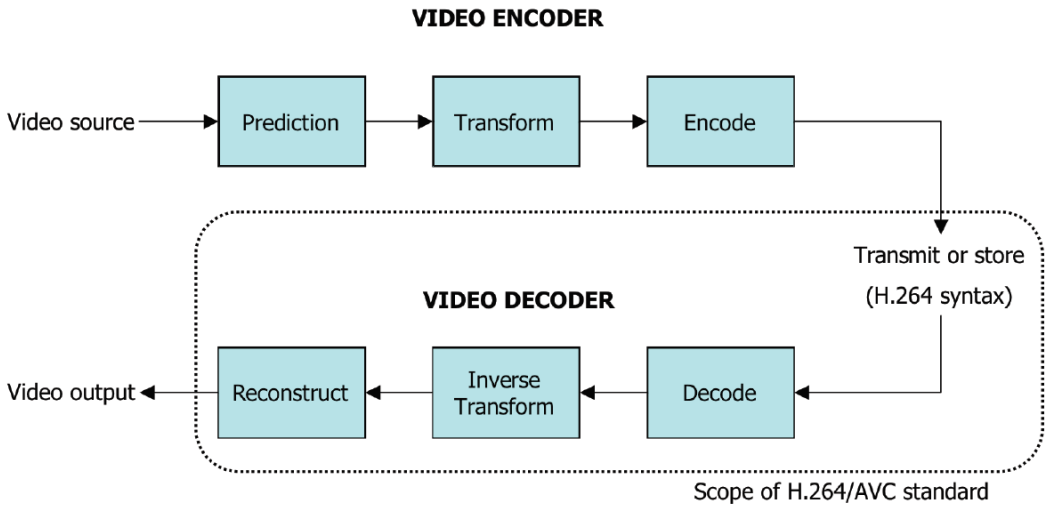

FIGURE 4 – Processus complet de codage et de décodage (source : [3]).

Processus de codage

Prédiction Une image d'une vidéo est subdivisée en macroblocs (blocs de 16×16 pixels). Le codeur calcule une prédiction du macrobloc courant d'après les données déjà codées provenant soit de la même frame (on parle alors de prédiction intra-image) ou d'autres frames déjà codées et transmises (prédiction inter-image ou d'estimation de mouvement). Le codeur soustrait ensuite la prédiction ainsi formée du macrobloc courant pour donner un résidu (on parle de compensation de mouvement lorsque la prédiction est inter-image).

Les méthodes de prédiction supportées par H.264 sont plus flexibles que dans les standards précédents, permettant une prédiction plus précise et une compression plus efficace. Par exemple, les tailles de blocs peuvent varier de 4×4 à 16×16 pixels pour la prédiction selon le niveau de détails local.

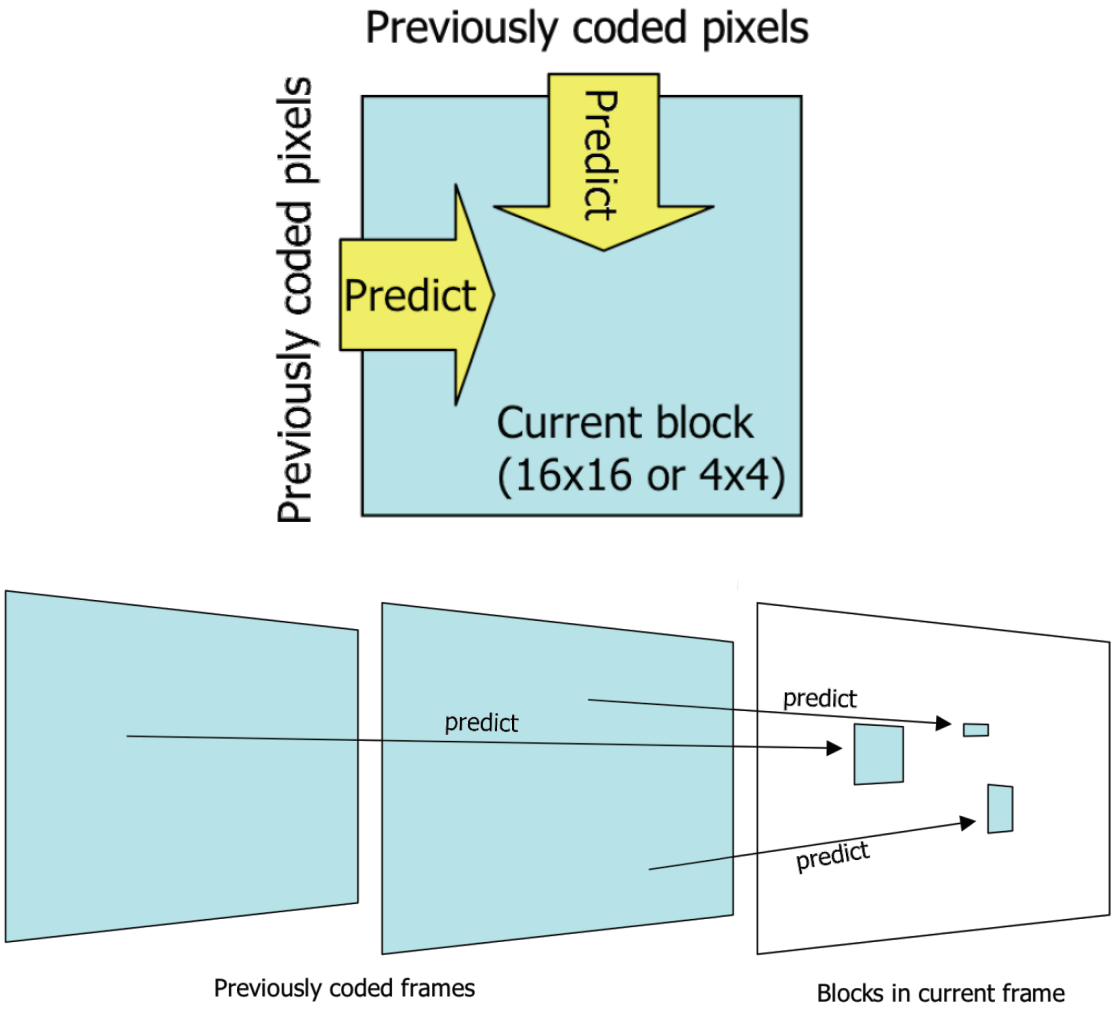

FIGURE 5 – Prédiction intra et inter image (source : [3]).

Transformation et quantification Un bloc de résidus est transformé par une transformée entière 4×4 ou 8×8, une forme inspirée de la transformée en cosinus discret (DCT), mais qui n'est pas soumise aux erreurs d'arrondis. On obtient un ensemble de cœfficients qui sont en fait des pondérations pour des motifs de base permettant de reconstruire les résidus par combinaison linéaire.

Le bloc de cœfficients issu de la transformation est ensuite quantifié : chaque coefficient est divisé par une valeur entière (appelée pas de quantification). L'étape de quantification va réduire la précision des cœfficients de la transformée selon un paramètre de quantification QP (c'est donc ici que se fait la compression avec pertes). Il est important de noter que la quantification n'est pas une opération réversible. Typiquement, le résultat de la transformation est un bloc où la plupart des cœfficients sont nuls et quelques-uns sont non-nuls (il s'agit en général des cœfficients correspondants aux motifs spatiaux de basse fréquence). En prenant QP élevé, on va augmenter le pas de quantification et donc augmenter le nombre de cœfficients nuls ; on aura donc une compression plus élevée au prix d'une qualité d'image dégradée. À l'inverse, en prenant QP faible, davantage de cœfficients resteront non-nuls après la quantification ; l'image décodée sera donc de meilleure qualité mais la compression sera moins efficace.

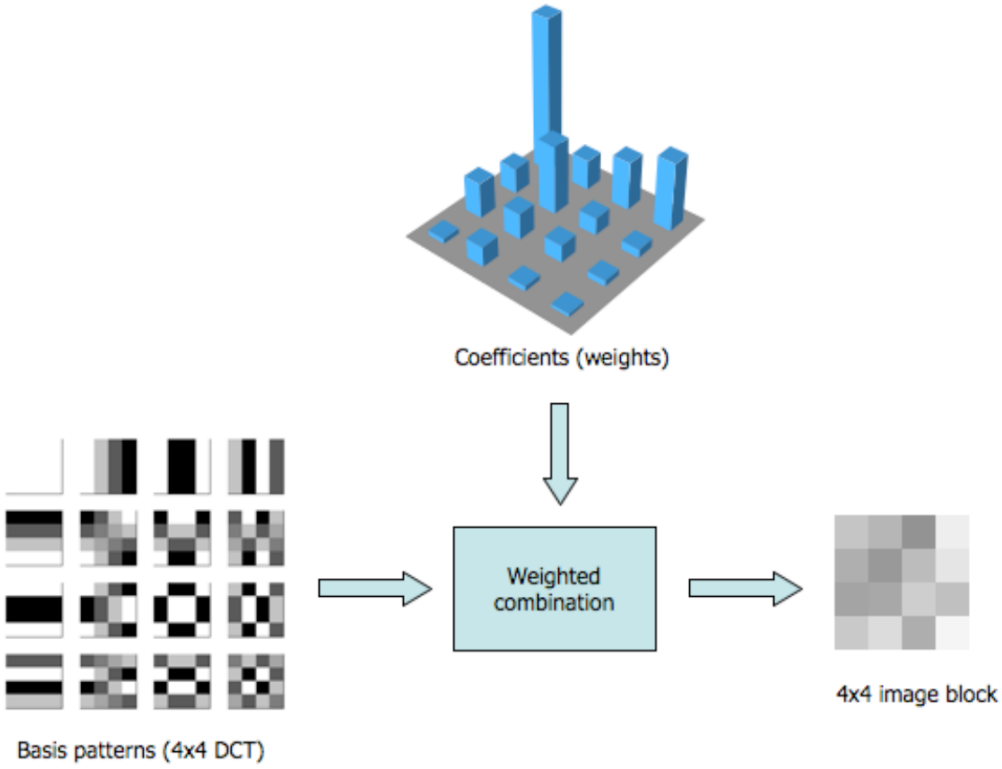

FIGURE 6 – Transformée entière inverse (source : [3]). On voit bien que l'amplitude des cœfficients de pondération tend vers zéro lorsqu'on se rapproche des hautes fréquences spatiales.

Codage binaire Le processus de codage vidéo produit un certain nombre de valeurs numériques qui doivent être codées selon la syntaxe du H.264 pour produire un flux binaire compressé interprétable. Parmi ces valeurs, on trouve :

- les cœfficients de transformée quantifiés ;
- des informations permettant au décodeur de reconstituer la prédiction ;
- des informations sur la structure des données compressées et les paramètres de compression utilisés pendant l'étape de codage ;
- des informations globales sur la séquence vidéo (résolution par exemple).

Ces valeurs et ces paramètres sont convertis en mots de code binaires grâce à des algorithmes de codage entropique ou arithmétique, qui produisent une représentation compacte et efficace de l'information. Après cette étape de codage, le flux encodé peut être stocké ou transmis.

Processus de décodage

Décodage du flux binaire Le décodeur vidéo H.264 reçoit le flux binaire compressé, décode chacun des éléments de syntaxe et extrait les informations décrites ci-dessus (cœfficients de transformée quantifiés, paramètres de compression…). Ces informations sont ensuite utilisées pour inverser le processus d'encodage vidéo et recréer une séquence d'images vidéos.

Quantification et transformation inverses Les cœfficients de transformée quantifiés sont remis à l'échelle : chaque coefficient est multiplié par le bon pas de quantification pour le restaurer à son échelle originale (comme la quantification n'est pas réversible, on n'obtient en général pas la même valeur que le coefficient de départ). Puis la transformée inverse va recombiner les motifs spatiaux de base avec leurs pondérations (les cœfficients déquantifiés) pour reconstruire les blocs de résidus. Ces blocs sont ensuite agrégés pour former un macrobloc de résidus.

Reconstruction Pour chaque macrobloc, le décodeur reforme une prédiction identique à celle effectuée par le codeur. Il lui ajoute ensuite les résidus précédemment calculés pour reconstuire le macrobloc, qui pourra enfin être affiché comme une partie d'une image de la séquence vidéo.

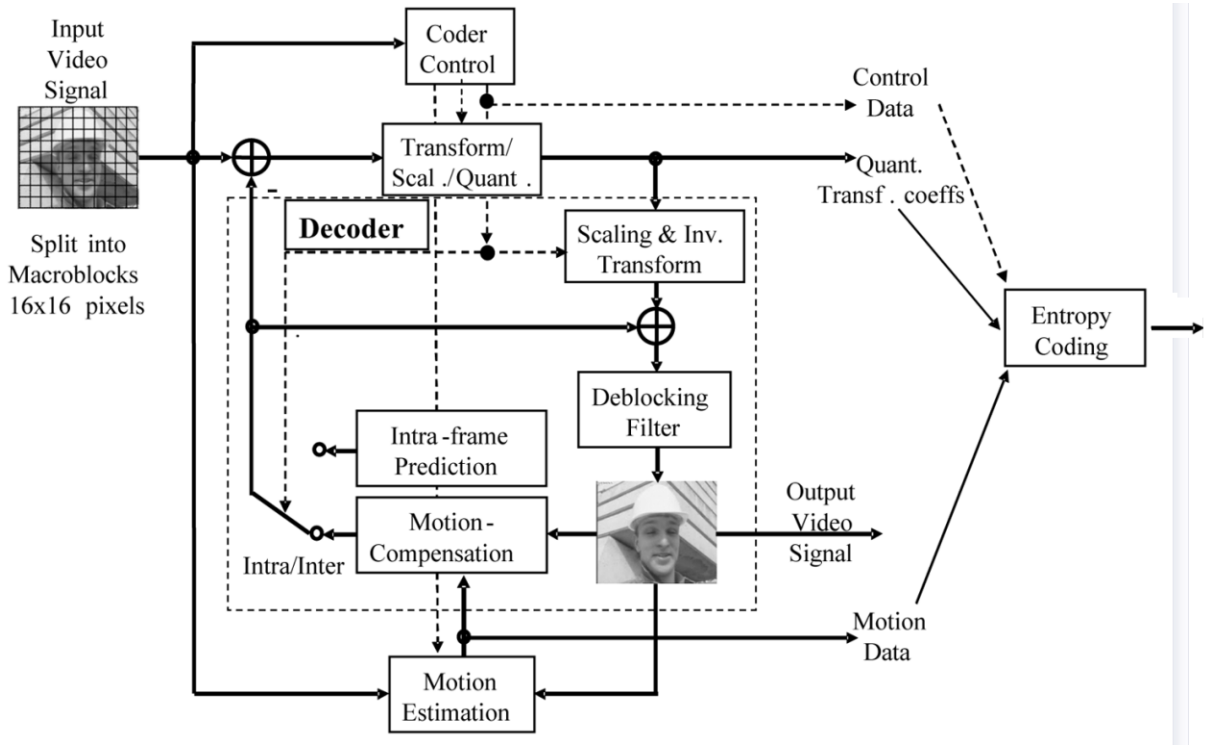

FIGURE 7 – Structure de codage détaillée pour un macrobloc (source : [2]).

3.1.3 Performances et applications

Le format H.264 est désormais un standard industriel pour la compression vidéo. Il offre, en sus de son efficacité de compression, une grande flexibilité en termes d'options de compression et de support de transmission. Un encodeur H.264 peut utiliser une grande variété de méthodes de compression, s'adaptant ainsi à des applications présentant des contraintes spécifiques comme de la transmission mobile en temps-réel à bas débit, de la télévision haute-définition ou encore de la production vidéo professionnelle. Le problème du stockage et de la transmission ont été intégrés dans la démarche de conception du standard, débouchant sur un format compressé paquétisé et des options permettant de réduire les effets des erreurs de transmission, comme des codes correcteurs d'erreurs. H.264 a été adopté pour de nombreuses applications, parmi lesquelles :

- les DVD haute-définition (HD-DVD et Blu-Ray) ;

- la télévision haute-définition en Europe ;
- les produits vidéos de la marque Apple (les téléchargements vidéos sur la plate-forme iTunes, les vidéos sur iPod et MacOS) ;
- les applications vidéo de l'OTAN et du Département de la Défense américain ;
- la télévision mobile ;
- les contenus vidéos en *streaming* sur Internet ;
- la vidéoconférence.

Le plus grand avantage du format H.264 sur les formats antérieurs est sans doute son efficacité de compression. En comparaison des standards MPEG-2 et MPEG-4 Visual, H.264 fournit en effet une meilleure qualité d'image à débit binaire égal ou un débit plus faible à qualité équivalente. Par exemple, un DVD simple-couche peut stocker un film d'environ deux heures au format MPEG-2 ; avec le format H.264, on pourrait stocker plus de quatre heures de vidéo en qualité DVD sur le même disque.

Évidemment, ce gain en taille mémoire a une contrepartie : la complexité, multiplée par 20 par rapport aux standards précédents. L'évolution du matériel permet néanmoins de remédier à cet inconvénient de taille, rendant le format H.264 accessible aussi bien aux plate-formes fixes que mobiles (CPU, GPU, FPGA, processeurs ARM...).

FIGURE 8 – Une image extraite d'une vidéo compressée à même débit au format MPEG-2 (à gauche), MPEG-4 Visual (au centre) et H264 (à droite). Source : [3].

3.2 Stéganographie et stéganalyse

La stéganographie (du grec $\sigma\tau\epsilon\gamma\alpha\nu o - \varsigma$: « je couvre » et $\gamma\rho\alpha\phi - \epsilon\iota\nu$: « j'écris ») est un cas particulier du problème de dissimulation d'informations. Le besoin de communications discrètes ou secrètes a toujours existé, comme l'attestent de nombreuses anecdotes historiques : microfilms pendant la Seconde Guerre Mondiale, histoire de l'esclave dont on avait tatoué le crâne pour en faire un messager une fois ses cheveux repoussés... Cependant les techniques de *data hiding* connaissent un intérêt croissant avec l'explosion des média numériques (image, son et vidéo). Elles ont différentes applications : communications secrètes pour la stéganographie, garantie de la propriété intellectuelle pour le *watermarking* (tatouage numérique), authentification de documents avec les signatures numériques... Cette variété d'applications induit des contraintes et exigences spécifiques : faible charge (*payload*) et grande robustesse aux attaques pour le *watermarking*, charge importante, indétectabilité pour la stéganographie.

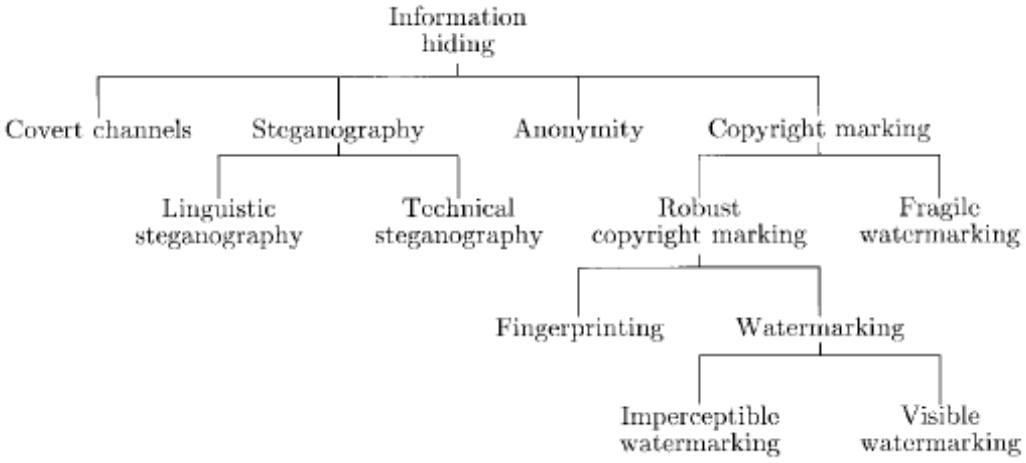

FIGURE 9 – Classification des sous-domaines du data hiding (source : [4]).

Même si l'on retrouve dans les deux cas un objectif de secret, la cryptographie et la dissimilation d'informations mettent en œuvre des moyens différents pour y arriver : en cryptographie, le message à transmettre est transformé de manière à le rendre incompréhensible à qui ne possède pas les clés de la transformation, rendant ainsi la communication suspecte. Les techniques de stéganographie permettent de protéger à la fois le message et les parties communicantes en cachant la communication dans un support d'apparence banale, qui n'attirera pas l'attention. En stéganographie pure, il n'est même pas utile de crypter le message à transmettre si le schéma d'enfouissement est suffisamment sûr.

On présentera tout d'abord une formulation théorique du problème de stéganographie, puis des techniques de data hiding pour la vidéo ; on introduira les notions de base de la stéganalyse.

3.2.1 Formalisation du problème de stéganographie

Formulation mathématique Considérons le paradigme du « problème des prisonniers » décrit par Simmons [5] : Alice et Bob sont en prison, enfermés dans des cellules séparées. Ils cherchent à communiquer pour mettre sur pied un plan d'évasion. Ils peuvent s'échanger des messages remis par des courriers, qui dévoilent leur contenu à Eve, la gardienne. Si Eve détecte un message suspect, elle interdira toute communication entre Alice et Bob. Ceux-ci sont au courant de la situation et ont convenu avant leur enfermement d'un code secret qu'ils comptent utiliser à leur avantage. Ce problème a été également formulé du point de vue de la théorie du jeu : sous cette perspective, Alice et Bob gagnent s'ils arrivent à échanger des informations sans qu'Eve ne s'en aperçoive ; Eve gagne si elle détecte une forme de communication suspecte, même sans décoder le message.

