Numéro de bibliothèque

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE PAU \mathbf{ET} DES PAYS DE L'ADOUR

ÉCOLE DOCTORALE DES SCIENCES EXACTES ET DE LEURS APPLICATIONS

PAR Théodore TOTOZAFINY

POUR OBTENIR LE GRADE DE

DOCTEUR

Spécialité :

INFORMATIQUE

COMPRESSION D'IMAGES COULEUR POUR APPLICATION À LA TÉLÉSURVEILLANCE ROUTIÈRE PAR TRANSMISSION VIDÉO À TRÈS BAS DÉBIT

Soutenue le 3 juillet 2007

Après avis de :

M^{me}Jenny BENOIS-PINEAU, Professeur à l'Université de Bordeaux 1 Rapporteur M. Michel PAINDAVOINE, Professeur à l'Université de Bourgogne Rapporteur

Devant la Commission d'examen formée de :

M. Pierre MARCHÉ, Professeur à l'université de Bourges Président M. Jean-Marc COUTELLIER, Directeur Général, MAGSYS Examinateur Examinateur

M. Franck LUTHON, Directeur de thèse, Professeur à l'UPPA

M. Olivier PATROUIX, Encadrant, Enseignant Chercheur à l'ESTIA Examinateur

À ma mère Mahatsara.

Remerciements

Le travail présenté dans ce mémoire a été réalisé au sein de l'entreprise MA-GYS, du Laboratoire Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA) et du Laboratoire d'Ingénierie des Processus et Services Industriels de l'ESTIA (LIPSI). C'est pourquoi je tiens à remercier Messieurs Congduc Pham, Directeur du LIUPPA, et Pascal Weil, Responsable scientifique du LIPSI, de m'avoir accueilli.

J'adresse ma profonde gratitude à Monsieur Jean-Marc Coutellier, directeur de MAGSYS, pour son encadrement spécifique et enrichissant durant la réalisation de cette thèse, pour tout ce qui concerne l'application industrielle du projet.

Un grand merci aussi à Monsieur Franck Luthon, mon directeur de thèse, et à Monsieur Olivier Patrouix, mon encadrant, pour leurs attentions, leurs encouragements et leurs conseils tout au long de ces trois années.

Je tiens à exprimer ma reconnaissance à Madame Jenny Benois-Pineau, Professeur à l'université de Bordeaux 1, et à Monsieur Michel Paindavoine, Professeur à l'université de Bourgogne, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs.

Et une reconnaissance toute spéciale va à Monsieur Pierre Marché, Professeur à l'université de Bourges, qui a accepté de présider le jury de soutenance.

J'adresse ici toute ma gratitude à Monsieur Jean-Roch Guiresse, directeur de l'ESTIA, CCI-Bayonne, qui m'a sans cesse soutenu durant ma formation à l'ES-TIA jusqu'à la fin de cette thèse, et sans qui je ne serais jamais arrivé jusque là.

Je voudrais aussi remercier mon cher ami Bruno Kieffer et ma chère amie Marie-José Deshayes de m'avoir aidé pour la rédaction du manuscrit de cette thèse.

Un grand merci à mes collègues de la société MAGYS : Bruno Claverie, Guillaume Pamart et Sabine Delarue, pour leur collaboration durant ces trois années passées chez MAGYS.

Enfin je remercie tout le personnel de l'ESTIA ainsi que les doctorants du LIPSI pour l'aide qu'ils m'ont apportée tout au long de la réalisation de cette thèse.

Résumé

L'objectif de cette thèse est de valider, pour un réseau sans fil bas débit GSM (débit maximal 9600 bits/s), la possibilité d'obtenir la cadence de la transmission à une image par seconde. Cela contraint la taille de l'image en mémoire à être inférieure à 1.2 Ko et l'image doit être encodée avec un standard existant pour envisager le développement d'un produit industrialisable par l'entreprise MAGYS. Afin de répondre à cet objectif, différents points ont été étudiés :

Le premier point concerne le type de données à envoyer : séquence d'images ou image fixe. Pour cela nous avons réalisé des tests comparatifs, dans notre contexte applicatif, entre la norme de codage vidéo MPEG-4 (la plus répandue actuellement) et le standard de codage d'image fixe JPEG2000 (le plus performant en terme de taux de compression). Les résultats de ces tests nous ont conduit à élaborer un nouveau système de codage (codec) basé sur la gestion de régions d'intérêt (ROI) du JPEG2000.