Pour ce qui est du vocabulaire, Alice et Bob sont les utilisateurs du stégosystème. Alice va modifier ou non un signal-hôte (*host signal* ou *cover signal*), selon qu'elle est active ou passive, pour y enfouir (*embed*) un message crypté selon un générateur pseudo-aléatoire initialisé par une clé privée, partagée avec Bob. Le signal modifié ou stégosignal est transmis sur un canal public contrôlé par l'adversaire (Eve). Bob le récupère, le décode et extrait le message, ou tout du moins une estimation. Eve est la stéganalyste et cherche à détecter si le signal transmis contient une information cachée. Ce processus est illustré par le schéma ci-dessous, tiré de [6]. Dans ce schéma, l'attaquant Eve est passif, c'est-à-dire

qu'il ne cherche pas à modifier le signal transmis.

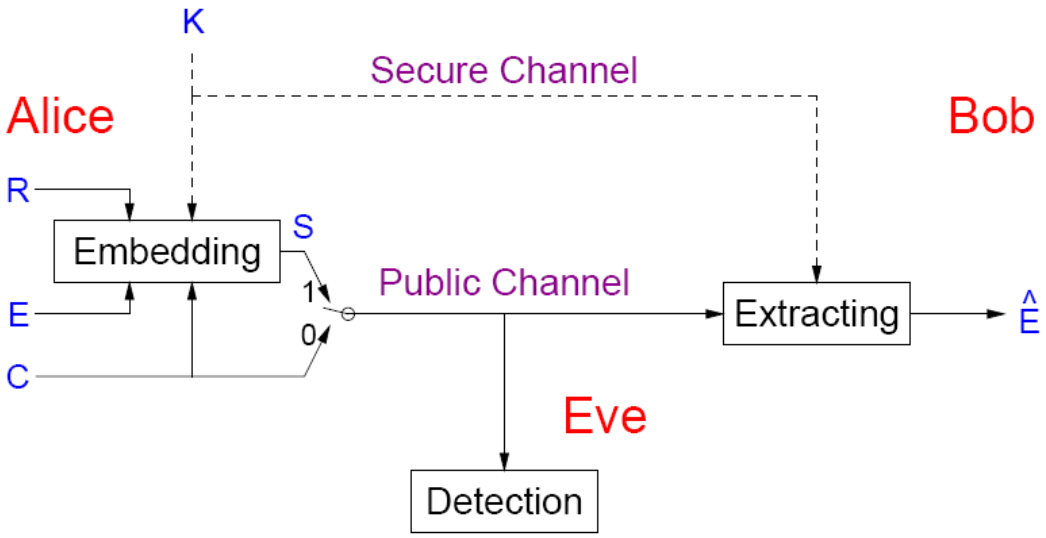

FIGURE 10 – Schéma de principe du « problème des prisonniers » (source : [6]).

Pour ne pas être détecté, le stégosignal doit résister à un examen visuel et quantitatif (souvent basé sur une analyse statistique). Ici le problème de la stéganalyse est considéré comme un problème de test d'hypothèse : Eve doit déterminer si le signal transmis a été généré selon la distribution P_C ou P_S, où C désigne le signal-hôte sans enfouissement (*cover signal*) et S le stégosignal.

On définit alors la sécurité du stégosystème par l'entropie relative entre P_C et P_S, notée $D(P_C \parallel P_S)$. Le système est dit e-sûr si $D(P_C \parallel P_S) < \epsilon$; il est parfaitement sûr si $D(P_C \parallel P_S) = 0$. La détection ne peut alors pas faire mieux que tenter de deviner au hasard.

Remarque : On définit l'entropie relative (aussi appelée divergence de Kullback-Leibler) entre deux distributions P et Q de la manière suivante [7] :

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

où P représente typiquement des données réelles et Q un modèle de P. Cette grandeur mesure la dissimilarité entre deux distributions de probabilité et peut s'interpréter comme la différence moyenne du nombre de bits nécessaires au codage d'échantillons de P selon que le codage est choisi optimal pour P ou Q. Bien qu'on puisse la considérer comme une distance, ce n'en est pas une véritable (non-respect de l'inégalité triangulaire et non symétrique).

Éléments d'un système stéganographique D'un point de vue plus systémique, un stégosystème comprend plusieurs éléments-clés :

- un modèle perceptuel pour assurer l'indétectabilité (plus éventuellement un modèle statistique de la source...) ;
- un mécanisme pour enfouir un bit d'information ;
- des techniques pour enfouir plusieurs bits d'information par modulation ou multiplexage (techniques tirées du monde des communications analogiques et numériques) ;

- les données (message) à enfouir ;
- une méthode pour gérer les zones du médium où il est difficile d'enfouir des données ;
- des méthodes d'optimisation de la robustesse et de la sécurité.

Wu et Liu [8], en faisant l'analogie avec l'architecture d'un système de communication, ont proposé un modèle en couches général de stégosystème.

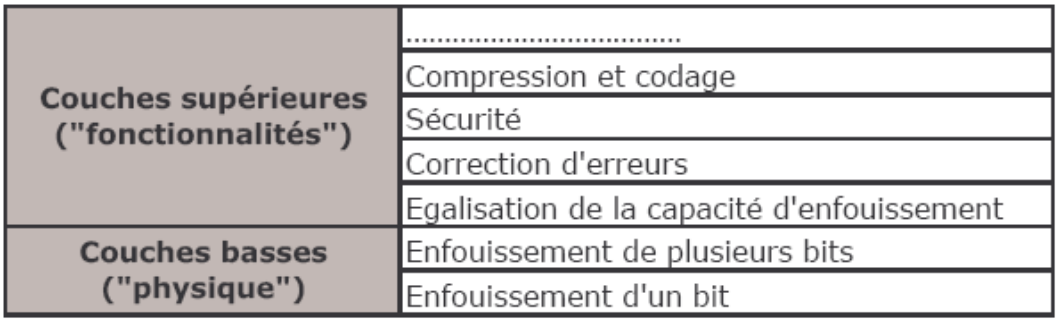

FIGURE 11 – Modèle en couches d'un stégosystème (source : [8]).

3.2.2 Techniques d'enfouissement d'information pour les images

Les articles traitant de techniques de *data hiding* pour les images ont commencé à apparaître vers le milieu des années 1990 contre la première moitié des années 2000 pour ceux s'intéressant à la vidéo. Il y a plusieurs raisons à cela :

- les images sont structurellement plus simples que les vidéos, les paradigmes de la stéganographie ont donc naturellement été mis au point sur les images ;
- le matériel informatique n'était pas encore assez performant pour une diffusion large de la vidéo à l'époque des premières recherches, ce qui avait plusieurs conséquences :
  - il n'y avait alors aucun débouché commercial ou même applicatif ;
  - les vidéos étant très rares, elles auraient pu être facilement suspectées si leurs seuls utilisateurs étaient les membres de services secrets s'échangeant des messages, or l'intérêt de la stéganographie est d'utiliser des supports qui se « fondent dans la masse » ; ce qui était un problème à l'époque ne l'est plus du tout de nos jours avec le très large trafic du à l'échange de fichiers vidéos.

De plus, les premières techniques de stéganographie vidéo ont été largement inspirées des techniques de stéganographie de l'image, au point de n'être parfois qu'une simple transposition, facilitée par les similitudes entre formats. Par exemple, les frames I du flux MPEG-2 ne sont que des images codées au format JPEG ; il est donc trivial d'étendre une technique de stéganographie spécifique au JPEG aux vidéos compressées selon le format MPEG-2.

Paradigme fondateur : la modification des LSB des pixels Une des techniques les plus simples et les plus faciles à implémenter consiste à utiliser les LSB (*Least-Significant Bit* pour bit de poids faible) des pixels d'une image numérique pour y enfouir des informations.

Une image numérique classique est formée de pixels dont les couleurs sont codées au format RGB (*Red-Green-Blue*) : chaque composante est codée sur 8 bits, un pixel a trois composantes et « pèse » donc 24 bits. Si on choisit d'enfouir les informations à transmettre dans la composante bleue par exemple, on mise sur le fait que l'œil humain ne fera pas la différence entre deux couleurs adjacentes sur 16 millions possibles (et c'est effectivement

le cas). Pour une taille d'image typique (640×480), à raison d'un bit par pixel enfoui dans la composante bleue, on peut donc transmettre de manière invisible un message de 38000 caractères (ce qui représente plus de 20 pages de texte plein).

FIGURE 12 – Environ 15000 bits, soit l'équivalent du premier chapitre de « La Chasse au Snark » de Lewis Caroll, sont enfouis dans l'image de droite (source : [9]).

FIGURE 13 – Décomposition en « plans de bits » de l'image Lena (poids fort en haut à gauche, poids faible en bas à droite). Les plans des deux derniers LSB semblent visuellement très peu corrélés à l'image originale (source : [10]).

On voit donc que la différence est imperceptible à l'œil nu. On pourrait même utiliser les deux derniers LSB de chaque composante de chaque pixel pour augmenter la capacité d'enfouissement de l'image sans créer d'artefacts visibles. Cependant cette méthode est facilement détectable par analyse des propriétés statistiques du stégofichier. De plus l'enfouissement n'est pas robuste à une compression avec pertes de type JPEG. Or une image non compressée circulant sur Internet pourrait éveiller des soupçons, même de nos jours.

Ainsi, cette technique permet de comprendre les principes de base de la stéganographie. Elle est cependant trop limitée et trop facilement détectable pour être utilisée dans des applications du domaine de la sécurité ou de la défense.

Classification des techniques existantes En se basant sur les travaux de Moulin et O'Sullivan [11] et de Chen et Wornell [12], Wu et Liu [8] ont proposé une classification des mécanismes d'enfouissement d'un bit d'information en deux types :

- mécanismes de type I : le message est ajouté au signal-hôte (éventuellement préalablement codé, modulé ou crypté) et se fond dans le bruit (inspiré des techniques SSM d'étalement de spectre du monde des télécoms). L'addition peut se faire dans un domaine spécifique (cœfficients DCT par exemple) ou sur des caractéristiques particulières calculées à partir des données brutes du signal-hôtes (somme des cœfficients d'un bloc par exemple) ;
- mécanismes de type II : l'espace des signaux est partitionné en sous-espaces (voir techniques de *binning* ou *quantization index modulation*). Chaque sous-espace est associé à un ensemble de valeurs prises par les données secondaires (c'est-à-dire le message) par une fonction de *mapping* $g()$. La relation $b = g(I_1)$ est déterministe et I_1 doit être aussi proche que possible de I_0 pour que le marquage reste imperceptible selon la notion de JND (*Just Noticeable Difference*) – on a pris pour notation I_0 le signal-hôte original et I_1 le stégosignal. La plupart des méthodes utilisant une détection aveugle (où le fichier-hôte original n'est pas connu du récepteur) sont de type II ; c'est la relation déterministe qui existe entre b et I_1 qui fait alors qu'on n'a plus besoin de I_0.

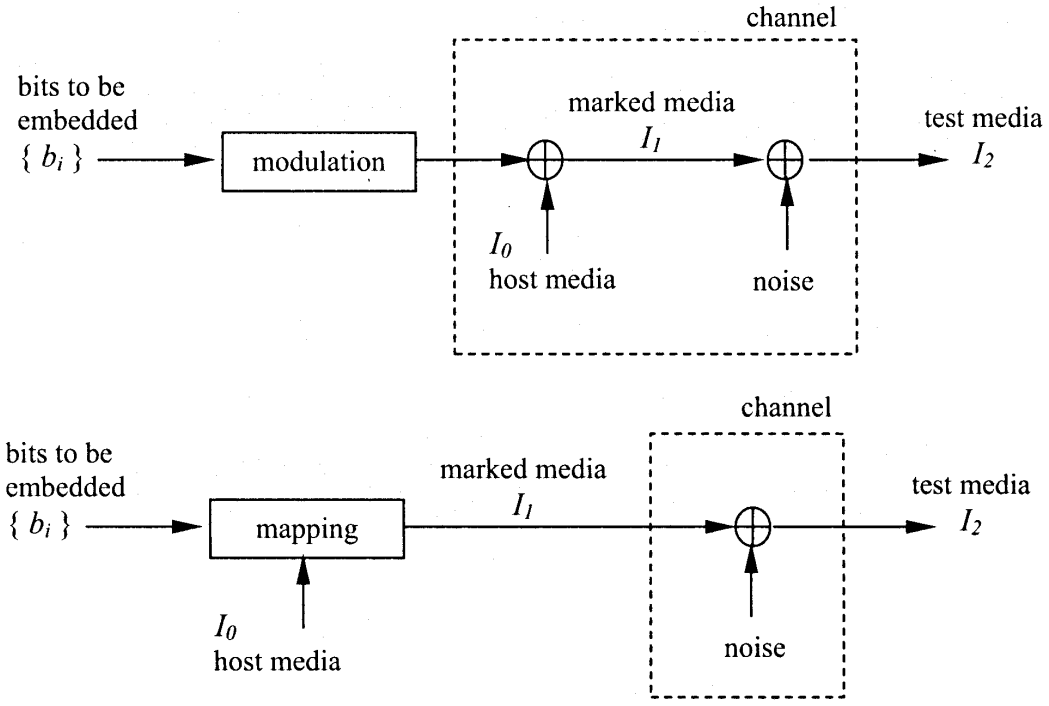

FIGURE 14 – Schémas des mécanismes de type I et II (source : [8]).

3.2.3 Techniques d'enfouissement d'information pour la vidéo

Comme on l'a dit précédemment, les techniques d'enfouissement pour la vidéo découlent pour la plupart des techniques d'enfouissement pour l'image. Cependant, avec l'explosion des contenus numériques, stockés de manière quasi-systématique dans des formats compressés, les techniques d'enfouissement au niveau pixélique sont très rares. Les recherches se penchent plutôt sur la manière d'exploiter au mieux les possibilités des formats de

compression existants. L'évolution que l'on avait vue pour l'image où le domaine d'enfouissement privilégié est progressivement devenu le domaine compressé avec l'apparition du format JPEG, se répète ici beaucoup plus rapidement. Mais là où le JPEG tient une position quasi-hégémonique, plusieurs formats pour la vidéo compressée cohabitent, étendant le champ des possibilités. Les techniques d'enfouissement modernes pour la vidéo ont donc quelque peu perdu sur le plan de la généralité puiqu'elles se spécifient pour un format donné en fonction des applications visées (MPEG-2 pour les DVD et le *watermarking*, H.264 pour la vidéosurveillance...). Elles sont moins transposables, même si les formats sont généralement basés sur les mêmes schémas de compression. Les domaines d'enfouissement utilisés sont généralement les cœfficients de transformées, les vecteurs de mouvement, des options de codage ou des méta-données.

La classification des techniques présentée ci-dessus est adaptable aux techniques vidéo.

3.2.4 Techniques de stéganalyse

La stéganalyse est la discipline duale de la stéganographie : l'objectif est la détection simple de signaux stéganographiés et non l'extraction des messages enfouis. On considère donc qu'un schéma de stéganographie n'est plus sûr si on peut détecter un stégosignal selon ce procédé. On distingue habituellement la stéganalyse passive, où le stéganalyste ne modifie pas les signaux qui transitent sur le canal de communication, de la stéganalyse active, où il peut attaquer les signaux par divers procédés dans le but de détruire ou d'altérer le message enfoui de sorte que le récepteur ne puisse en saisir le sens. On se place généralement dans le cas de la stéganalyse passive en stéganographie (la stéganalyse active est plutôt appliquée dans le cas du *watermarking* pour tenter de retirer la marque du support numérique et ainsi empêcher son identification ou son traçage).

Il est tout naturel de s'intéresser de près aux techniques de stéganalyse lorsque l'on conçoit un schéma stéganographique : on s'informe ainsi sur outils utilisés par « l'adversaire », on peut prévoir les attaques et tenter de s'en prémunir. Il n'est bien sûr pas nécessaire dans notre cas d'être au courant des techniques les plus avancées vu que l'indétectabilité parfaite n'est pas notre objectif critique, mais une connaissance de base des attaques stéganalytiques les plus courantes est tout de même essentielle pour tenter de trouver un compromis entre capacité d'enfouissement et indétectabilité.

Chandramouli et Subbalakshmi [13] ont proposé une ébauche de classification des techniques de stéganalyse en quatre groupes :

Apprentissage supervisé : ces techniques emploient une stratégie en deux étapes : une phase d'apprentissage et une phase de test. Dans la phase d'apprentissage, des exemples de médias stéganographiés ainsi que des descripteurs associés sont fournis à un classifieur statistique, qui les utilisent pour « apprendre » les meilleures règles de classification. Le modèle inféré est ensuite testé sur des exemples inconnus et utilisé pour de la stéganalyse s'il est validé.

Identification aveugle : le problème de la stéganalyse est vu ici comme un problème d'identification de système. Des propriétés statistiques comme l'indépendance du médium-hôte et du message secret sont exploitées et l'algorithme d'enfouissement est représenté comme un canal de communication. Le but est alors d'inverser ce canal pour identifier le message caché.

Paramétrisation statistique : ces approches se basent sur un modèle statistique paramétrique des médiums stéganographiés, de couverture et du message caché. La

stéganalyse est alors formulée comme un problème de test d'hypothèse, que doit résoudre un algorithme de décision statistique.

Techniques hybrides : elles utilisent plusieurs techniques décrites ci-dessus.

4 Développement d'un système stéganographique : de la conception à la validation

Les différentes démarches qui ont mené à la réalisation du système stéganographique final sont exposées dans cette partie. On trouvera d'abord le détail du choix des solutions techniques les plus à même de remplir les objectifs du cahier des charges, puis les différentes améliorations apportées, destinées à atteindre ces objectifs. La mise en place d'un « banc de test » de la résistance du système à la stéganalyse est ensuite présentée.

4.1 Étude préliminaire et choix des algorithmes

On s'est particulièrement intéressé aux techniques d'enfouissement en domaine vidéo compressé et plus spécifiquement aux schémas adaptés au format H.264/AVC. Ce choix a été fait en considérant plusieurs points :

- l'algorithme développé doit être intégré au sein d'un système d'analyse vidéo dans le domaine compressé ; travailler directement sur le flux compressé permet en effet d'éviter les opérations de décodage, de traiter des volumes de données beaucoup moins importants et donc de diviser les temps de calcul par un facteur 50 (quelques centaines de frames traitées par secondes au lieu de quelques frames par secondes pour du flux non-compressé), au prix d'une perte de précision tout à fait acceptable ;
- le format H.264 a été préféré au format MPEG-4 bien que ce dernier soit le plus commun actuellement sur les équipements intégrés de vidéo-surveillance dans une volonté de compatibilité ascendante et de placement sur les outils du futur proche. H.264 s'annonce de fait comme le standard de vidéo compressée pour la prochaine dizaine d'années ;
- malheureusement, le format H.264/AVC permettant une compression très efficace, il est plus difficile d'y enfouir des données de manière invisible ; les articles sur le sujet sont donc assez rares. Les techniques d'enfouissement de stéganographie s'appuient effectivement sur la redondance du signal pour remplacer une partie de l'information non-significative par un message à transmettre. En décorrélant le signal et en minimisant la redondance, la compression rend donc la tâche d'enfouir des informations sous de faibles distorsions très ardue.

On s'est donc également renseigné sur des techniques indépendantes du format d'encodage et sur des techniques spécifiques à des formats obsolètes (MPEG, MPEG-2...) dans un soucis d'exhaustivité et pour étudier l'évolution historique du domaine en fonction des contraintes nouvelles apportées par l'apparition de standards de compression plus performants.

4.1.1 Présentation détaillée de trois schémas de l'état de l'art

Schéma de Nguyen, Tay et Deng (2006) Les auteurs ont présenté dans [14] une technique de *watermarking* spécifique au format H.264/AVC, basée sur les *motion vectors* dans un but de faible complexité. Cette approche est également applicable aux formats MPEG.

L'enfouissement se fait dans les LSB des *motion vectors* (MV) pour diminuer le temps de calcul. Un MV se présente sous la forme {MVX, MVY}, où MVX et MVY représentent respectivement les composantes horizontale et verticale du déplacement du bloc. Chaque composante est codée par un nombre entier représentant le nombre d'unités du mouvement

dans la direction considérée, une unité valant un quart de pixel. L'information est enfouie dans les deux LSB de la composante la plus grande. Avant d'être marquée, la composante est quantifiée et réglée à la position du pixel entier le plus proche ; ainsi l'erreur maximum introduite par l'enfouissement sera d'un demi-pixel.

Un point à considérer est la réduction de la propagation des erreurs. En effet, dans le format H.264/AVC, les MV sont codés de manière différentielle comme suit : $MV = MVP + MVD$ où MVP est un MV prédit en fonction des MV voisins et MVD est l'erreur de prédiction, codée dans le flux binaire H.264 avec le code Exp-Golomb. L'enfouissement provoquera la modification de MVD. Il faudra donc un mécanisme de compensation d'erreur sur les MVP voisins pour être sûr qu'on récupère les bonnes valeurs de tous les MV au décodage (ce processus peut éventuellement faire augmenter la taille du flux binaire).

Un processus de sélection des blocs éligibles à l'enfouissement est effectué au préalable. Il élimine directement les blocs des régions de faible mouvement et les blocs « sautés » (*skipped*) et trie les autres par rapport à une fonction de coût qui va calculer le nombre de pixels qui seront modifiés (*distorted*) dans cette frame et les frames qui lui font référence si on enfouit des données dans le bloc considéré. On travaillera ensuite sur les blocs dans l'ordre ainsi établi.

L'enfouissement se fait directement sur les éléments syntaxiques du flux binaire H.264 : on repère les MVD et on modifie leur code Exp-Golomb, ce qui est rapide car c'est un code beaucoup plus simple que les autres codes entropiques normalement utilisés (CABAC et CAVLC par exemple).

Le processus de pré-tatouage suivant est appliqué à la frame F_i en cours de décodage :

1. Si F_i est une frame I, on passe directement à la frame suivante ; sinon on continue sur les étapes suivantes.
2. Tous les MV de F_i sont décodés et on retient les positions des MVD correspondantes dans le flux binaire.
3. Le coût de chaque bloc est initialisé avec l'aire du bloc en pixels.
4. On calcule effectivement le coût pour chaque bloc.
5. On stocke la frame F_i dans le DPB (*Decoded Picture Buffer*) et on passe à la frame suivante.

Quand F_i est retirée du DPB ou qu'elle n'est plus référencée, on applique le véritable processus de tatouage :

1. Dans la frame courante F_i, tous les blocs dont les composantes du MV sont inférieures à 2 et tous les *skipped macroblocks* ainsi que leurs voisins sont étiquetés comme « non-marquables ». Si le nombre de MV éligibles est inférieur à un seuil prédéfini L (*watermarking rate*), F_i ne sera pas tatouée et les étapes suivantes seront sautées.
2. Tous les MV de F_i sont triés selon leur coût ; on utilisera les L premiers pour la suite.
3. On enfouit les bits de la marque dans les LSB des MV choisis par quantification.
4. On modifie les MVD correspondants.
5. On modifie les MVD des blocs voisins pour éviter la propagation des erreurs due au mécanisme de prédiction des MV.
6. On remplace tous les MVD modifiés dans le flux binaire.

On a donc une latence entre la récupération de la marque et le décodage des frames due au fait que H.264 supporte des références jusqu'à 16 frames en avant ou en arrière (la latence maximum est donc de 16 frames). On peut choisir de ne pas marquer les frames faisant des références à long terme pour éviter de trop grandes latences.

Cette technique de *watermarking* est donc intéressante par sa faible complexité (le temps d'enfouissement ou d'extraction de la marque représente moins de 30% du décodage vidéo) car elle peut être appliquée dans un contexte temps-réel. La capacité maximale théorique est uniquement fonction de la résolution puisqu'un vecteur de mouvement est calculé par macrobloc si l'on n'utilise pas la prédiction intra-image dans les frames P (on peut monter jusqu'à un vecteur par bloc 4×4 sur les zones fortement texturées). À titre d'ordre de grandeur, elle est donc de 9 Kbits/s en résolution CIF avec un GoP de longueur 16 de la forme I/P, à 25 images/s et en considérant que l'on a un vecteur de mouvement par macrobloc et que l'on n'enfouit qu'un bit par vecteur. L'article indique que les tests menés par les auteurs ont permis d'atteindre au maximum 210 bits/frame P en résolution CIF (soit environ 2950 bits/s à 15 images/s, *framerate* des vidéos de leur corpus de test), ce qui est utilisable dans un contexte de stéganographie. Enfin, la technique présentée résiste à différents types d'attaques (compression...) ; seule l'évaluation de la sécurité stéganalytique n'a pas été conduite.

Schéma de Hu, Zhang et Su (2007) Cette méthode spécifique au format H.264/AVC [15] permet l'enfouissement de données par association des modes I4 en prédiction intra-frame avec des mots binaires (se reporter à [2] pour plus de détails sur les modes I4).

Le format H.264/AVC va encore plus loin dans l'élimination des redondances spatiales en mettant en place un mécanisme de prédiction dans les frames I du flux : le contenu des blocs des coefficients IT est prédit d'après les coefficients du bord du bloc (pour plus de détails sur la prédiction intra, se référer à [3]). Le mode de prédiction permettant la meilleure approximation est sélectionné parmi plusieurs disponibles, représentant différentes pondérations des coefficients de bord. De plus, le codec permet de prédire le mode utilisé d'après les modes des blocs adjacents de gauche et du dessus. Si le mode est correctement prédit par ceux des blocs voisins, il porte un *flag* F $= 1$ (on utilise le mode le plus probable) ; sinon, il porte le *flag* F $= 0$. La distribution des *flags* est fortement corrélée avec celle des *features* de l'image : on aura majoritairement des 0 pour les régions texturées et des 1 pour les régions unies.

Comme on l'a dit plus haut, les modes, au nombre de neuf en prédiction INTRA-4×4, sont associés à des mots binaires. Le codec va calculer pour le bloc courant le mode le plus probable (MPM) en fonction des blocs voisins et le mode optimal (OPT). Si $MPM \neq OPT$, le *flag* F vaudra 0 et le bloc sera éligible pour l'enfouissement. Pour enfouir l'information, on va remplacer le mode optimal désigné par le codec par le mode directement suivant. Il nous faut donc une règle de *mapping* qui, pour un MPM donné, divise les huit modes restant en deux groupes, l'un associé au bit à enfouir 0, l'autre à 1. Pour cela, un apprentissage sur des échantillons d'images a été réalisé pour trouver les regroupements optimaux dans la plupart des cas.

MPM	**Candidate Modes**	
	Group *M* (mapping to 0)	**Group *N*** (mapping to 1)
Mode 0	1, 2, 3, 4	5, 6, 7, 8
Mode 1	0, 3, 4, 8	2, 5, 6, 7
Mode 2	0, 3, 4, 8	1, 5, 6, 7
Mode 3	0, 5, 6, 8	1, 2, 4, 7
Mode 4	0, 3, 6, 8	1, 2, 5, 7
Mode 5	0, 3, 6, 8	1, 2, 4, 7
Mode 6	0, 3, 4, 8	1, 2, 5, 7
Mode 7	0, 5, 6, 8	1, 2, 3, 4
Mode 8	0, 1, 3, 4	2, 5, 6, 7

FIGURE 15 – Regroupements des modes les plus probables, d'après un apprentissage réalisé sur plus de 100 000 blocs I4 tirés de 10 séquences vidéos différentes (source : [15]).

D'autre part, l'enfouissement est contrôlé par un paramètre (*embedding strength*) qui permet d'ajuster le taux de blocs éligibles et donc de limiter la capacité du fichier-hôte. Ce faisant, on limite également une éventuelle augmentation du débit binaire.

Cette technique est donc assez innovante car elle utilise les modes I4 pour le transport de l'information, alors que les techniques de stéganographie ou de *watermarking* de la littérature se basent habituellement sur les coefficients DCT ou IT ou encore sur les *motion vectors*. De plus elle n'introduit pas de distorsion visible puisque les erreurs sont compensées par les résidus [2]. Elle est enfin de faible complexité et peut venir s'insérer directement à l'encodage. La capacité maximale théorique est de 1584 bits/frame I en résolution QCIF ; dans les tests, les auteurs enfouissent entre 60 et 870 bits/frame I [3], selon un paramètre de « force d'enfouissement ». Il faut néanmoins noter le biais dans le choix du mode optimal fait augmenter le débit binaire du fichier-hôte, et cela linéairement par rapport à la charge enfouie (à 870 bits/frame I, le débit augmente de plus de 4%).

Schéma de Noorkami et Mersereau (2005) Dans leur article [16], les auteurs présentent une méthode de *watermarking* de faible complexité conçue spécialement autour des caractéristiques du format H.264. L'enfouissement de la marque se fait dans les coefficients DCT des frames I du flux compressé. Ce choix s'explique par la trop forte compression des frames de type P et B, entraînant une faible capacité stéganographique.

On enfouit un bit de la marque par bloc sélectionné, dans un coefficient choisi aléatoirement pour plus de sécurité. Le fait de n'altérer qu'un seul coefficient évite la formation

2. Attention, cela n'est vrai qu'à faible compression car les résidus sont ensuite quantifiés. À forte compression, la quantification est trop forte et les résidus ne permettent pas la reconstruction exacte de l'image de base au décodage. Les auteurs n'avaient pas relevé ce problème dans l'article.

3. Cela équivaut à une plage de 180 à 2600 bits/s pour un *framerate* de 30 images/s et une taille de GoP de 10.

d'artefacts visibles et contraint un attaquant à modifier en moyenne au moins la moitié des coefficients pour retirer la marque, rendant alors la vidéo inutile car trop dégradée.

La sélection du coefficient du $i^{ème}$ bloc est contrôlée par une clé. Or se baser sur la même clé pour toutes les frames rendrait la détection trop aisée ; une clé très longue générée par une clé publique (extraite à partir des informations locales de la vidéo) et une clé privée (gardée secrète par le détenteur des droits d'auteur) est donc utilisée. Bien entendu, la clé publique K_p doit être construite à partir de caractéristiques de la vidéo que l'attaquant ne peut pas modifier. Il est donc proposé d'exploiter les différences relatives entre coefficients DC des blocs 4×4 du macrobloc courant.

En pratique, la sélection du coefficient se fait ensuite de la manière suivante :

- deux bits de la clé sélectionnent le bloc 8×8 au sein du macrobloc considéré ;
- les deux bits suivants sélectionnent le bloc 4×4 dans le bloc 8×8 choisi ;
- les quatre bits suivants donnent l'emplacement du coefficient AC où est enfoui le bit de la marque.

Concernant l'étape d'enfouissement proprement dite, elle est effectuée sur les coefficients AC quantifiés pour éviter toute perte d'informations. Le coefficient cw_i, désigné par un portion binaire de la clé K, est modifié comme suit :

si $W_i = 0$

$$cw_i = \begin{cases} cw_i - 1 & \text{si } cw_i \equiv 0 \pmod 2 \\ cw_i & \text{sinon} \end{cases}$$

si $W_i = 1$

$$cw_i = \begin{cases} cw_i & \text{si } cw_i \equiv 0 \pmod 2 \\ cw_i - 1 & \text{sinon} \end{cases}$$

On voit donc que l'on augmente ou diminue le coefficient désigné pour l'enfouissement que d'une unité au maximum. Cela permet de limiter les distorsions visuelles.

D'après les auteurs, cette méthode ne change pas le débit binaire moyen du fichier-hôte, puisqu'on ne modifie qu'un coefficient choisi à chaque fois aléatoirement et qu'on utilise ensuite un codage entropique. Bien entendu, on aura pour cela évité d'enfouir des données dans les blocs *all-zeros*.

Malgré le fait qu'il soit conçu pour une application de watermarking, ce schéma de data hiding peut être utilisé pour de la stéganographie car sa capacité maximale théorique est fixée par la résolution utilisée. Si elle paraît faible en QCIF (99 bits/frame I, soit 154 bits/s à 25 images/s et avec une taille de GoP de 16), elle se rapproche de la consigne du cahier des charges en 4CIF, résolution visée (1584 bits/frame I, soit 2475 bits/s avec les mêmes conditions). D'après les résultats présentés dans l'article, l'augmentation du débit binaire est faible (0,5% en moyenne). La sécurité n'a en revanche pas été évaluée du point de vue de la stéganalyse.

4.1.2 Comparatif des schémas

Les algorithmes retenus ont été évalués selon plusieurs critères pertinents par rapport au cahier des charges. Le tableau récapitulatif des résultats de ces évaluations se trouve ci-dessous.

	Schéma 1 (Nguyen, Tay et Deng)	Schéma 2 (Hu, Zhang et Su)	Schéma 3 (Noorkami et Mersereau)
Capacité	Assez importante (170 bits/frame P en résolution CIF - 352*288)	Très variable (de 60 à 870 bits/frame I en QCIF)	Faible (jusqu'à 99 bits/frame I en résolution QCIF - 176*144)
Capacité selon les conditions du Cahier des Charges	Projection d'après les résultats de l'article : 15,9 Kbits/s ; capacité maximale théorique : 37,1 Kbits/s	Projection : de 1,5 à 21,8 Kbits/s ; capacité maximale théorique : 39,6 Kbits/s	Capacité maximale théorique : 2,5 Kbits/s
Sécurité	Résistant à des attaques de types *watermarking* (compression...) ; sécurité du point de vue de la stéganalyse non évaluée	N'introduit pas de distorsion (très important) ; cependant l'encodage n'est pas optimal (augmentation du débit binaire) et il peut y avoir un problème de "compatibilité H,264" ; évaluation par des méthodes de stéganalyse non effectuée	Pas ou peu d'augmentation du débit binaire ; coefficients choisis aléatoirement ; évaluation par des méthodes de stéganalyse non effectuée
Domaine d'enfouissement	LSB des *motion vectors* des frames P et B	Modes I4 en intra-prédiction dans les frames I	Coefficients IT dans les frames I
Générique ?	Non ; spécifique H.264 mais transposable au MPEG	Non ; spécifique H.264	Non ; spécifique H.264
Classe du mécanisme	Type II	Classification non applicable	Type II
Complexité	Faible (l'extraction du message représente environ 20% du temps de calcul pour le décodage H.264)	Faible (non quantifiée)	Moyenne (non quantifiée)
Non-perturbation des ROI	Possible a priori ; pose par contre un problème de diminution de la capacité	Pas de distorsion ! On peut donc enfouir sur les ROI	Possible a priori
Date	2006	2007	2005
Particularités	Latence au décodage (jusqu'à 16 frames)	H.264 Main Profile (RDO, CABAC, QP = 28, 30 fps et GoP = IBPBPBPBPB)	Néant

FIGURE 16 – Comparatif des trois schémas sélectionnés (source : [1]).

4.1.3 Choix des algorithmes implémentés

Sur les trois algorithmes retenus et présentés ci-dessus, seuls les schémas 1 (Nguyen, Tay et Deng) et 3 (Noorkami et Mersereau) ont été implémentés. Il existe plusieurs raisons à cela :

- le troisième schéma est basé sur les modes I4, or cette fonctionnalité n'existe pas encore dans le codeur propriétaire développé au sein du laboratoire ;
- l'article présentait très peu de détails, compliquant la tâche d'implémentation.

On a donc implémenté le schéma de Norkaami et Mersereau et celui de Nguyen, Tay et Deng, conformément à la description donnée dans les articles dans un premier temps. Cela a permis de se familiariser avec les domaines d'enfouissement utilisés et de reproduire les résultats présentés. D'un point de vue technique, la programmation s'est effectuée en langage C, avec une grande variété algorithmique (algorithmes de tri, génération aléatoire de nombres, manipulation de bits, gestion de fichiers...).

4.1.4 Faiblesses à adresser

Une fois implémentés, les deux schémas finalement retenus ont été testés. On a ainsi pu pointer certaines faiblesses vis-à-vis des objectifs du cahier des charges.

Pour le schéma de Noorkami et Mersereau :

Réglage de la quantité de données enfouies : en l'état, l'algorithme enfouit un bit dans chaque macrobloc éligible de chaque frame I, de manière figée. Il n'est pas possible de moduler la charge.

Problème de la distorsion par rapport au critère de non-perturbation des ROI : l'algorithme crée une légère distortion dûe à la modification des coefficients quantifiés de la transformée. Il faut donc soit régler ce problème, soit renoncer à enfouir des données sur les ROI, ce qui aurait pour résultat de diminuer la capacité d'enfouissement. De plus, les erreurs sont amenées à se propager du fait de la prédiction intra-image et cela n'est pas du tout maîtrisé en l'état.

Augmentation du débit binaire : pour enfouir un bit d'information, un coefficient AC quantifié est choisi aléatoirement. Or le résultat typique de la transformation entière d'un bloc suivi de l'opération de quantification est un bloc dont les coefficients de basse fréquence sont non-nuls et les coefficients de moyenne et haute fréquence sont nuls. On a donc une probabilité non-négligeable de choisir un coefficient nul et de le rendre non-nul par l'enfouissement. Cela a pour effet de perturber le codage entropique (la suite de zéros est brisée, elle ne peut plus être codée en une seule fois) et donc d'augmenter le débit.

Capacité stéganographique : l'algorithme enfouit un bit par macrobloc de chaque frame I. La capacité maximale théorique est donc fixée par la résolution, le *framerate* et les caractéristiques de taille et de forme du GoP (*Group of Picture*). À titre d'exemple, en résolution 4CIF à 25 images/s avec un GoP de 16 images (valeurs usuelles), on atteint un débit maximal d'informations enfouies d'environ 2500 bit/s, ce qui reste assez loin de l'objectif de 5000 bits/s fixé dans le cahier des charges.

Problèmes de désynchronisation : on s'est aperçu en testant le schéma que certains cas particuliers d'enfouissement n'étaient pas correctement gérés et causaient une désynchronisation au décodage. Par exemple, considérons un bloc dont tous les coefficients sont nuls sauf un, dont la valeur est 1. Si ce coefficient est choisi pour l'enfouissement et qu'il est modifié en 0, on obtient un bloc *all-zeros*, que le décodeur ignorera à la réception. Il ne s'agira donc pas d'une simple erreur, mais d'un « effacement au codeur » (*deletion*), qui provoquera une désynchronisation entre l'émission du message et la réception.

Pour le schéma de Nguyen, Tay et Deng :

Réglage de la quantité de données enfouies : en l'état, l'algorithme enfouit deux bits par vecteur de mouvement éligible. Il n'est pas possible de moduler la charge.

Problèmes de désynchronisation : on a également remarqué des problèmes de désynchronisation lors des tests. Ils sont visiblement dûs à des imprécisions dans les règles de sélection de la composante-hôte du vecteur.

Problème de la résistance au transcodage : les vecteurs de mouvements sont très souvent modifiés lors d'un transcodage. Les informations enfouies seraient donc perdues le cas échéant. C'est particulièrement problématique pour un algorithme censé être utilisé dans des applications de *watermarking*, où le tatouage numérique doit être robuste à toutes sortes d'attaques, mais c'est aussi gênant pour dans le cadre de notre projet car le transcodage H.264-H.264 est fréquent dans une chaîne complète de traitement vidéo. Il faut noter que cette remarque n'est pas directement reliée aux critères de notre cahier des charges. Il nous a cependant semblé important de la mettre en évidence car c'est une faiblesse indéniable de l'algorithme.

Ainsi, l'analyse des besoins et des solutions techniques disponibles a permis d'établir avec un regard critique les points forts et les points faibles des algorithmes sélectionnés.

4.2 Améliorations apportées à l'existant

Des solutions ont été conçues au cours du stage pour pallier aux manquements mis en évidence plus haut. Certaines ont été validées et retenues. Elles sont présentés ci-dessous.

Il faut cependant noter qu'il existe une différence de taille entre les techniques de l'état de l'art et le système qui a été développé : les premières sont issues du *watermarking* et travaillent donc sur des vidéos déjà compressées. Nous nous situons dans le cas d'un module de stéganographie intégré dans un codec, ce qui est un avantage, ne serait-ce qu'en termes de complexité : on n'a pas besoin d'effectuer de décodage entropique ou d'étape de *parsing* avant de réaliser l'enfouissement, le module est directement appelé depuis l'encodeur lorsque les données adéquates sont prêtes à être modifiées.

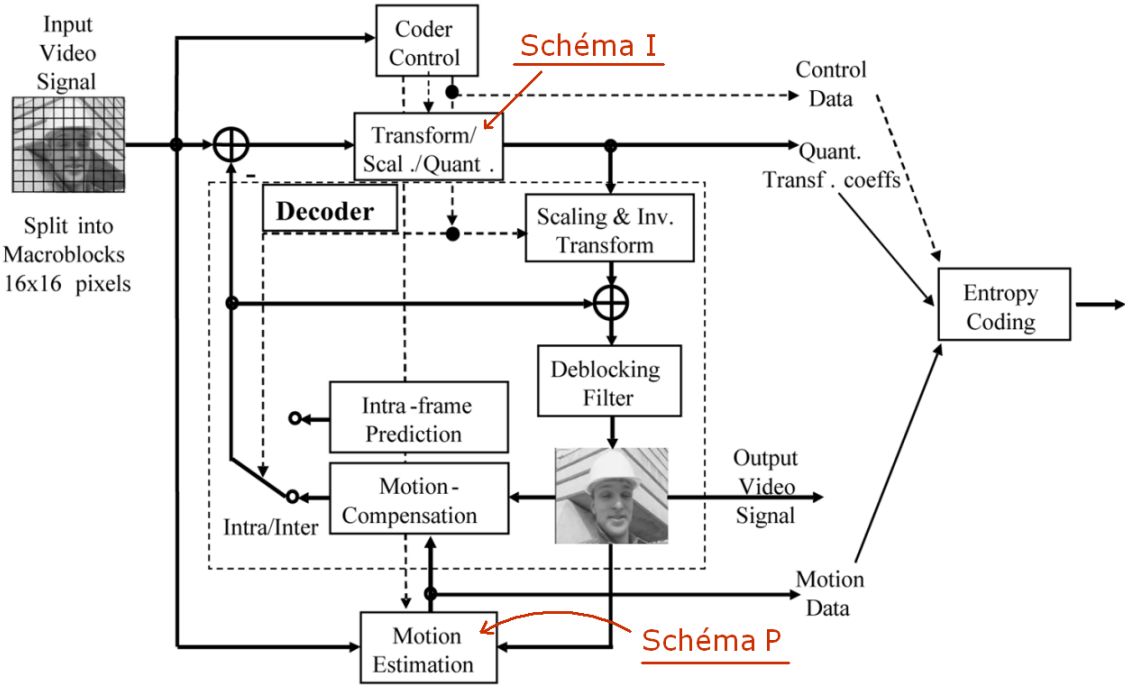

FIGURE 17 – Ce schéma présente l'interfaçage des schémas I et P (respectivement Noorkami et Mersereau ; Nguyen, Tay et Deng) au niveau du codeur.