Le deuxième point concerne la réduction maximale de données à envoyer. L'idée est la suivante : détourner la fonctionnalité de la gestion de régions d'intérêt du standard JPEG2000, afin d'obtenir un taux de compression très élevé (1 : 250). Nous divisons l'image en deux régions : le fond fixe et les régions mobiles. L'image contenant les régions mobiles est ensuite encodée par la gestion de régions d'intérêt du JPEG2000, qui est mise en œuvre par la technique Maxshift. La propriété principale de cette technique est que le décodeur peut décoder l'image sans avoir recours aux informations spatiales relatives au masque utilisé par l'encodeur. Notre stratégie exploite cette propriété, notamment si le décodeur dispose d'une image de référence; la reconstruction d'image finale est possible par l'apposition de l'image reçue sur l'image de référence utilisant le masque reconstruit implicitement. Ainsi, nous pouvons transmettre seulement les régions d'intérêt. Le reste de l'image est à ignorer donc affecté à la valeur médiane de la dynamique de composantes couleurs, typiquement égale à 128 afin d'améliorer la compression. Après l'encodage par JPEG2000 à très fort taux de compression, nous obtenons une taille en mémoire de données à envoyer inférieure à 1.2 Ko.

Le dernier point étudie la réactualisation de l'image de référence au décodeur. Nous proposons une technique originale réalisant une mise à jour de l'image de référence du décodeur par morceaux. Ces derniers représentent les régions pertinentes dans l'image de référence. Nous conduisons cette mise à jour selon le même principe basé sur la gestion de région d'intérêt. Le morceau de l'image de référence nous impose de disposer de deux masques : mouvement (régions mobiles) ou partie du fond (région pertinente). La mise à jour est effectuée en deux étapes. La première étape définit la stratégie du déclenchement de la réactualisation, considérant les trois configurations suivantes : sans mouvement, peu de mouvement ou beaucoup de mouvement, définies en fonction du taux de pixels mobiles dans l'image. La seconde étape concerne le choix des régions à mettre à jour; pour définir le (ou les) morceau(x) prioritaire(s), nous découpons l'image de référence en blocs carrés et chacun de ces blocs possède un coefficient de priorité. Par cette stratégie, les données à envoyer, pour les régions mobiles ou pour le morceau de l'image de référence, sont inférieures à 1.2 Ko, garantissant la cadence de transmission à une image par seconde. Cet objectif étant atteint, un réseau haut débit, par exemple, l'UMTS nous permettra d'envisager une transmission à 25 img/sec.

Abstract

This thesis presents a feasibility study for transmitting images in GSM wireless networks (9600 bits/s maximal bit rate) with a frequency of one image per second, by using widespread image or video codecs. Due to this constraint, the maximal size of image data is 1.2 KiB. In particular, the following aspects were studied :

First, the type of data to be sent : video streaming or still image sequencing. We carried out several comparative tests, in the context of video surveillance, between video streaming with the MPEG-4 video coding standard (currently, the most widespread) and still image sequencing with JPEG2000 coding standard (currently, the best compression ratio). The results of these tests led us to propose a new coding system (codec) based on a particular feature of the JPEG2000 codec : the Region of Interest (ROI).

The second aspect is the maximal reduction of transmission data. Our approach is as follows : to divert the functionality of the JPEG2000's ROI feature at the start, in order to obtain a very high compression ratio — around 1 :250. The image is divided in to areas : background and regions of interest (i.e. mobile object areas). Only the image mobile object regions are compressed with the JPEG2000 ROI feature, implemented using the Maxshift technique. The most important property of this technique is that image decoding can be completed without the spatial information of the mask (used at coding). Our technique exploits this property by sending exclusively the ROI data. The background is modified with the median value of the dynamics of component colours, typically equal to 128. This modification improves the compression ratio. Finally, the image is compressed with JPEG2000 at a very high compression ratio. So, the resulting data is always lower than 1.2 KiB.

The last aspect refers to the reference image updating at the decoder. We propose an original technique to update by pieces. These pieces represent the relevant areas of the reference image. To achieve this, we employ the same mechanism for encoding mobile object regions. The pieces strategy incorporates the need of an additional mask : one for motion (mobile areas) and other one for the reference image (relevant pieces). The update is carried out in two stages. First, the definition of the updating strategy, considering the three following configurations : no mobile objects, few mobile objects or many mobile objects, defined accordingly to the rate of mobile pixels in the image. Then, the second stage is the choice of the region to be updated by priorities. To define the pieces priorities, we cut out the reference image by square blocks and a priority coefficient is assigned to each block. Using this strategy, the size of the data to be sent (mobile region and piece of the reference image) is lower than 1.2 KiB, keeping the transmission rate to an image per second. Consequently, real-time transmission (i.e. 25 images per second) could be achieved in the case of high bit rate networks, for example UMTS.