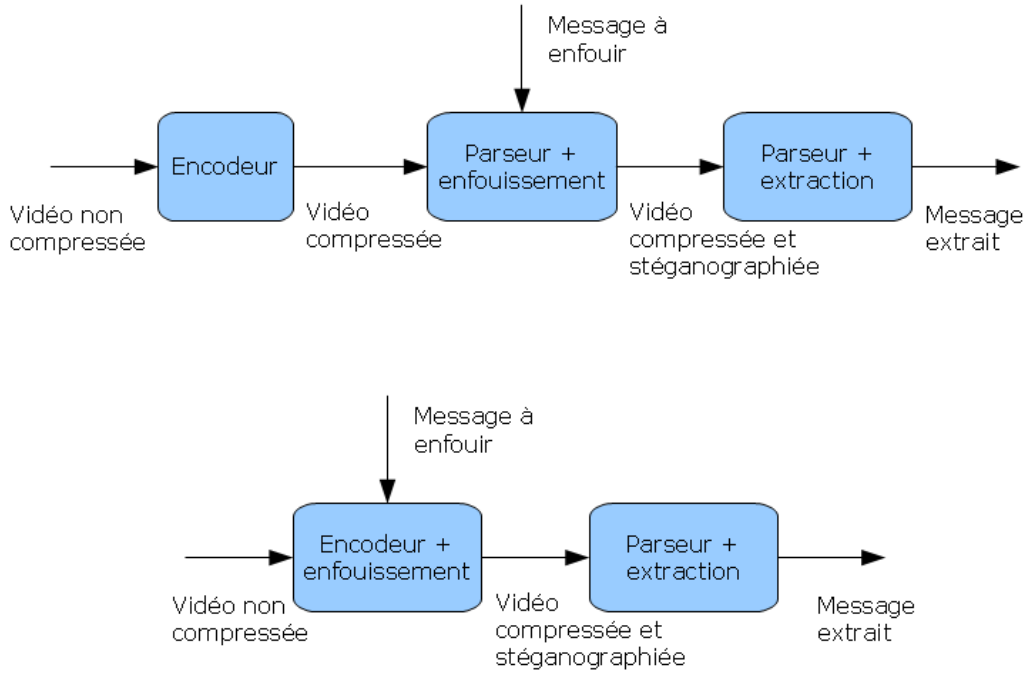

FIGURE 18 – La chaîne d'enfouissement complète représentée en haut correspond au cas d'application dans lequel se situent les algorithmes de l'état de l'art. La chaîne du bas représente notre cas d'application, où l'enfouissement est réalisé pendant l'étape de compression.

4.2.1 Améliorations apportées au schéma de Noorkami et Mersereau

Capacité d'enfouissement variable L'algorithme de base a été adapté pour pouvoir spécifier une capacité d'enfouissement variable. Celle-ci est contrôlable via deux paramètres : une capacité globale par frame I C_g et une capacité locale C_l exprimée en bits par macrobloc. Le principe est le suivant : pour chaque frame I, on sélectionne pour l'enfouissement $p = \frac{C_g}{C_l}$ macroblocs parmi les N disponibles, de manière aléatoire. Puis pour chaque macrobloc sélectionné, on sélectionne aléatoirement C_l blocs. On enfouit un bit d'information dans un coefficient dans chacun de ces C_l blocs, toujours aléatoirement.

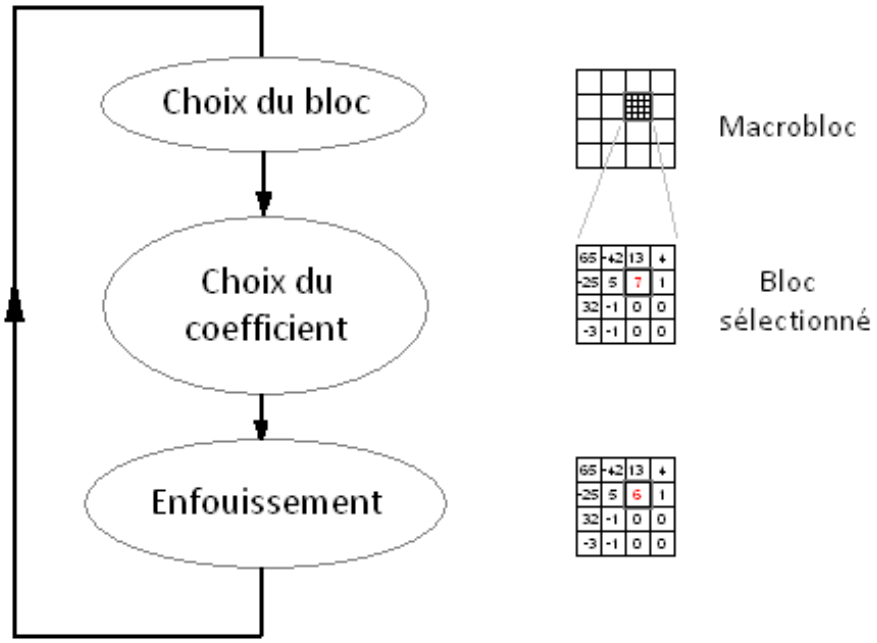

FIGURE 19 – Enfouissement d'un bit d'information. Ce processus est répété C_l fois pour le macrobloc courant.

Pour la sélection du premier bloc et du coefficient au sein du macrobloc courant, on garde la méthode employée par Noorkami et Mersereau, basée sur une clé d'enfouissement générée par une clé publique et une clé privée. Les blocs et coefficients suivants sont sélectionnés en transformant la clé d'enfouissement selon la technique du registre à décalage.

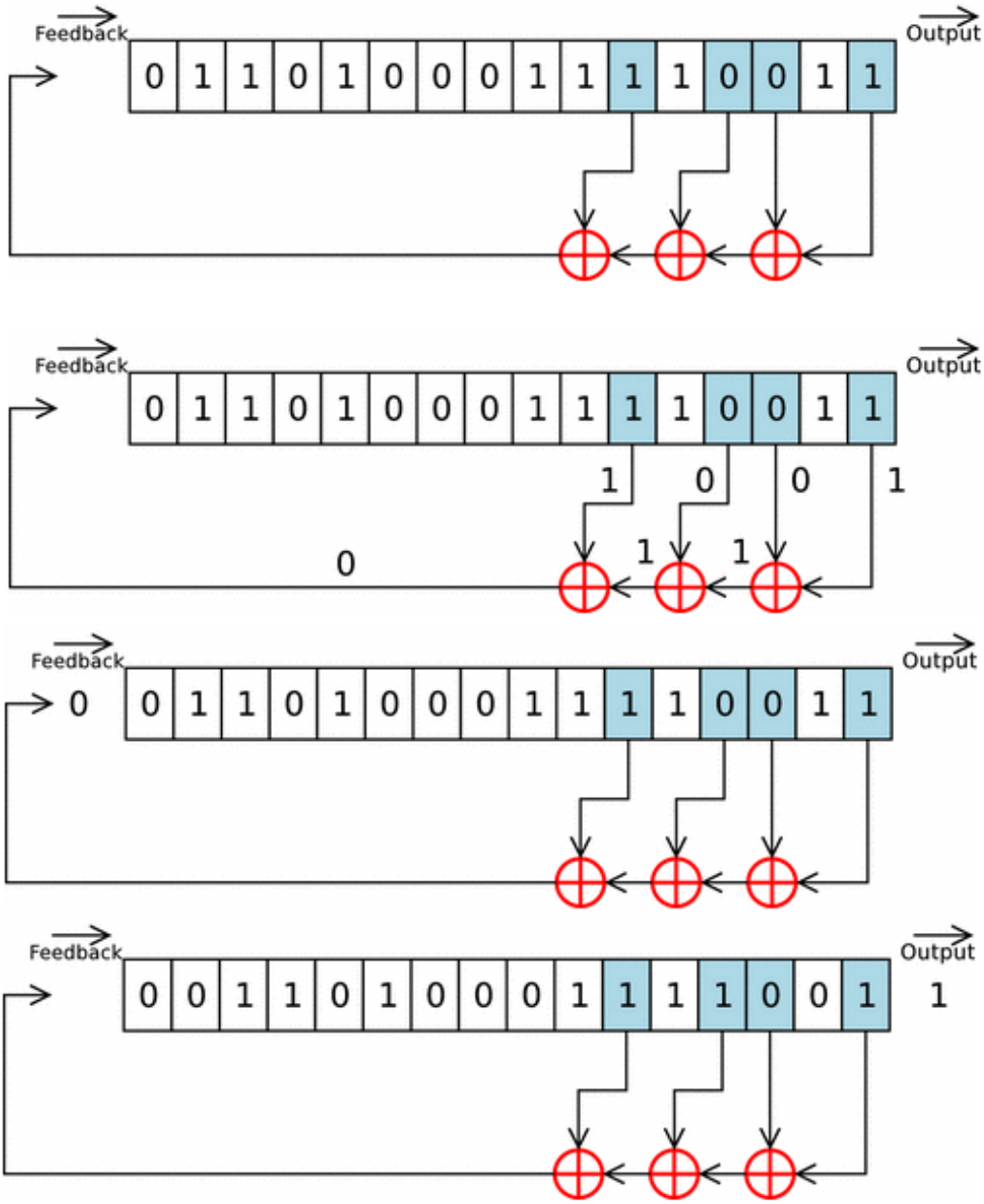

FIGURE 20 – Fonctionnement d'un registre à décalage avec rétroaction linéaire basée sur un XOR entre plusieurs bits (source : [17]).

Diminution du débit Comme on l'a vu, l'algorithme dans sa version originale choisit aléatoirement les coefficients AC pour enfouir des données. Il risque donc de perturber le codage entropique en modifiant les coefficients nuls de fin de bloc. On a donc modifié les règles d'éligibilité de bloc :

> un bloc est éligible tant que la longueur de la chaîne des coefficients nuls de fin de bloc est strictement inférieure à 14.

Ou, dans d'autres termes :

> un bloc est éligible s'il a au moins un coefficient non-nul entre la $3^{ème}$ et la $16^{ème}$ position (indice 2 à 15).

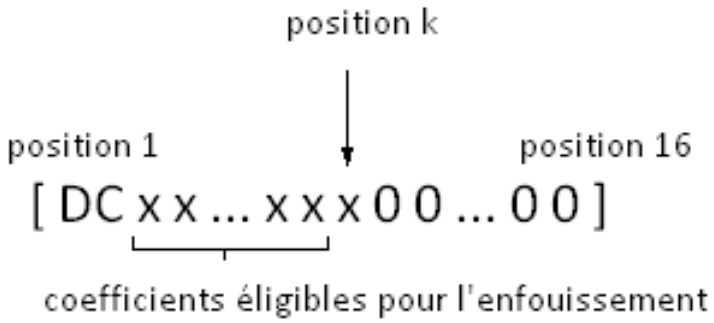

FIGURE 21 – Position des coefficients éligibles à l'enfouissement dans le bloc sélectionné (k est compris entre 3 et 16 et le coefficient de la position k doit être non-nul pour que le bloc soit éligible).

En effet, on ne modifie pas le coefficient DC quoiqu'il arrive et on évite de modifier les coefficients nul de fin de bloc. Mais on ne peut pas modifier aveuglément le dernier coefficient non-nul non plus car on risquerait d'avoir des erreurs au décodage, voire de perdre la synchronisation. Supposons que le dernier coefficient non-nul, à la position 8 par exemple, a une valeur de 1 et que l'enfouissement le transforme en 0. Au décodage, on cherchera le dernier coefficient non-nul à la position 7 ou inférieure et on aura donc une erreur.

Pour éviter cela, on va donc choisir aléatoirement un coefficient situé entre la position 1 et la position précédant la position du dernier coefficient non-nul. On supprime ainsi les éventuelles erreurs et on ne perturbe pas le codage entropique.

Diminution de la distorsion Il a été remarqué lors de tests que la distorsion introduite par l'enfouissement se propage dans les frames I, du fait de l'étape de prédiction intra-frame au niveau du codeur. Le codeur propriétaire du laboratoire ne gère actuellement que la prédiction Intra-16 × 16 [4], dans laquelle le macrobloc entier est prédit à partir de la ligne supérieure de pixels et de la colonne de pixels directement à gauche du macrobloc.

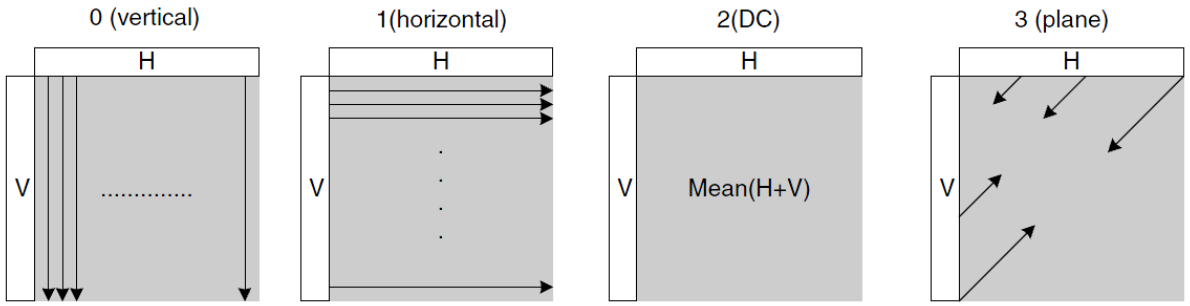

FIGURE 22 – Ce schéma présente les quatre modes de prédiction Intra-16 × 16 (source : [18]).

4. Détail des modes :
   - Mode 0 (*vertical*) : extrapolation à partir de la ligne des pixels supérieurs (H).
   - Mode 1 (*horizontal*) : extrapolation à partir de la colonne des pixels de gauche (V).
   - Mode 2 (DC) : moyenne des pixels supérieurs et des pixels du côté gauche (H + V).
   - Mode 3 (*plane*) : une fonction linéaire plane est calculée à partir des pixels de H et de V et appliquée ensuite au macrobloc (efficace dans les zones où la luminance varie peu).

On voit donc que si l'on a enfoui des informations dans les blocs inférieurs du macrobloc supérieur, les distorsions potentiellement créées peuvent se propager au macrobloc courant. Pour éviter cela, les blocs éligibles seront les neuf blocs les plus en haut et à gauche du macrobloc.

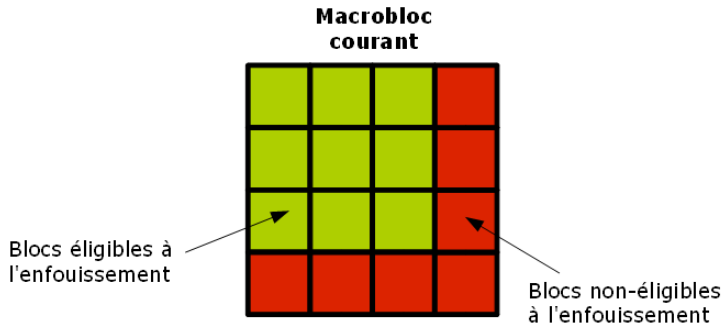

FIGURE 23 – On limite l'éligibilité de certains blocs à l'enfouissement pour éviter la propagation d'erreurs.

Cette amélioration n'a pas encore été implémentée dans l'algorithme actuel.

Réversibilité de l'enfouissement En stéganographie classique, on exploite la redondance du signal pour enfouir des informations : les données non-significative (LSB de composantes couleur sur une image non-compressée par exemple) sont donc purement et simplement remplacées par le message à transmettre et cette opération est totalement irréversible. Il existe cependant des domaines d'application dans lesquels il n'est pas possible d'altérer les données de cette manière, comme l'imagerie médicale, pour des raisons évidentes, ou l'imagerie dans un contexte militaire. Dans notre cas, on souhaite ne pas perturber les détails de régions d'intérêt, or cela conduit automatiquement à une diminution de la capacité d'enfouissement en utilisant des techniques de stéganographie classiques. Dans le domaine non-compressé, cette diminution n'est pas vraiment significative car la capacité reste très élevée. Par contre, elle est beaucoup plus gênante lorsqu'on travaille dans le domaine compressé, ou la quantité d'information est faible et très significative : il n'y a pas beaucoup de place pour enfouir des données et on ne peut pas le faire n'importe où.

C'est pour cela que des études ont été menées assez récemment sur des techniques réversibles d'enfouissement de données [19] [20] [21], c'est-à-dire qu'il est possible après extraction du message de restaurer complètement l'image originale. Malheureusement les techniques présentées n'étaient pas transposables à la vidéo compressée au format H.264 ; on a donc tenté de créer un schéma simple d'enfouissement réversible. Le principe part du constat suivant : en voulant garder l'information originale du médium-hôte et transmettre un message en plus par le même canal (ici, les coefficients quantifiés de transformée entière), on essaie de stocker deux informations là où il n'y a de la place que pour une seule. Il faut donc artificiellement créer un emplacement mémoire supplémentaire. On a proposé de doubler le coefficient choisi pour l'enfouissement puis de lui rajouter 1 ou 0 selon le bit à enfouir. Ainsi au décodage, on peut récupérer le bit du message et retrouver le coefficient original en divisant par deux. Cette approche a été implémentée et testée, mais elle ne s'est pas révélée concluante car du fait de la dynamique limitée des composantes de couleur des pixels, des artefacts visibles à l'œil nu étaient créés par saturation. Ces

artefacts, non contents d'être très facilement repérables (taches blanche ou de couleur sur l'image), permettent en sus de repérer graphiquement l'endroit exact de l'enfouissement, épargnant au stéganalyste la peine d'une étude statistique poussée. On a donc abandonné cette approche.

4.2.2 Améliorations apportées au schéma de Nguyen, Tay et Deng

Capacité d'enfouissement variable Cet algorithme a également été adapté pour pouvoir utiliser un paramètre de capacité par frame P, noté C_P. Un peu de la même manière que pour l'algorithme amélioré de Noorkami et Mersereau, on sélectionne aléatoirement C_P vecteurs de mouvement dans le champ de vecteurs de la frame P courante et on leur applique ensuite l'enfouissement.

Compensation de l'enfouissement sur les vecteurs voisins Cette étape, assez fastidieuse dans l'algorithme original, était essentielle pour éviter des distorsions visibles. En effet, les vecteurs de mouvement sont prédits à partir des vecteurs voisins et seules les erreurs de prédiction sont codées. Si celles-ci ne sont pas compensées après enfouissement, le vecteur final pointe donc sur une zone qui ne correspondra plus avec les résidus calculés au moment de l'estimation de mouvement. A la reconstruction, on ne récupèrera donc pas les données de départ.

On n'a pas implémenté l'étape de compensation car elle n'est pas nécessaire dans notre cas, puisqu'on travaille au niveau de l'encodeur, directement pendant la compression. La prédiction et le codage différentiel des vecteurs se fait donc de manière automatique par le codeur au cours de l'étape de compensation de mouvement. Les auteurs de l'article travaillaient dans une optique de *watermarking*, où le tatouage doit être appliqué sur une vidéo déjà compressée.

Modification du processus d'enfouissement L'opération de quantification, qui dans l'algorithme de base garantissait la synchronisation entre l'enfouissement et l'extraction du message, a été abandonnée au profit d'un système plus simple car elle était susceptible de modifier les composantes des vecteurs au pixel près, ce qui aurait perturbé l'algorithme d'analyse de mouvement du projet Infom@gic.

On a donc modifié le protocole stéganographique en passant de l'enfouissement de deux bits sur une composante avec quantification à un bit sur chaque composante sans quantification (le LSB est purement et simplement écrasé). La règle d'éligibilité à l'enfouissement pour un vecteur de mouvement a également été retravaillée pour refléter cette modification :

> si les valeurs des deux composantes du vecteur sont inférieures en valeur absolue à un seuil défini par l'utilisateur [5], il ne peut pas être sélectionné pour l'enfouissement.

On garde ainsi le principe expliqué dans [14] tout en l'adaptant aux contraintes spécifiques de notre projet. On remarquera de plus qu'on a un gain théorique très faible en débit (puisque l'erreur maximale en norme est de $\sqrt{2}$ contre 2) ainsi qu'en complexité (car on n'effectue pas la quantification des valeurs des composantes des vecteurs). De plus, on avait constaté des erreurs au décodage lors des tests de l'algorithme original, dûes au fait

5. Ce seuil peut évidemment agir sur la capacité d'enfouissement de la frame. Il a été fixé à 2 pour nos tests.

que la condition de sélection de la composante la plus grande n'était pas assez précise. Ces erreurs ont été corrigées par les modifications apportées.

4.3 Mise en place d'un schéma hybride d'enfouissement

4.3.1 Motivations et présentation

Arrivé à ce point, un constat demeure : certains points du cahier des charges ne peuvent pas être satisfaits par les solutions technologiques choisies, même après les améliorations apportées. Le schéma de Noorkami et Mersereau pourrait permettre d'atteindre la capacité ciblée sans augmentation de débit, mais au prix de dégradations visuelles, ce qu'on ne souhaite pas. De plus ce schéma, conçu à l'origine pour une application de *watermarking*, est adapté à de faibles capacités et en forçant la charge à enfouir, on risque d'augmenter fortement la détectabilité. L'autre schéma choisi ne crée pas de distorsion, mais est fragile et mène à une augmentation du débit car on ne choisit pas le vecteur de mouvement optimal, ce qui augmente l'énergie des résidus et handicape donc le codage entropique. Enfin, la charge par frame est actuellement fixée par un seuil. On ne tient donc pas compte des caractéristiques de la frame courante pour l'enfouissement, ce qui a aussi pour effet d'augmenter les scores de détectabilité.

On propose donc un schéma d'enfouissement hybride basé sur les deux algorithmes retenus. Les informations seront cachées à la fois dans les frames I, par modification des cœfficients quantifiés de transformée, et dans les frames P, en utilisant les LSB des valeurs des composantes des vecteurs de mouvement. Ce type d'architecture a déjà été décrit dans [22], où l'enfouissement d'une marque robuste dans les frames I et d'une marque fragile dans les frames P peut être applicable à la fois pour protéger des droits d'auteur ou pour authentifier des documents numériques.

On introduit une fonctionnalité supplémentaire à cette architecture hybride : l'allocation de débit d'enfouissement. On fixera un débit moyen à atteindre sur la durée d'un GoP, mais le débit d'enfouissment par frame pourra varier en fonction des caractéristiques locales du médium. Il sera contrôlé par le seuil de sélection des vecteurs de mouvement (voir section 4.2.2), que l'on fera varier en fonction de la capacité de la frame courante. Les informations relatives à ces changements de seuil sont nécessaires au décodage, elles doivent donc être protégées. On les enfouira dans la frame I, de manière robuste, ainsi que les données critiques des résultats de l'analyse. Le reste des données sera enfoui dans les frames P. De cette manière, on gère la protection des données critiques (gestion du type UEP — *Unequal Error Protection*) et l'enfouissement se fait de manière adaptative pour diminuer la détectabilité.

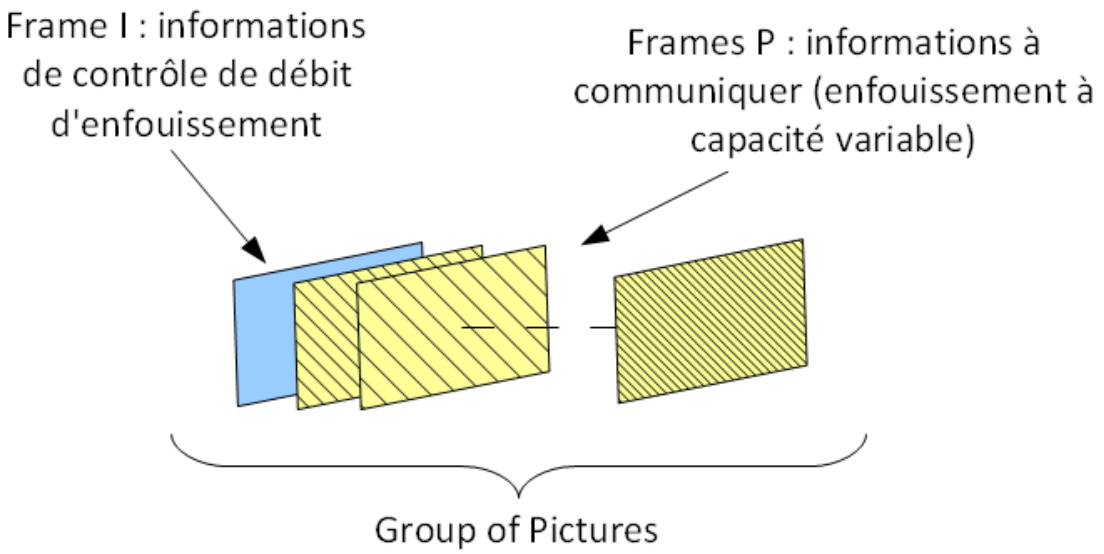

FIGURE 24 – Schéma hybride avec allocation de débit d'enfouissement.

4.3.2 Étude de faisabilité

Une étude a été menée pour vérifier la pertinence et la faisabilité de la démarche proposée. Les distributions des composantes verticales et horizontales des vecteurs de mouvement ont ainsi été examinées, ainsi que la dynamique temporelle des vecteurs, pour justifier l'allocation du débit à l'échelle du GoP. L'ensemble de test était composé de vidéos de différents types (plans fixes ou pas, objets mouvants...) à différents taux de compression.

Des constats ont ainsi pu être établis :

- la capacité moyenne sur les frames P, exprimée comme la proportion des vecteurs éligibles à l'enfouissement, est beaucoup plus importante sur les vidéos présentant du mouvement (objets en mouvement, arrière-plan mobile ou mouvements de caméras) ;
- l'augmentation du *framerate* a pour conséquence directe la diminution de cette capacité sur les frames P. Cela peut être expliqué facilement : le champ de vecteurs de mouvement est fortement corrélé au mouvement réel présent dans la vidéo, du moins à faible compression. En échantillonnant la vidéo avec une fréquence plus élevée, les vecteurs de mouvement d'une frame sur l'autre ont donc naturellement une dynamique moins importante. Néanmoins, cette baisse est largement compensée, entre termes de débit d'enfouissement, par le nombre plus élevé de frames par seconde ;
- l'augmentation de la compression, via le pas de quantification, a pour effet d'augmenter la capacité moyenne.

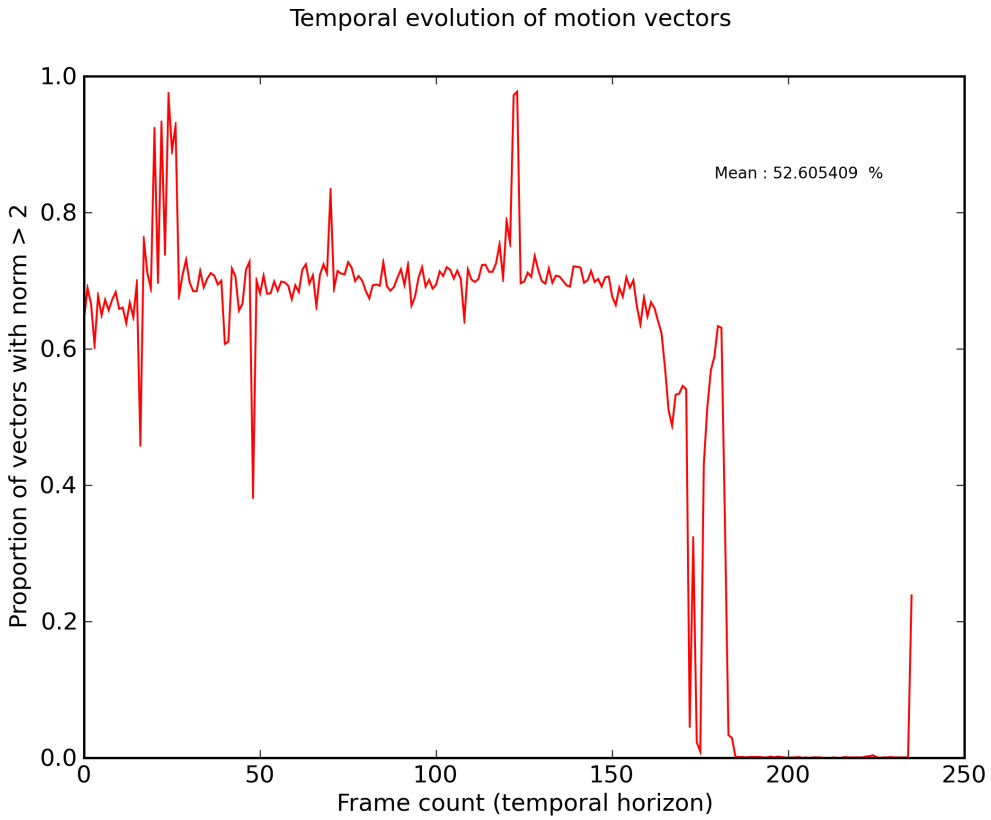

FIGURE 25 – Ce graphique représente l'évolution temporelle du pourcentage de vecteurs dont les valeurs des composantes sont supérieures en valeur absolue au seuil d'éligibilité à l'enfouissement, pour la vidéo *parkrun*. On peut aisément faire le lien entre le mouvement dans la vidéo et l'évolution de cette proportion : durant les deux premiers tiers de la séquence, on a un *travelling* horizontal. Les pics correspondent à des mouvements verticaux de la caméra. Le *travelling* s'arrête sur le dernier, avec un soubresaut au niveau de la frame 180, qui provoque le pic visible sur le graphique.

L'analyse de la dynamique des vecteurs de mouvement a mis à jour un fait intéressant : c'est la frame P suivant directement la frame I qui présente la plus grande capacité. Cela est dû à l'allocation de débit effectuée par le codeur pour compenser la charge de la frame I. On le voit graphiquement par des pics espacés régulièrement de la taille du GoP. Ce phénomène s'accentue lorsqu'on augmente la compression.

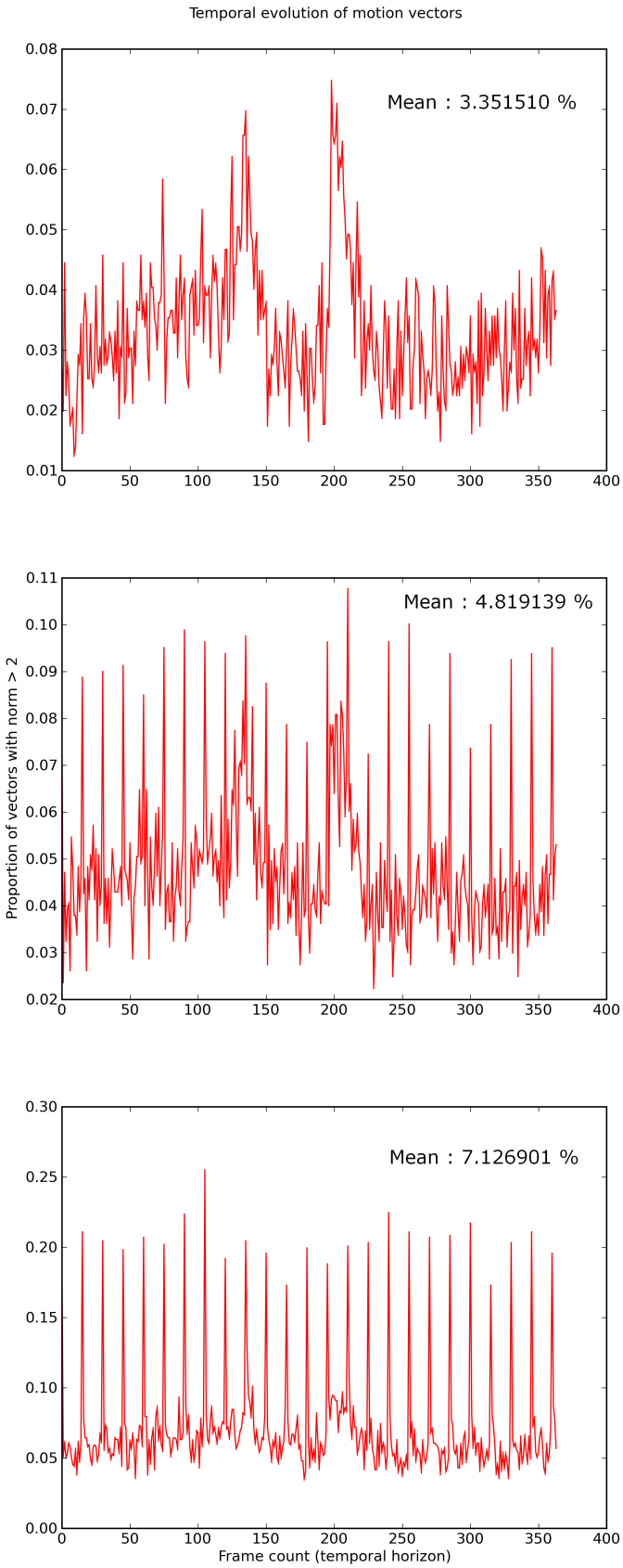

FIGURE 26 – L'évolution de la proportion de vecteurs éligible est ici représentée pour la vidéo *Turin* avec un pas de quantification variant, de haut en bas, de 20 à 40. On constate l'apparition des pics de capacité.

Ce phénomène est encore en cours d'étude ; de même, si l'algorithme hybride a été implémenté, la fonctionnalité d'allocation de débit d'enfouissement ne l'est pas encore. On atteint pour l'instant la consigne (en bits par frame) en compensant d'une frame sur l'autre une perte temporaire de capacité (voir figure ci-dessous). Cette méthode est simple mais est loin d'être optimale : on prête le flanc à la stéganalyse lorsqu'on se retrouve en défaut de capacité par rapport à la charge cible ou lorsque le défaut a été compensé et qu'on redescend brutalement à la charge cible. En effet, c'est dans ces deux situations que l'on est le plus détectable car on augmente les risques de dégradation (enfouissement à capacité maximale dans le premier cas) ou on fait varier violemment les descripteurs calculés pour la stéganalyse (deuxième cas).

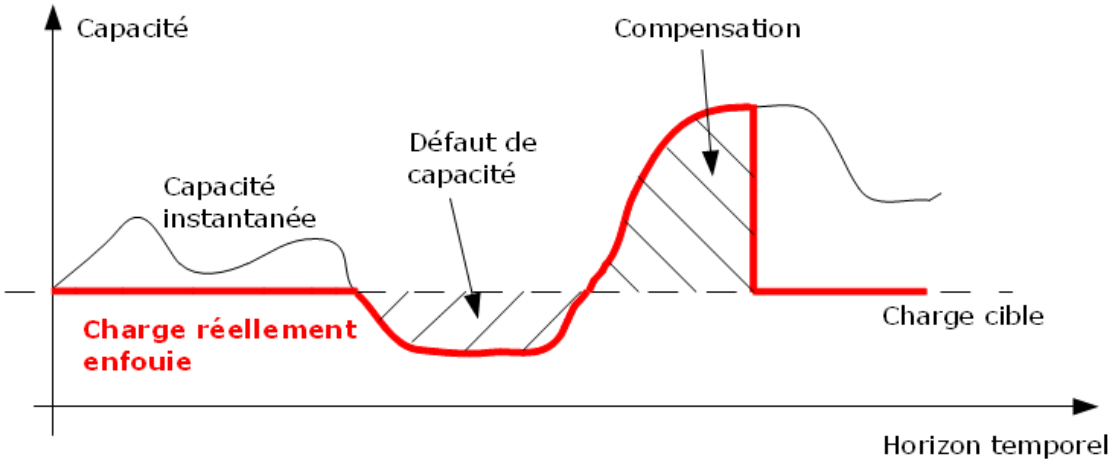

FIGURE 27 – Ce graphique représente l'évolution temporelle de la charge réellement enfouie, limitée par la capacité instantanée de la vidéo.

4.3.3 Inconvénients - Conclusion

La fonctionnalité d'allocation de débit introduit néanmoins un inconvénient : enfouir en fonction de la capacité locale des frames implique une analyse préalable de celle-ci. Une latence équivalente au paramètre de taille du GoP utilisé par le codec est inévitable. De plus, des remaniements profonds au niveau du codec sont à prévoir car le GoP doit être « bufferisé » alors que le processus de codage n'est pas achevé (en particulier, le codage entropique doit être retardé et appliqué en fin de processus d'enfouissement au GoP entier) pour pouvoir réaliser l'analyse de capacité à l'échelle du GoP. Or l'architecture actuelle du codec ne permet pas ce type de fonctionnement.

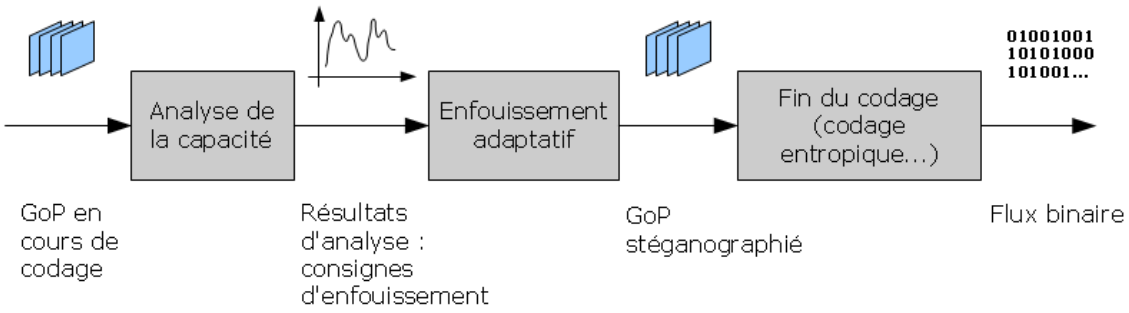

FIGURE 28 – Schéma représentant le détail du processus d'allocation de débit d'enfouissement. L'algorithme travaille au niveau du GoP.

Le schéma hybride a donc été implémenté, à l'exception de la fonctionnalité d'allocation de débit d'enfouissement, qui nécessite un changement d'architecture du codec. L'idée a cependant été jugée très intéressante et pourra faire l'objet de travaux futurs au sein du laboratoire.

4.4 Mise en place d'un *benchmark* de stéganalyse

Le planning établi en début de stage prévoyait que l'évaluation de la résistance à la stéganalyse de l'algorithme de stéganographie développé se déroulerait de la façon suivante :

- définition et réalisation d'un corpus de test ;
- reprise et approfondissement de l'étude bibliographique en se concentrant sur les techniques de stéganalyse ;
- identification de métriques pertinentes (qualité d'image...) et des outils de stéganalyse existants ;
- implémentation des métriques et codage d'un *framework* de test sous la forme d'un script (fonctionnement en mode batch).

Le but était d'obtenir un environnement de test automatique auquel on aurait fourni des vidéos et qui leur attribuerait une probabilité d'avoir été utilisées comme médium-hôte. On aurait alors comparé les résultats obtenus par les algorithmes de l'état de l'art et celui développé au cours du stage.

Malheureusement, alors que les logiciels de stéganographie, libres ou non et de qualité très variable, pullulent sur Internet, il n'existe pour ainsi dire pas de logiciels de stéganalyse vidéo. Les seuls dont on a eu connaissance sont soit très chers, soit commercialisés uniquement à des entités gouvernementales (forces de police, cellules anti-terroriste...). On a donc revu notre manière de procéder et plus de temps a été accordé à la recherche bibliographique pour pouvoir déterminer des techniques de stéganalyse adaptées à notre application qui soient implémentables avec les moyens et le temps disponible. En un sens, cela a été bénéfique car on en a retiré une connaissance plus riche que si l'on avait simplement utilisé des outils *off the shelf*.

La démarche générale suivie par les techniques de stéganalyse est la suivante :

1. Extraction de descripteurs pertinents : ce sont des métriques locales ou globales, dans le domaine spatial ou fréquentiel, qui doivent avoir un pouvoir discriminant vis-à-vis de l'impact de la stéganographie.
2. Application de méthodes de *data mining* : on cherche ensuite à établir une décision du type « médium stéganographié ou non » pour la vidéo analysée en se basant sur

les descripteurs précédemment calculés.

Cette démarche est assez standard et a été formalisée plus précisément il y a quelques années, notamment avec l'utilisation de SVM (voir [23]). Elle entre dans la catégorie « Apprentissage supervisé » de la classification établie dans [13] (voir section 3.2.3). On reviendra plus loin sur les techniques de *data mining* (voir section 4.4.3).

Remarque : On n'a pas effectué une véritable étude stéganalytique en aveugle, puisqu'on connaît en détail les schémas stéganographiques utilisés. Le choix des descripteurs est donc totalement biaisé, puisqu'il a été fait dans l'optique d'obtenir les résultats de détection les plus élevés possibles.

4.4.1 Choix de descripteurs adaptés au problème

Descripteurs issus de la bibliographie Parmi les différentes techniques de stéganalyse rencontrées dans l'état de l'art, on en a sélectionné principalement deux, en se basant sur un critère de pertinence (est-ce que la méthode de stéganalyse est adaptée à la méthode de stéganographie employée ?) et un critère de rapidité et de facilité d'implémentation (on a donc évité les techniques s'appuyant sur un nombre trop important de descripteurs, comme [24]).

Stéganalyse du schéma de Noorkami et Mersereau Pour la stéganalyse du schéma d'enfouissement sur les frames I, une technique basée sur une attaque par collusion temporelle, travaillant dans le domaine pixélique, a été choisie (voir [25] et [26]). L'idée développée par les auteurs est la suivante : il s'agit d'exploiter la redondance temporelle présente dans les séquences vidéo. Pour cela, une estimation du médium d'origine est construite grâce à une attaque par collusion, issue du *watermarking* : plusieurs images tirée de la séquence sont utilisées pour retirer la marque, c'est-à-dire judicieusement réduire son énergie par rapport à celle du médium-hôte. L'attaque linéaire est un cas particulier de collusion qui représente une simple opération de moyenne pondérée sur les frames sélectionnées. Le résultat sera d'amplifier les zones similaires et d'atténuer les détails. De cette manière, on espère garder le contenu significatif d'un point de vue perceptuel dans la vidéo, en gommant les détails, dont la marque fait a priori partie. Ce genre d'attaque est utilisée pour retirer les tatouages numériques et favoriser la copie illégale de contenus protégés par les lois sur la propriété intellectuelle.

Une fois cette estimation du médium d'origine calculée, on peut récupérer une estimation de la marque par soustraction avec le médium testé. Une analyse statistique de ses caractéristiques est ensuite réalisée et une décision est prise grâce à un classifieur (dans l'article, un simple classifieur k plus proches voisins – *K-NN*). Les auteurs formulent l'hypothèse raisonnable que la distribution de la marque est gaussienne. Les descripteurs extraits sont donc les suivants : kurtosis, entropie et premier quartile. En effet, le paramètre de kurtosis d'une gaussienne vaut 3 et est assez différent pour d'autres types de distribution et à variance égale, l'entropie d'une distribution gaussienne est maximale.

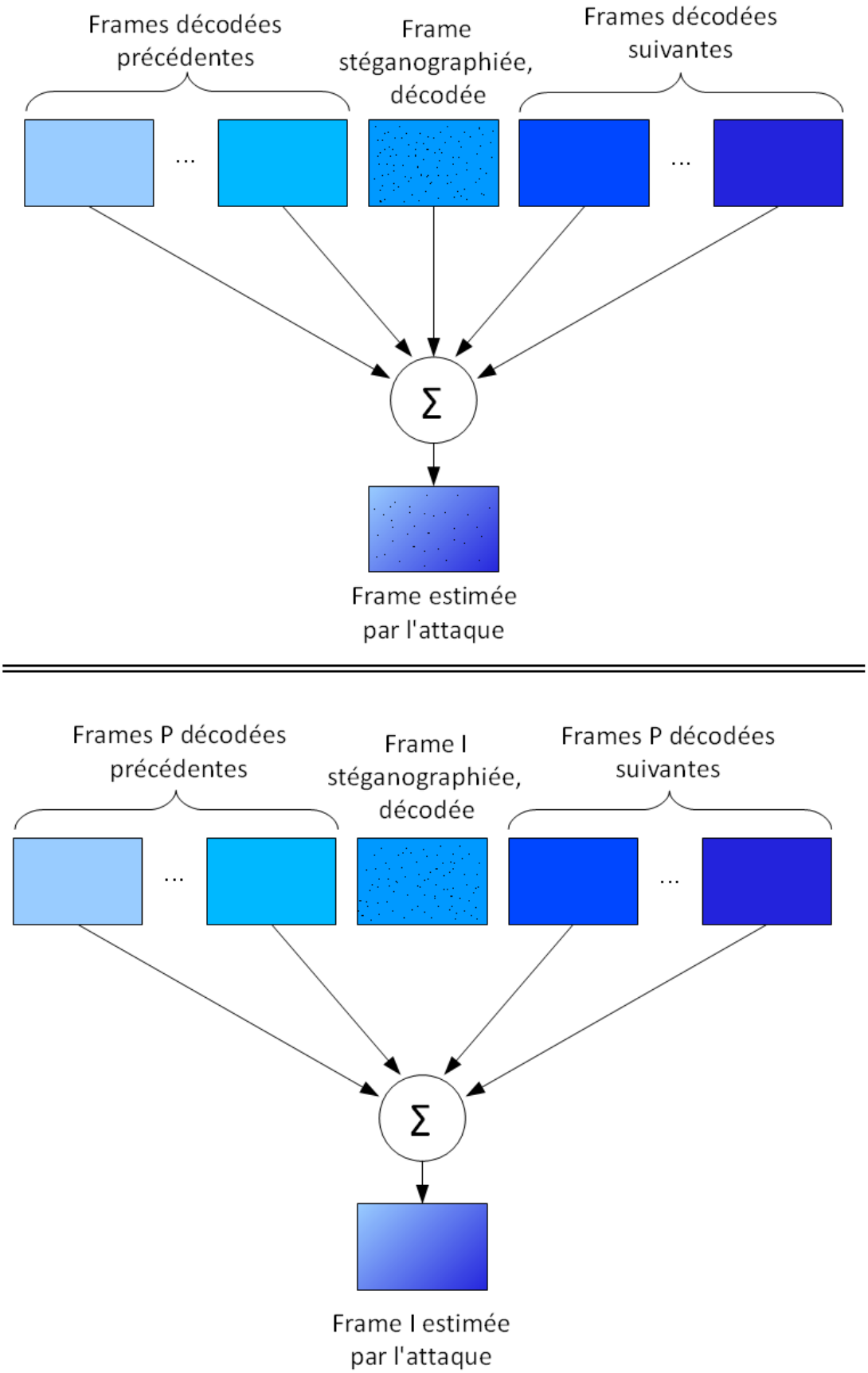

FIGURE 29 – Ce schéma représente, en haut, l'attaque décrite dans [25]. L'attaque modifiée en exploitant la connaissance de l'algorithme d'enfouissement sur les frames I est présenté en dessous.

L'algorithme décrit dans l'article a été jugé adapté pour la stéganalyse du schéma de Noorkami et Mersereau. On l'a cependant modifié au niveau de l'attaque par collusion, pour augmenter son efficacité : comme l'enfouissement des données ne se fait que sur les frames I, on calcule une estimation de ces frames uniquement, en faisant une moyenne sur une fenêtre temporelle d'une demi-largeur L (fixée à 1 dans nos expériences) mais en sautant la frame I ; en d'autres termes, on estime la frame par les frames environnantes. Comme il n'y a pas d'enfouissement sur celles-ci, elles ne présentent pas le bruit rajouté par la marque ; l'estimation de la frame I du médium original sera donc plus correcte. D'autre part, la possibilité d'utiliser des coefficients de pondération dans la moyenne a été rajoutée.

Stéganalyse du schéma de Nguyen, Tay et Deng Pour les frames P, on s'est davantage intéressés à la distribution du champ de vecteurs de mouvement, comme expliqué dans [27]. Les auteurs partent du constat qu'il existe certaines propriétés statistiques dans les caractéristiques du champ de vecteurs de mouvement d'une séquence vidéo compressée en exploitant la redondance temporelle de l'information : les valeurs des composantes des vecteurs de mouvement voisins (à faible distance) sont corrélés. En ajoutant un bruit indépendant du médium d'origine aux composantes, l'enfouissement perturbe cette propriété de corrélation.

Un opérateur différentiel est alors défini sur les vecteurs de mouvement et plusieurs descripteurs sont calculés à partir de son histogramme. Sur un médium non stéganographié, la distribution des valeurs de l'opérateur différentiel est une gaussienne centrée en zéro et de très faible variance. Sur un médium stéganographié, l'enfouissement, en rompant la corrélation, va faire augmenter la variance de la distribution. L'article présente d'autres descripteurs calculés à partir de l'histogramme des valeurs du gradient, mais on n'a retenu que la variance dans les deux directions pour des raisons de temps principalement.

Descripteurs développés au cours du stage Comme on avait une connaissance approfondie des algorithmes d'enfouissement utilisés ainsi que des conséquences de l'enfouissement en termes de distorsion, on a développé, parallèlement aux méthodes de l'état de l'art, des descripteurs spécifiques.

Descripteurs fréquentiels globaux Pour la stéganalyse des frames I, on s'est intéressés à la transformée de Fourier discrète à deux dimensions d'une frame complète car on souhaitait repérer les caractéristiques fréquentielles de la marque dans le spectre. L'idée est la suivante : comme la marque peut être considérée comme un bruit additif gaussien, l'enfouissement provoque un déplacement du centroïde spectral. De plus, le schéma de Noorkami et Mersereau, dans sa version originale, enfouit des données dans les hautes fréquences. En étudiant l'énergie dans une bande de fréquence correspondant à l'enfouissement d'un bit sur les 8 derniers coefficients d'un bloc 4×4, on devrait pouvoir repérer l'influence de l'enfouissement de manière globale sur la frame.

	Hall			Speedway			Torino		
	A	B	C	A	B	C	A	B	C
Énergie	0,012	0,0121	0,0131	0,00437	0,00437	0,00474	0,00394	0,00396	0,00421
Centroïde spectral (composante horizontale)	0,381	0,384	0,419	0,237	0,237	0,263	0,204	0,206	0,227
Centroïde spectral (composante verticale)	1,048	1,057	1,147	0,768	0,768	0,833	1,385	1,391	1,478

TABLE 1 – Résultats des descripteurs fréquentiels globaux pour trois vidéos, testées dans trois situations (A, B et C, correspondant respectivement à : vidéo non stéganographiée, vidéo stéganographiée à 1 bit/macrobloc, vidéo stéganographiée à 16 bits/macrobloc). On a utilisé pour l'enfouissement le schéma de Noorkami et Mersereau original, en sélectionnant tous les macroblocs des frames I. Seules celles-ci ont été analysés pour le calcul des descripteurs.

L'étude a été menée et on a effectivement observé un déplacement du centroïde spectral vers les hautes fréquences et une augmentation de l'énergie moyenne, ce qui cadrait avec les attentes et la théorie. Cependant ces différences étaient trop faibles pour des consignes d'enfouissement réalistes (situation B sur le tableau), pour être exploitables.

Descripteurs fréquentiels locaux Comme l'approche globale n'a pas permis de trancher pour la stéganalyse des frames I, on s'est penché sur une approche locale. En modifiant un coefficient pour enfouir un bit, le schéma de Noorkami et Mersereau perturbe la distribution des coefficients de transformée. Ce n'est pas flagrant si cela est fait sur les coefficients non-nuls, mais ça l'est plus sur les coefficients quantifiés de haute fréquence qui sont typiquement nuls. Il se trouve qu'une thèse ([28]), dont l'auteur a collaboré avec des membres du laboratoire, explore ce sujet et propose une modélisation de la distribution en sous-bandes des coefficients quantifiés de transformées entières sur les frames I. Un stage effectué précédemment au sein du laboratoire ([29]) a approfondi le sujet et recense deux modèles de distribution des coefficients de transformée :

- le modèle de Cauchy, plus proche de la distribution réelle, mais peu utilisé car il ne permet pas d'obtenir une expression simple dans les calculs de débit et de distorsion ;
- le modèle de Laplace, déjà présenté dans [28].

Les deux distributions sont exprimées comme suit :

$$f_{Cauchy}(x) = \frac{1}{\pi}\frac{\mu}{\mu^2 + x^2}$$
$$f_{Laplace}(x) = \frac{\lambda}{2}\exp(-\lambda.|x|)$$

FIGURE 30 – Densités de probabilité de Cauchy, à gauche, et de Laplace, à droite (source : [29]).

L'idée est alors de garder l'attaque par collusion décrite dans [25] et modifiée comme expliqué précédemment et de comparer les distributions avant et après attaque de la manière suivante :

$$\Delta_k = (\sum_{i=-N}^{N} (p_{avant}(i) - p_{après}(i))^2)^{\frac{1}{2}}$$

où Δ_k est la mesure d'écart entre les distributions p_{avant} et $p_{après}$ pour la sous-bande fréquentielle k ($\forall k \in [1, 15]$). Les Δ_k sont moyennés sur les frames I de la séquence vidéo analysée et servent de nouveaux descripteurs.

À l'heure actuelle, ces descripteurs ont été simplement formalisés et n'ont pas été implémentés. Ils seront testés durant le dernier mois du stage, si le temps le permet.

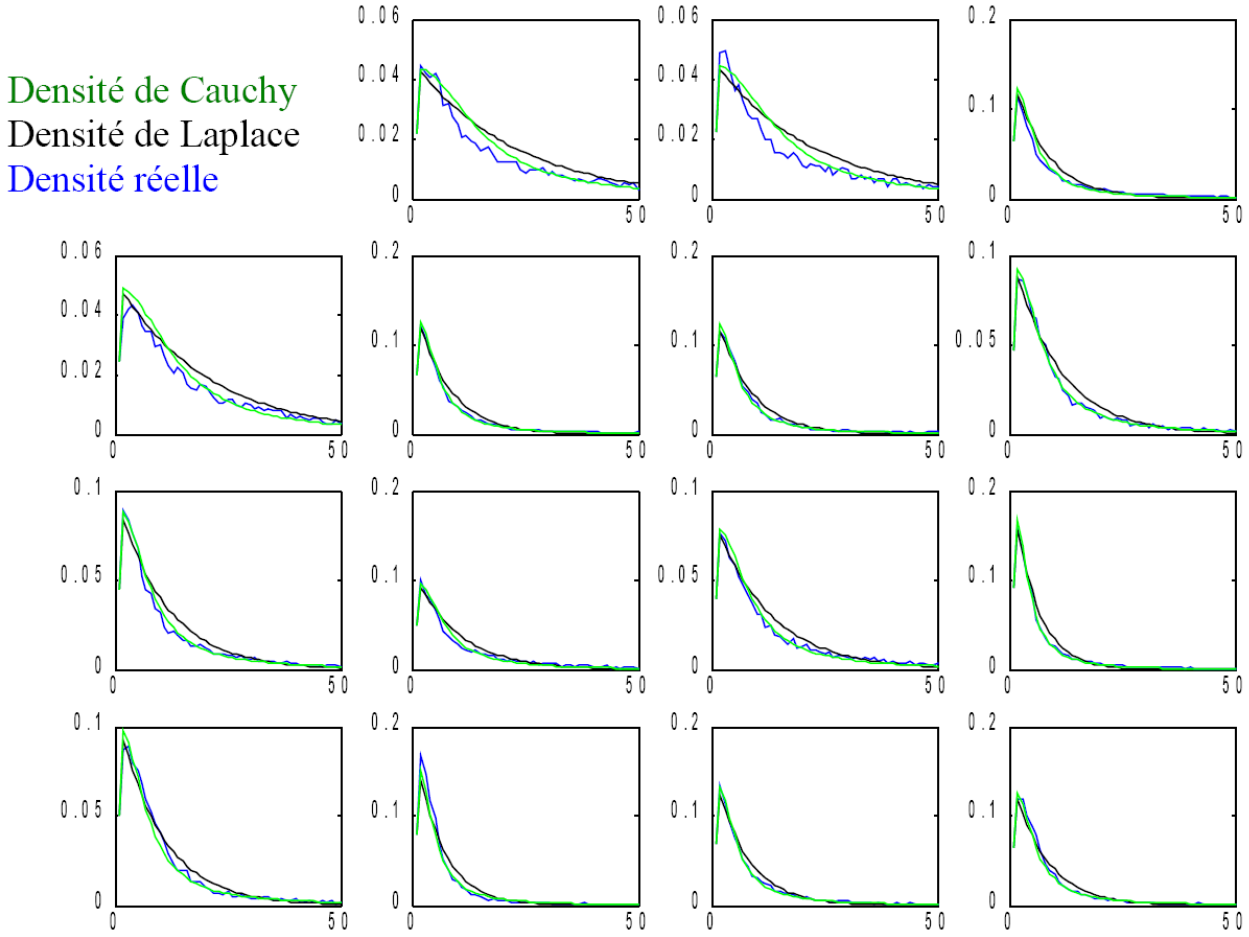

FIGURE 31 – Comparaison de la distribution des coefficients AC de transformée entière (en valeur absolue) avec les modèles de Cauchy et de Laplace. On voit bien que les coefficients des sous-bandes de fréquence spatiale élevée ont une dynamique plus faible qu'en basse fréquence spatiale (source : [29]).

4.4.2 Construction du *benchmark*

On a adopté une démarche de *batch* automatisé (traitement par lots des calculs) : on ne fournit au script que le nom d'une liste de vidéos. Chaque vidéo est encodée puis décodée plusieurs fois en utilisant des options différentes (pas de stéganographie, stéganographie en utilisant une méthode donnée, à une charge donnée...) avec des efficacités de compression différentes (valeurs du pas de quantification utilisées : 20, 30 et 40). Certaines métriques (PSNR, débit, capacité...) sont extraites et enregistrées sur fichiers pendant l'étape de codage/décodage. On calcule ensuite les descripteurs et on stocke les résultats sur le disque, dans différents fichiers. Le script est programmé en Python, les opérations lourdes (encodage, décodage, attaque par collusion...) étant programmées en C ; le script appelle donc des exécutables avec les paramètres voulus. Les résultats des calculs sont stockés de manière brute et seront exploités plus tard. Certaines mises en forme (graphes par exemple) sont cependant effectuées dans la foulée, ne demandant pas d'intervention humaine.

Le processus complet est décrit sur le schéma ci-dessous :

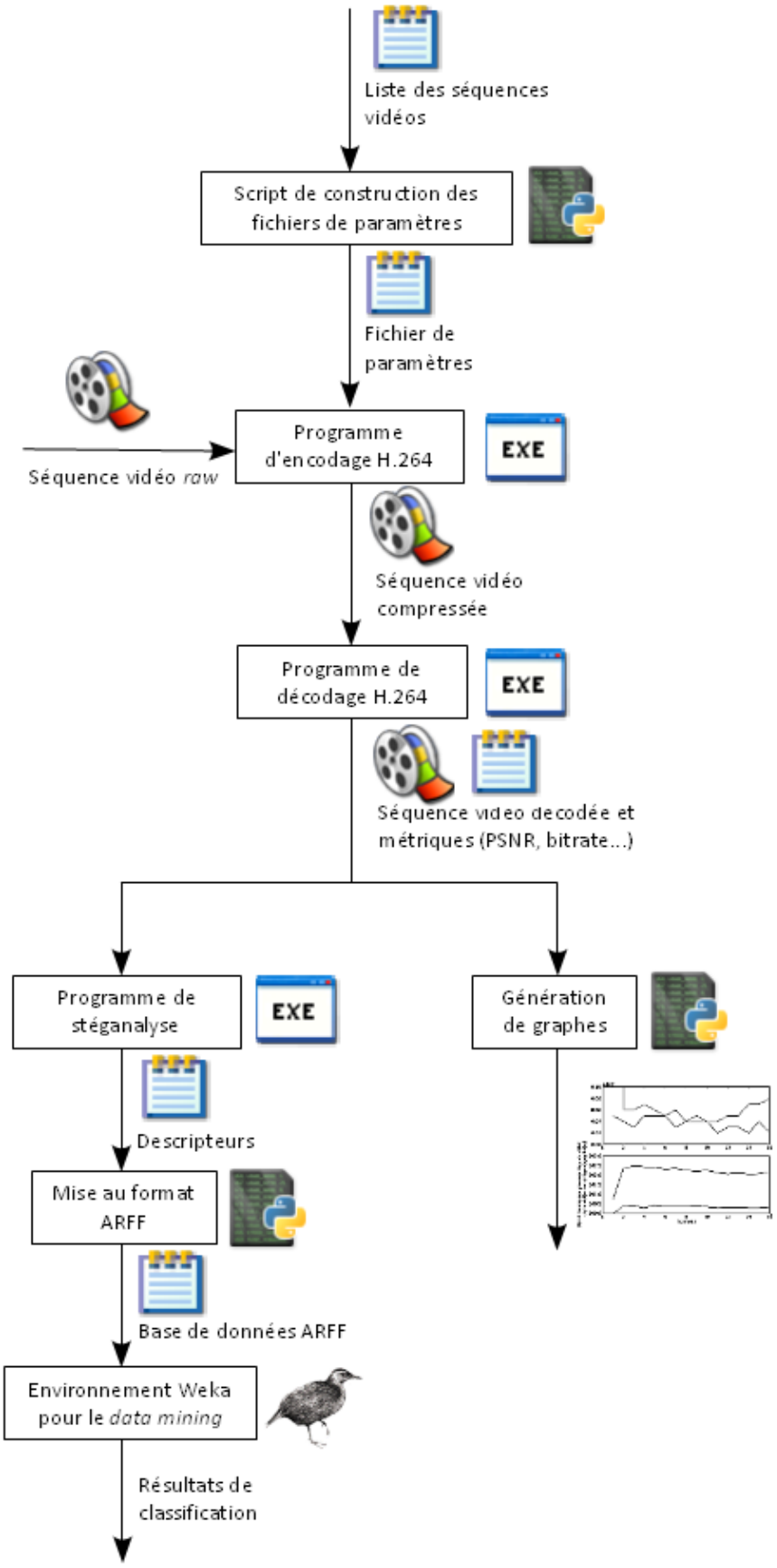

FIGURE 32 – Structure du *benchmark* de stéganalyse.

4.4.3 Exploitation des résultats

On a fait appel à des techiques de « fouilles de données » (*data mining*) pour exploiter les informations fournies par l'environnement de stéganalyse décrit ci-dessus. D'après [30], le *data mining*, à la croisée des mondes des statistiques, de l'intelligence artificielle et des bases de données, est un processus d'analyse dont l'approche est différente de celle utilisée en statistique. Cette dernière présuppose en général que l'on se fixe une hypothèse que les données permettent ou non de confirmer. Au contraire, le *data mining* adopte une démarche sans a priori (approche pragmatique) et essaie ainsi de faire émerger, à partir des données brutes, des inférences que l'expérimentateur peut ne pas soupçonner, et dont il aura à valider la pertinence.

Plusieurs environnements *open source* de *data mining* sont disponibles sur Internet. Nous avons choisi de travailler avec Weka [31], qui est le plus complet au niveau des algorithmes et le plus mature, puisqu'il propose des outils de visualisation, une interface graphique et des routines de pré-traitement des données. Les données doivent lui être fournies au format propriétaire ARFF, qui est un format CSV légèrement amélioré ; il est donc facile de formater correctement nos données.

4.5 Tâches annexes

Au cours du stage, certaines parties du travail effectué n'étaient pas directement en relation avec le domaine de la stéganographie. En particulier, on a travaillé sur le codec H.264 du laboratoire, principalement sur les deux aspects :

Algorithme d'estimation de mouvement : l'estimation de mouvement, précise alors au pixel près, a été raffinée au quart de pixel. Le niveau sub-pixélique est atteint par interpolation [6] et une technique dite de *Diamond Search* permet de trouver la meilleure estimation au sens de l'erreur quadratique moyenne. Le vecteur de mouvement est construit en deux étapes, d'abord avec une précision au demi-pixel, qui est ajoutée au vecteur déjà donné par l'algorithme de base, puis au quart de pixel. Cette fonctionnalité n'était pas implémentée dans la codec du laboratoire et était nécessaire pour l'utilisation du schéma de Nguyen, Tay et Deng travaillant sur les vecteurs de mouvement.

Refonte d'une partie du code : il s'agit plutôt d'une démarche de génie logiciel. Le code a été réorganisé car on avait besoin d'une granularité plus fine des fonctions pour utiliser le module d'enfouissement. Certaines fonctions ont d'autre part été recodées pour des raisons d'optimisation.

6. La librairie IPP d'Intel (*Integrated Performance Primitives*) a pour cela été utilisée. Il s'agit d'une librairie optimisées de fonctions d'usage courant en traitement de données et en calcul scientifique.

5 Évaluation des performances

Nota Bene Tous les résultats ont été obtenus à partir d'un corpus vidéo et du codec propriétaire développé au sein du laboratoire. Les options d'encodage utilisées sont les suivantes :

- pas de quantification identique sur les frames I et les frames P ;
- prédiction intra-image dans les frames P désactivée ;
- GoP de la forme IP (prédiction bidirectionnelle désactivée) avec référence limitée à la frame précédente.

5.1 Schéma de Noorkami et Mersereau amélioré

Beaucoup de modifications ont été apportées au schéma original. On a comparé leurs effets sur le plan de la capacité, du débit et de la distorsion : des données ont été enfouies avec une consigne proche de la capacité maximale (1300 bits par frame I) dans la vidéo *576i25_parkrun_ter.yuv* (voir annexe B). Les résultats sont présentés ci-dessous.

Remarque La consigne de capacité de 1300 bits par frame I ne correspond pas à un cas réel d'utilisation ; elle est beaucoup plus élevée que la cible du cahier des charges (200 bits par frame, voir section 2.1). Cependant, en exagérant les effets négatifs de l'enfouissement, elle permet de valider la démarche engagée.

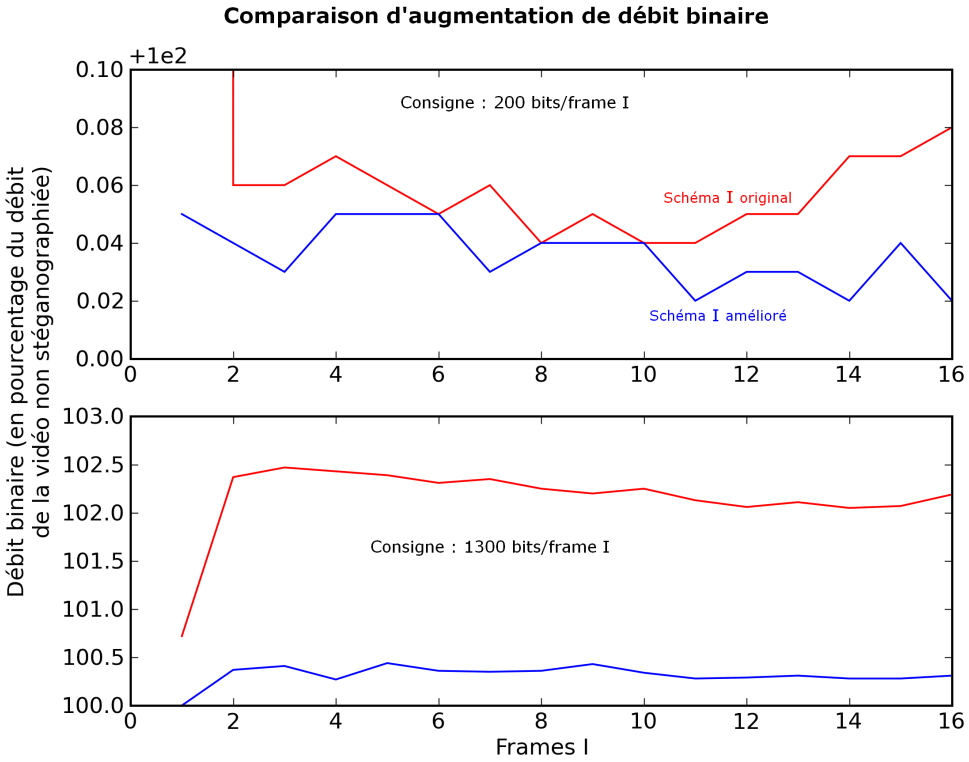

FIGURE 33 – Augmentation du débit binaire causé par l'enfouissement avec un pas de quantification de 20 (faible compression, graphe du haut) et de 40 (forte compression, graphe du bas) ; la courbe rouge correspond au schéma original et la bleue, au schéma amélioré.

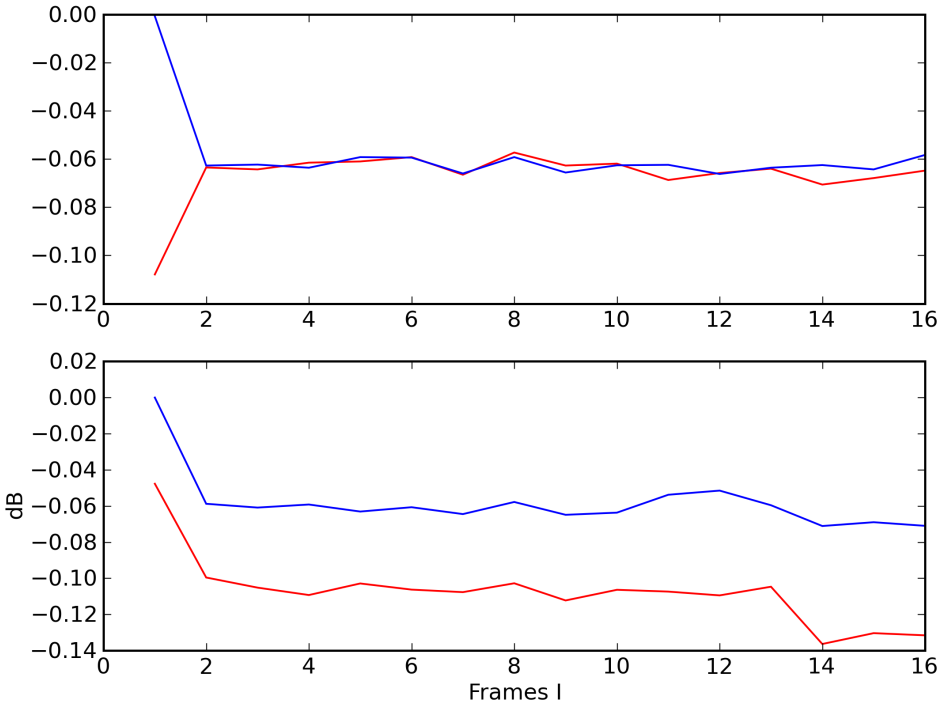

FIGURE 34 – Diminution de la qualité : pour chaque frame I, on a évalué la différence entre le PSNR du médium de couverture et celui du médium stéganographié.

	$QP = 20$	$QP = 40$
Schéma original	19874	14081
Schéma amélioré	18963	8194

TABLE 2 – Nombre de bits enfouis dans la vidéo (consigne : 20800 bits).

On voit donc d'après les résultats que les améliorations apportées ont pour effet de :

Limiter le débit binaire. L'augmentation moyenne de taille pour une frame I est de 424 octets pour l'ancien schéma et de 111 octets pour le nouveau, en utilisant un pas de quantification de 40 ; elle est respectivement de 113 et 69 octets pour un pas de 20. La taille moyenne d'une frame I est de 200 Ko pour un pas de 20 et de 30 Ko pour un pas de 40. Le fait que le gain soit plus important en utilisant une compression plus forte s'explique facilement : la proportion de cœfficients nuls augmente avec la compression. Le codage entropique est donc d'autant plus perturbé par le schéma original que le pas de quantification augmente. L'amélioration apportée est conçue pour éviter ce phénomène. L'augmentation de débit avec le nouveau schéma est donc logiquement plus faible. Il faut cependant ramener le résultat à la charge moyenne réellement enfouie : d'après le tableau ci-dessus, elle est de 155 octets pour l'ancien schéma et de 148 pour le nouveau avec un pas de quantification de 20 ; et elle est respectivement de 110 et 64 octets pour un pas de 40.

Limiter la distorsion. La qualité est moins affectée par le schéma amélioré, comme l'attestent les courbes de la figure 34. On peut objecter que la perte de qualité est très faible (au maximum -0.1 dB), il faut cependant garder à l'esprit que l'enfouissement

a pour objectif de ne pas être visible, ce qui explique les très faibles différences de PSNR. De plus, même si la consigne de 1300 bits par frame I est considérée comme élevée dans un cadre de stéganographie, elle ne représente que 0,5% de la taille en mémoire d'une frame I en résolution 4CIF.

Diminuer la capacité. En effet, on atteint la capacité maximale au cours du test et celle-ci est moins élevée pour le schéma amélioré que pour le schéma original. On reste cependant bien au delà des 200 bits par frame du cahier des charges , cela n'est donc pas gênant.

5.2 Schéma complet

Les résultats du schéma d'enfouissement hybride utilisant les schémas de Noorkami et Mersereau et de Nguyen, Tay et Deng améliorés sont présentés ci-dessous. On a implémenté un schéma hybride basé sur les schémas originaux pour pouvoir établir une comparaison non biaisée. Quand on parle de schéma original ou amélioré dans ce qui suit, il faut donc comprendre « schéma hybride utilisant les algorithmes originaux » et « schéma hybride utilisant les algorithmes améliorés ».

5.2.1 Non-perturbation des ROI

L'évaluation selon ce critère du module développé n'a pas pu être conduite car on n'a pas pu disposer des outils nécessaires au niveau du laboratoire. Le choix des solutions technologiques a cependant pris en compte cet aspect.

5.2.2 Capacité stéganographique

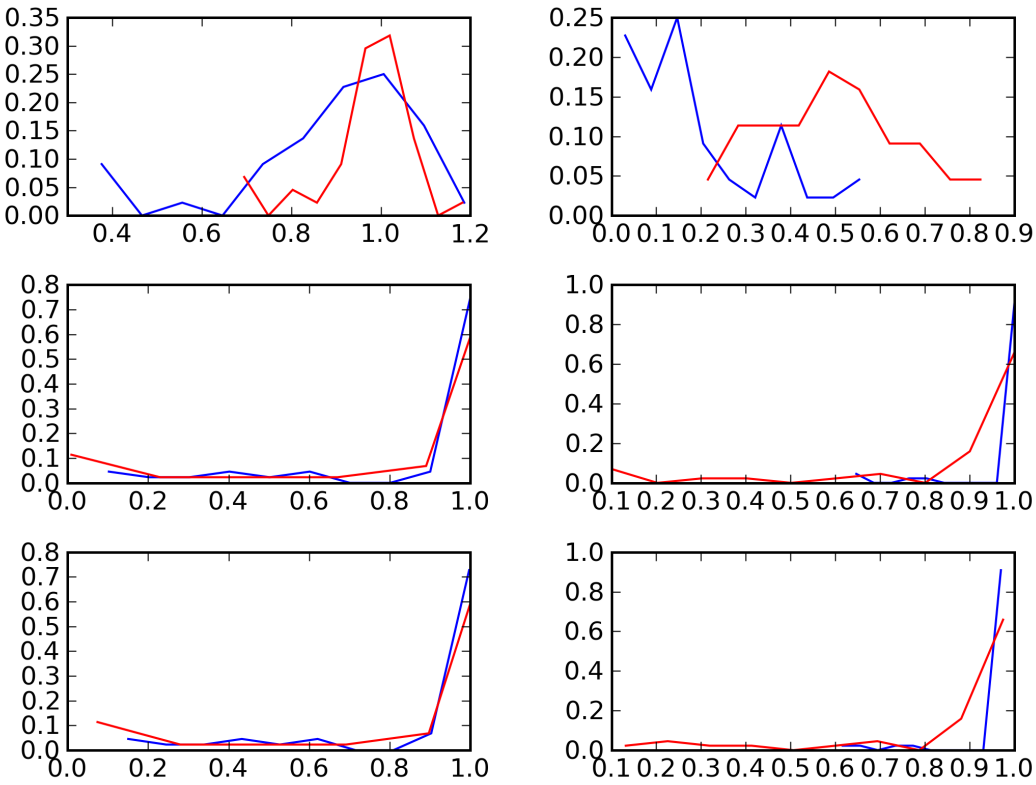

FIGURE 35 – Histogrammes de la charge enfouie en pourcentage de la charge cible sur le corpus vidéo (première ligne : charge sur les frames I ; deuxième ligne : charge sur les frames P ; troisième ligne : charge totale ; première colonne : $QP = 20$; deuxième colonne : $QP = 40$; en rouge : schéma original ; en bleu : schéma amélioré).

	$QP = 20$	$QP = 40$
Schéma original	76%	84%
Schéma amélioré	86%	92%

TABLE 3 – Charge enfouie en pourcentage de la consigne (200 bits par frame).

On constate une hausse de la capacité globale avec l'augmentation du pas de quantification, or les résultats précédents montraient justement une chute de la capacité des frames I. L'analyse du champ de vecteurs de mouvement met en évidence le fait que l'augmentation de la compression crée des aplats de couleurs assez larges, qui ont pour effet d'augmenter la dynamique des composantes des vecteurs. La perte de capacité des frames I est donc largement compensée par l'augmentation de capacité des frames P lorsque la compression est plus forte, ce qui explique les résultats plus élevés.

D'autre part, le schéma amélioré a de meilleures performances que le schéma original alors qu'on avait vu que de manière générale, les améliorations apportées au schéma de Noorkami et Mersereau avaient pour effet de diminuer la capacité d'enfouissement. Il se trouve que les conditions d'enfouissement (éligibilité des vecteurs) sur le schéma de Nguyen, Tay et Deng original étaient assez restrictives et sont plus souples dans le schéma original, résultant automatiquement en une augmentation de la capacité maximale.

Enfin, on a analysé en détail les résultats pour chaque vidéo du corpus. Deux types de vidéos présentent des mauvais résultats et font donc chuter la moyenne :

- les vidéos de faible résolution (CIF). Il est évident que si la même séquence a été échantillonnée à deux résolutions différentes, la séquence de plus faible résolution présentera une capacité plus faible ;
- les vidéos présentant peu de mouvement, typiquement issues de caméras de vidéo-surveillance. Les séquences présentant des mouvements amples de caméra (*travelling*, *zoom*...) ou de larges objets en déplacement rapide ont des champs de vecteurs de norme plus élevée en moyenne (on notera d'ailleurs que la diminution du *framerate* augmente naturellement la dynamique des vecteurs de mouvement).

5.2.3 Résistance à la stéganalyse

Comme expliqué plus haut, les résultats ont été obtenus avec l'environnement Weka, à partir des descripteurs extraits par les algorithmes de stéganalyse qu'on avait implémentés. Les pré- et post-traitements classiques ont été appliqués, notamment un brassage aléatoire des données et une validation croisée à 10 plis (*10 folds cross-validation*). Plusieurs méthodes ont été testées, majoritairement des classifieurs basés sur un regroupement en *clusters* et des SVM basés sur des noyaux à base radiale.

Pour les schémas original et amélioré, les résultats de bonne classification sont au maximum de 50% pour un ensemble de test comprenant autant de vidéos stéganographiées que de couverture, ce qui veut dire que le classifieur ne fait pas mieux qu'une décision au hasard. Il peut y avoir deux explications à cela :

- soit le classifieur ou les descripteurs ne sont pas adaptés ; cela semble peut probable cependant vu qu'ils sont spécifiquement choisis pour détecter les techniques d'enfouissement utilisées ;
- soit la charge enfouie est trop faible pour perturber significativement la statistique de la vidéo ; c'est plus probable vu que la charge stéganographique représente environ un millième de la taille du médium.

On ne dispose actuellement pas d'une réponse définitive à cette question. L'implémentation de descripteurs supplémentaires permettra peut-être de mieux discriminer les données.

5.2.4 Erreurs au décodage

Le schéma amélioré ne présente aucune erreur au décodage d'après les tests effectués sur l'ensemble du corpus ; on avait en effet accordé une attention particulière aux problèmes de synchronisation entre codeur et décodeur. Par contre, le schéma original présente ces problèmes ; on n'a pas cherché à calculer un taux d'erreur puisque le décodeur se désynchronise : lorsqu'un « effacement au codeur » se produit, tous les bits suivants sont faux et on ne peut reprendre le décodage qu'en début de frame suivante si des signaux de synchronisation ont été introduits dans les données (protocole de type paquétisation).

Les améliorations apportées aux schémas de base sont donc décisives de ce point de vue.

5.2.5 Complexité

	$QP = 20$	$QP = 40$
Charge de 5 Kbits/s	0.3%	0.2%
Charge de 32.5 Kbits/s	1.3%	0.4%

TABLE 4 – Augmentation du temps de codage causée par l'enfouissement, en pourcentage par rapport au temps de codage seul.

	$QP = 20$	$QP = 30$	$QP = 40$
Charge de 5 Kbits/s	-0.5%	0.5%	2.7%
Charge de 32.5 Kbits/s	0.1%	0.9%	3.8%

TABLE 5 – Augmentation du temps de décodage causée par l'extraction du message enfoui, en pourcentage par rapport au temps de décodage seul.

Pour la charge de consigne (5 Kbits/s), l'augmentation du temps de codage causée par l'enfouissement est donc minime (de l'ordre du millième). Le cahier des charges est donc respecté puisqu'on ne perturbera pas le caractère temps-réel du codec. Ces résultats sont rendus possibles par le fait que l'enfouissement du message est réalisé au sein du codeur ; on n'a donc pas besoin d'une étape de *parsing* avant de réaliser l'enfouissement proprement dit.

L'augmentation du temps de décodage dûe à l'extraction du message enfouie est également minime. Le temps de décodage est globalement le même, les scores inférieurs à 1% étant du même ordre de grandeur que l'imprécision de mesure. Le choix d'algorithmes de faible complexité est donc conforté par les performances présentées ici : en effet le décodage implique beaucoup moins d'opérations complexes que le codage ; le fait que l'utilisation du module de stéganographie n'affecte pas trop le temps de décodage (et donc encore moins le temps de codage) montre que l'on a atteint l'objectif de complexité.

6 Méthodes de travail

Cette partie expose brièvement les protocoles et méthodes mis en place au cours du stage dans un but d'efficacité de l'effort développé.

6.1 Méthodes scientifiques

6.1.1 Démarches générales

Disposant d'un temps limité, des processus permettant de faciliter la prise de décision ont été nécessaires. Par exemple, l'étape de recherche bibliographique devant déboucher sur une sélection d'un nombre restreint de techniques de l'état de l'art à implémenter et tester, on a produit des notes synthétiques sur les articles les plus intéressants, contenant un descriptif bref de la méthode employée, une analyse critique des éventuels défauts, une évaluation systématique selon les critères du cahier des charges et une estimation du temps d'implémentation. Lorsque des méthodes ou de nouvelles fonctionnalités ont été proposées, une étude de faisabilité a été préalablement menée, incluant des tests numériques, une éventuelle recherche bibliographique et un bilan, présentant les résultats et servant de base à la décision du GO - NO GO. Enfin, les algorithmes implémentés ont été validés sur des ensembles de test les plus larges possibles pour détecter les erreurs de réalisation ou de conception, en utilisant des métriques pertinentes.

6.1.2 Génie logiciel

Approches objets en C Le monde des systèmes embarqués, auquel appartient le laboratoire MMP, utilise encore le langage *C*, essentiellement pour des raisons de compatibilité matérielle. Cependant il est dommage d'abandonner les apports de l'approche orientée objet, dont le *C++* est l'un des langages le plus représentatifs. Des méthodes permettant de continuer à utiliser des schémas de pensée et de conception adaptés à l'approche objet existent donc, sans pour autant nécessiter de réimplémenter *Objective-C* ou *C++*. Elles sont utilisées par les développeurs et ingénieurs du laboratoire, qui se sont inspirés notamment de l'ouvrage de Schreiner [32]. Les structures ainsi que les pointeurs de fonctions, bien utilisés, permettent de retrouver un peu de l'aisance procurée par les classes et les méthodes de *C++*. On a, en mettant en œuvre ces techniques, cherché de plus à modulariser les blocs fonctionnels réalisés et à les regrouper sous une seule interface pour faciliter leur utilisation en tant que librairie.

Utilisation d'un gestionnaire de versions Un outil essentiel aux projets informatiques collaboratifs où plusieurs développeurs travaillent sur les mêmes fichiers sources est un gestionnaire de versions du type SVN [33]. Ce logiciel conserve les versions successives de tous les documents qu'il supervise et peut calculer les différences de version à version. Cela évite la perte accidentelle de données et permet de revenir à d'anciennes versions stables des fichiers en cas de besoin. Cela impose un formalisme de plus dans l'organisation des fichiers et des dossiers sur lesquels on travaille mais il s'intègre très rapidement vu tous les avantages acquis. C'est un outil indispensable de management du code source et donc de la production logicielle du laboratoire, qui utilise le client Tortoise SVN.

Programmation par script La programmation par script est très adaptée à la problématique de la recherche [34] car elle permet de prototyper rapidement des chaînes de trai-

tement complexes impliquant des modules et des opérations variées comme ici : parseur, gestion de fichiers, lancement de *threads*, utilisation de librairies graphiques. . .. Le durée du cycle de développement logiciel classique s'en trouve réduite car les instructions sont plus haut niveau et on évite la compilation ; par contre, les langages de script sont généralement interprétés à l'exécution, ce qui les rend beaucoup plus lents que les programmes compilés. C'est le langage Python qui a été choisi dans ce cas car il dispose d'une documentation conséquente, d'une communauté très active, de bibliothèques abouties et qu'il est relativement facile à apprendre.

Documentation du code Commenter le code est une condition nécessaire à une réutilisation aisée dans le futur par son auteur ou par d'autres. Il est donc essentiel de documenter le code pendant son écriture et non pas *a posteriori*. On utilise pour cela des outils de génération de documentation automatiques comme Doxygen [35], qui imposent un formalisme particulier aux indications que l'on rajoute au code et qui permettent ensuite de générer des diagrammes et des pages Internet liées très facilement. Des notes de fonctionnement global et de « démarrage rapide », donnant plus de recul au lecteur sont également rédigées.

Conventions de développement Des conventions de programmation pour la production de code lisible et compréhensible par tous ont été mises en place aussi niveau de Thales Communications, dans la continuité de l'effort constant de standardisation des procédures, des outils et de la qualité des produits qui est au cœur de la politique du groupe. Ces conventions réglementent entre autres l'orthographe et la syntaxe lors du nommage de variables, de types de données, de fonctions et de classes et énoncent des bonnes pratiques de programmation. Les règles sont compilées dans un manuel dont la lecture fait partie de la formation initiale à l'entrée dans la société.

6.2 Gestion de projet

6.2.1 Planification et communication

Dès le début du stage, on a veillé à bien cadrer les activités en termes de contenu et de temps. Le diagramme GANTT a été l'outil principal de pilotage du stage, avec les inévitables recadrages en cas d'impasses ou de nouvelles voies à défricher. D'autre part, on a veillé à maintenir des réunions fréquentes (en moyenne une fois par semaine) pour rendre compte de l'état d'avancement du projet d'une part et éventuellement solliciter de l'aide ou soumettre une question d'autre part ; un document résumant les sujets abordés et les solutions envisagées était rédigé à l'issue de la réunion. La fréquence de ces points d'avancement était suffisamment juste pour permettre une certaine autonomie et dans le même temps maintenir un contact rapproché pour que le tuteur ne perde pas de vue le travail en cours.

6.2.2 Capitalisation des connaissances

Pour éviter au laboratoire une perte des connaissances acquises au cours du stage lors du départ et plutôt capitaliser le savoir, des procédures de *Knowledge Management* ont été mises en place, notamment un wiki (site Internet collaboratif) sur le réseau interne, dépassant le simple cadre du stage car conçu dans l'objectif d'améliorer la communication et le partage du savoir au sein du laboratoire. Un document simple et portable

pour explorer l'étude bibliographique réalisée a également été mis à disposition des personnes intéressées, sous forme d'une *mindmap* [36] (réalisée avec le logiciel libre *Freemind*), présentant les articles sous forme arborescente et comprenant un résumé et un lien vers le fichier PDF. Les rapports et présentations produits au cours du stage seront aussi accessibles par le biais du wiki.

Conclusion et perspectives

Le bilan à quelques semaines de la fin du stage est globalement positif : un prototype fonctionnel de système stéganographique complet a été développé. Il est basé sur des techniques d'enfouissement sur les frames I et P de l'état de l'art, qui ont été améliorées et fusionnées pour former un schéma hybride, permettant d'atteindre une performance de débit d'enfouissement de 5 Kbits/s en résolution SD et à 25 images/s, avec un GoP de 16 images, de la forme I/P et une profondeur de référence limitée à 1. Le système vient s'intégrer au codec du laboratoire et remplit les exigences du cahier des charges en termes de capacité, de complexité, de qualité de transmission et d'indétectabilité, alors que les méthodes de l'état de l'art ne parvenaient pas à satisfaire tous ces critères. Un *benchmark* complet de stéganalyse a également été implémenté, à partir de techniques issues de la littérature adaptées au cadre particulier de notre projet ; des approches originales ont aussi été testées.

On n'a encore pas pu valider le critère de non-perturbation des ROI car les outils nécessaires sont encore en cours de réalisation au sein du laboratoire et ne nous ont donc pas été fournis. Ce critère a néanmoins été pris en compte dans le choix des solutions techniques. D'autre part, des procédures de capitalisation des connaissances acquises ont été mises en œuvre pour pérenniser le savoir au sein du laboratoire et pour faciliter la réutilisation future du travail accompli.

Différentes tâches vont venir occuper le mois de stage restant. On travaillera notamment avec l'équipe audio du laboratoire sur un démonstrateur couplant le vocodeur HSX à 2.4 Kbits/s développé par Thales Communications au système stéganographique réalisé pendant ce stage dans le but d'enfouir discrètement un flux audio dans un flux vidéo dans le cadre de communications sécurisées. Ce démonstrateur dépasse le cadre du projet UrbanView, mais sera partie prenante du projet Infom@gic pour la thématique « Fusion d'informations multimédia ». On étudiera également plus en profondeur les descripteurs de comparaison aux modèles de distributions des cœfficients de transformée, qui ont été formalisés mais pas encore implémentés.

Les perspectives d'évolution du système réalisé sont nombreuses, notamment du fait du développement constant du codec H.264 et des nouvelles fonctionnalités qui lui sont ajoutées, constituant autant de nouvelles contraintes ou voies à exploiter. Dans un futur proche, l'équipe du laboratoire se réappropriera les différents blocs logiciels réalisés pour mettre au point un démonstrateur complet de la chaîne de traitement pour le projet UrbanView, incluant l'enfouissement de données. Il est aussi envisagé de mener des tests de robustesse de l'enfouissement à la perte de frames dans le cadre d'une utilisation en *streaming* en étudiant l'apport éventuel de codes correcteurs d'erreurs et l'introduction de symboles de synchronisation dans le flux binaire à enfouir. L'allocation de débit d'enfouissement, idée qui a germé tardivement,

Références

[1] C. Chastagnol. Stéganographie en domaine vidéo compressé — Rapport Bibliographique, 2009. École Centrale de Lyon.

[2] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7), 2003.

[3] I. Richardson. H.264 / MPEG-4 Part 10 White Paper - Prediction of Intra Macroblocks, 2003. `http://www.vcodex.com/`.

[4] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information Hiding - A Survey. In *Proceedings of the IEEE*, volume 87, 1999.

[5] G. J. Simmons. The Prisoners' Problem and the Subliminal Channel. In D. Chaum, editor, *Advances in Cryptology : Proceedings of Crypto 83*, pages 51–67. Plenum Press, 1983.

[6] C. Cachin. An Information-Theoretic Model for Steganography. In *Proceedings of the Second International Workshop on Information Hiding*, pages 306–318. Springer-Verlag, 1998.

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, second edition, 2006.

[8] M. Wu and B. Liu. Data Hiding in Image and Video - Fundamental Issues and Solutions (Part 1). *IEEE Transactions on Image Processing*, 12(6), 2003.

[9] N. Provos and P. Honeyman. Hide and Seek : An Introduction to Steganography. *IEEE Security and Privacy*, 2003.

[10] Polytechnic University. Image steganography and steganalysis, 2004.

[11] P. Moulin and J. A. O'Sullivan. Information-Theoretic Analysis of Information Hiding. *IEEE Transactions on Information Theory*, 49(3), 2003.

[12] B. Chen and G. W. Wornell. Quantization index modulation : a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory*, 47(4) :1423–1443, May 2001.

[13] R. Chandramouli and K. P. Subbalakshmi. Current Trends in Steganalysis : A Critical Survey. 2006.

[14] C.-V. Nguyen, D.B.H. Tay, and G. Deng. A Fast Watermarking System for H.264/AVC Video. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 81–84, Dec. 2006.

[15] Y. Hu, C. Zhang, and Y. Su. Information Hiding Based on Intra Prediction Modes for H.264/AVC. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1231–1234, 2007.

[16] M. Noorkami and R. M. Mersereau. Compressed-Domain Video Watermarking for H.264. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages 890–893, 2005.

[17] Contributeurs Wikipédia. Registre à décalage. `http://fr.wikipedia.org/wiki/Registre_%C3%A0_d%C3%A9calage`.

[18] I. E. Richardson. *H.264 and MPEG-4 Video Compression : Video Coding for Next-generation Multimedia*. Wiley, 2003.

[19] M. Goljan, J. J. Fridrich, and R. Du. Distortion-Free Data Embedding for Images. In *IHW01 : Proceedings of the 4th International Workshop on Information Hiding*, pages 27–41. Springer-Verlag, 2001.

[20] J. Fridrich, M. Goljan, and R. Du. Invertible Authentication Watermark for JPEG Images. In *Information Technology : Coding and Computing, International Conference on*, pages 223–227, 2001.

[21] D. Dönigus, S. Endler, M. Fischlin, A. Hülsing, P. Jäger, A. Lehmann, S. Podrazhansky, S. Schipp, E. Tews, S. Vowe, M. Walthart, and F. Weidemann. Security of Invertible Media Authentication Schemes Revisited. In *Information Hiding*, volume 4567 of *Lecture Notes in Computer Science*, pages 189–203. Springer, 2007.

[22] G. Qiu, P. Marziliano, A. T. S. Ho, D. He, and Q. Sun. A Hybrid Watermarking Scheme for H.264/AVC Video. In *ICPR04 : Proceedings of the Pattern Recognition, 17th International Conference on*, volume 4, pages 865–869. IEEE Computer Society, 2004.

[23] S. Lyu and H. Farid. Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines. In *IH02 : Revised Papers from the 5th International Workshop on Information Hiding*, pages 340–354. Springer-Verlag, 2003.

[24] V. Pankajakshan and A. T. S. Ho. Improving Video Steganalysis Using Temporal Correlation. In *IIH-MSP '07 : Proceedings of the Third International Conference on International Information Hiding and Multimedia Signal Processing*, pages 287–290. IEEE Computer Society, 2007.

[25] U. Budhia. Steganalysis of Video Sequences Using Collusion Sensitivity. Master's thesis, Texas A&M University, 2005.

[26] U. Budhia, D. Kundur, and T. Zourntos. Digital Video Steganalysis Exploiting Statistical Visibility in the Temporal Domain. *Information Forensics and Security, IEEE Transactions on*, 1(4) :502–516, 2006.

[27] C. Zhang, Y. Su, and C. Zhang. A New Video Steganalysis Algorithm against Motion Vector Steganography. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–4, 2008.

[28] C. Parisot. *Allocations basées modèles et transformée en ondelettes au fil de l'eau pour le codage d'images et de vidéos*. PhD thesis, Université de Nice - Sophia Antipolis, Projet CREATIVE, 2003.

[29] E. Renan. Allocation Débit-Distorsion pour H.264/MPEG4-AVC. Master's thesis, École Nationale Supérieure de l'Électronique et de ses Applications — ENSEA, 2005.

[30] Contributeurs Wikipédia. Exploration de données. `http://fr.wikipedia.org/wiki/Exploration_de_donn%C3%es`.

[31] Wikipedia contributors. Presentation of Weka. `http://en.wikipedia.org/wiki/Weka_(machine_learning)`.

[32] T. A. Schreiner. *Objekt-orientierte Programmierung mit ANSI-C*. Hanser, 1994. `http://www.planetpdf.com/codecuts/pdfs/ooc.pdf`.

[33] Wikipedia Contributors. Subversion (software). `http://en.wikipedia.org/wiki/Subversion_(software)`.

[34] J. K. Ousterhout. Scripting : Higher Level Programming for the 21st Century. *IEEE Computer*, 31 :23–30, 1997.

[35] Wikipedia Contributors. Doxygen. http://en.wikipedia.org/wiki/Doxygen.

[36] Wikipedia Contributors. Mind map. http://en.wikipedia.org/wiki/Mind_map.

[37] Wikipedia Contributors. Just-Noticeable Difference. http://en.wikipedia.org/wiki/Just-noticeable_difference.

[38] Wikipedia Contributors. Pseudo-Random Noise Generator. http://en.wikipedia.org/wiki/PRNG.

[39] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

A Lexique, acronymes employées

4CIF : voir CIF.

AC : dans ce contexte, désigne les composantes alternatives (ordre supérieur à 1) dans une décomposition fréquentielle de type DCT, transformée de Fourier discrète ou transformée entière.

CIF : acronyme de *Common Intermediate Format*. Norme standardisant la résolution verticale et horizontale en pixels des composantes YC_bC_r d'un signal vidéo. Cette résolution est de 352×288 pixels. Les résolution QCIF (pour *Quarter* CIF, de 176×144 pixels) et 4CIF (704×576 pixels) existent également.

DC : désigne dans ce contexte la composante continue (moyenne) dans une décomposition fréquentielle de type DCT ou transformée de Fourier.

DCT : acronyme de *Discrete Cosine Transform* ou transformée en cosinus discrète. C'est une transformation dans le domaine des fréquences proche de la transformée de Fourier et très utilisée en traitement du signal.

Frame : anglicisme utilisé dans le domaine du traitement de l'image. Désigne une des images ou « trame » composant une séquence vidéo.

GoP : acronyme de *Group of Pictures*. Désigne le motif de base d'agencement temporel des frames codées. Un GoP commence par une frame I, suivie de frames P et B alternées ; une longueur de 16 est typique.

H.264 : aussi H.264/AVC ou MPEG-4 partie 10. Nom d'un nouveau format de compression vidéo permettant des gains substantiels en stockage de mémoire au prix d'une complexité beaucoup plus élevée. Exploite notamment la redondance spatiale des séquences vidéo.

IT : acronyme de *Integer Transform* ou transformée entière ; elle est utilisée dans le standard de compression H.264 car elle est exacte et se calcule sur des entiers, permettant ainsi de réduire les temps de calcul.

JND : acronyme de *Just Noticeable Difference* ou seuil différentiel. Il s'agit d'un concept théorique de la psychophysique qui définit la limite en dessous de laquelle un individu ne parvient plus à différencier deux stimulations. On l'étend au domaine de la stéganographie pour désigner la limite maximale de distorsion qu'un processus d'enfouissement peut introduire dans un médium-hôte sans être détectable visuellement ou par un algorithme de stéganalyse (source : [37]).

LSB : acronyme de *Least Significant Bit* ; désigne le bit de poids faible dans un codage binaire.

MV : acronyme de *Motion Vector* ; désigne le déplacement d'un bloc 8×8 pixels dans les schémas de compression vidéo de type MPEG et est utilisé pour réduire l'information temporelle redondante lors de l'étape d'estimation de mouvement.

PRNG : acronyme de *Pseudo-Random Noise Generator* ou générateur de bruit pseudo-aléatoire. D'après [38], c'est un algorithme qui génère une séquence de nombres présentant certaines propriétés du hasard. Par exemple, les nombres sont supposés être approximativement indépendants les uns des autres, et il est potentiellement difficile de repérer des groupes de nombres qui suivent une certaine règle (comportements de groupe).

QCIF : voir CIF.

QIM : acronyme de *Quantization Index Modulation*. Il s'agit d'une technique d'enfouissement de données adaptative basée sur l'utilisation de quantificateurs scalaires (voir [12] pour plus de détails).

ROI : acronyme de *Region Of Interest*. Désigne une zone d'une image présentant des caractéristiques intéressantes du point de vue d'une application donnée.

SSM : acronyme de *Spread-Spectrum Modulation* ou modulation par étalement de spectre. C'est une technique issue du monde des télécommunications qui va étaler un signal à l'origine confiné sur une bande de fréquence précise sur une bande beaucoup plus large. L'objectif est en général de rendre la communication plus robuste au bruit et aux éventuelles interférences ou moins facilement détectable.

SVM : acronyme de *Support Vector Machine*. Désigne une technique d'apprentissage supervisé utilisée pour de la classification ou de la discrimination de données. La méthode construit un hyperplan séparateur entre deux ensembles de données vectorielles (source : [39]).

B Corpus vidéo

Les expériences ont été menées sur un corpus de 25 vidéos de provenances diverses (séquences originales, *data sets* de référence...), encodées au format YUV 4 : 2 : 0. Plusieurs résolutions et *framerates* ont été employés quand ils étaient disponibles. Les résolutions utilisées ne sont pas inférieures au CIF. Le corpus est détaillé ci-dessous.

Nom	Résolution	Framerate	Nombre de frames	Remarques
576i25_parkrun_ter	720x576	25	252	Libre de droits. Travelling.
576i25_shields_ter	720x576	25	252	Libre de droits. Zoom.
576i25_stockholm_ter	720x576	25	252	Libre de droits. Pan.
BUS_352x288_15_orig_01	352x288	15	75	Pan, zoom et objets en mouvement.
BUS_352x288_30_orig_01	325x288	30	150	
carphone_cif	352x288	30	382	Corpus ACTICOM/Université d'État de l'Arizona. Plan rapproché, peu de mouvement.
CITY_352x288_15_orig_01	352x288	15	150	©ABC. Travelling circulaire.
CITY_352x288_30_orig_01	352x288	30	300	
CITY_704x576_30_orig_01	704x576	30	300	
CITY_704x576_60_orig_01	704x576	60	600	
Coastguard_CIF	352x288	30	300	Pan, tilt.
Container_CIF	352x288	30	300	Corpus ACTICOM/Université d'État de l'Arizona. Caméra fixe, objets en mouvement.
CREW_352x288_15_orig_01	352x288	15	150	©NASA. Dézoom, pan, personnages en mouvement.
CREW_352x288_30_orig_01	352x288	30	300	
CREW_704x576_30_orig_01	704x576	30	300	
CREW_704x576_60_orig_01	704x576	60	600	
Flower_CIF	352x288	25	250	Travelling, objets en mouvement.
FOOTBALL_352x288_15_orig_C	352x288	15	130	Mouvements de caméra rapides, personnages en mouvement.
FOOTBALL_352x288_30_orig_C	352x288	30	260	
FOREMAN_352x288_15_orig_0	352x288	15	150	Corpus ACTICOM/Université d'État de l'Arizona. Grands mouvements de caméra, personnage.
FOREMAN_352x288_30_orig_0	352x288	30	300	
Hall_Monitor_CIF	352x288	30	300	Caméra fixe, personnages en mouvement.
HARBOUR_352x288_15_orig_0	352x288	15	150	©Demographx. Mouvements de caméra, objets en mouvement.
HARBOUR_352x288_30_orig_0	352x288	30	300	
HARBOUR_704x576_30_orig_0	704x576	30	300	
HARBOUR_704x576_60_orig_0	704x576	60	600	
ICE_352x288_15_orig_02	352x288	15	120	©ABC. Caméra fixe, personnages en mouvement.
ICE_352x288_30_orig_02	352x288	30	240	
ICE_704x576_30_orig_02	704x576	30	240	
ICE_704x576_60_orig_02	704x576	60	480	
MOBILE_352x288_15_orig_01	352x288	15	150	Corpus ACTICOM/Université d'État de l'Arizona. Dézoom, pan, objets en mouvement.
MOBILE_352x288_30_orig_01	352x288	30	300	
Mother_Daughter_CIF	352x288	30	300	Caméra fixe, personnages.
Paris_CIF	352x288	30	300	Corpus ACTICOM/Université d'\'Etat de l'Arizona. Caméra fixe, personnages.
Silent_CIF	352x288	30	300	Caméra fixe, personnages.
SOCCER_352x288_15_orig_02	352x288	15	150	Mouvements de caméra complexes, personnages en mouvement.
SOCCER_352x288_30_orig_02	352x288	30	300	
SOCCER_704x576_30_orig_02	704x576	30	300	
SOCCER_704x576_60_orig_02	704x576	60	600	
Speedway_CIF_ref	352x288	30	200	Corpus projet Infom@gic. Caméra fixe, objets en mouvement.
Tempete_CIF	352x288	30	260	Dézoom.
Tennis_720x576_420	720x576	30	702	Mouvements de caméra complexes, changements de scènes, personnages en mouvement.
turin_704x288_ref	704x288	15	389	Corpus projet Infom@gic. Caméra fixe, personnages en mouvement.
Waterfall_CIF	352x288	30	260	Dézoom.

TABLE 6 – Liste des vidéos du corpus de test et de leurs caractéristiques.