Mots clés

Région d'intérêt

Détection de mouvement

Réactualisation d'une image de référence

Priorité

Logique floue

Compression élevée

Codec

Implantation matérielle

Transmission sans fil

Bas débit

Vidéosurveillance

Key Words

Region of Interest Motion Detection Reference Image Updating Priority Fuzzy Logic High compression ratio Codec Hardware Implantation Wireless Transmission Low Bit Rate Video Surveillance

Table des matières

Introduction

1.	Un	état de	e l'art : Système de codage vidéo	5
	1.1.	Nécess	sité de la compression	5
	1.2.	Redon	dances dans une vidéo	6
	1.3.	Techni	ique de codage	6
	1.4.	Codag	ge sans perte	8
		1.4.1.	Codage de Huffman	8
		1.4.2.	Codage RLC (Run Length Coding)	9
		1.4.3.	Codage Lempel-Ziv	9
	1.5.	Codag	ge avec perte	10
		1.5.1.	Codage par quantification	10
			1.5.1.1. Quantification scalaire (QS) $\ldots \ldots \ldots \ldots$	11
			1.5.1.2. Quantification vectorielle (QV)	12
		1.5.2.	Codage par prédiction	12
		1.5.3.	Codage par transformée	13
			1.5.3.1. Transformée de Karhunen-Loeve KLT	13
			1.5.3.2. Transformée de Fourier Discrète DFT	14
			1.5.3.3. Transformée en Cosinus Discrète DCT	14
			1.5.3.4. Transformée en Ondelettes Discrètes DWT	14
			1.5.3.5. Décomposition pyramidale	21
	1.6.	Codag	ge d'images fixes : standards JPEG	23
		1.6.1.	JPEG	23
			1.6.1.1. Principe	24
			1.6.1.2. DCT	24
			1.6.1.3. Quantification	24
			1.6.1.4. Réorganisation en zig-zag	24
			1.6.1.5. Codage entropique	24

1

		1.6.1.6. Limite du stan	dard JPEG			. 25
	1.6.2.	JPEG2000				. 25
		1.6.2.1. Performance de	e JPEG2000			. 25
		1.6.2.2. Architecture du	ı standard			. 26
		1.6.2.3. Découpage en t	uiles			. 27
		1.6.2.4. Transformées c	ouleur			. 27
		1.6.2.5. Décomposition	en ondelettes			. 28
		1.6.2.6. Quantification				. 29
		1.6.2.7. Codage entropi	que			. 31
		1.6.2.8. Organisation d	u flux de sortie			. 38
		1.6.2.9. Résistance aux	erreurs			. 38
		1.6.2.10. Gestion de régi	ons d'intérêt			. 39
1.7.	Codag	e vidéo : standards du mu	ultimédia			. 42
	1.7.1.	Norme MPEG				. 43
		1.7.1.1. Principe de la r	norme MPEG			. 43
		1.7.1.2. MPEG-1				. 45
		1.7.1.3. MPEG-2				. 45
		1.7.1.4. MPEG-4				. 45
		1.7.1.5. MPEG-7				. 46
		1.7.1.6. MPEG-21				. 47
	1.7.2.	Norme UIT-T				. 47
		1.7.2.1. H.261				. 47
		1.7.2.2. H.263				. 47
		1.7.2.3. AVC/H.264 .				. 47
1.8.	Choix	stratégique : image fixe o	u vidéo			. 48
	1.8.1.	Objectif des tests				. 48
		1.8.1.1. MPEG-4 à très	bas débit			. 48
		1.8.1.2. JPEG2000 à fo	rt taux de compre	ssion		. 50
	1.8.2.	Conclusion				. 50
1.9.	Vers la	compression par régions	d'intérêt			. 51
	1.9.1.	Extraction de régions de	mouvement			. 52
		1.9.1.1. Différence temp	oorelle d'images .			. 52
		1.9.1.2. Différence temp	oorelle du fond			. 56
1.10	. Conclu	sion				. 63
Cod	age pa	le contenu pour la con	npression d'image	es mobiles		65
2.1.	Descri	ption générale du système			• •	. 65
2.2.	Définit	ion du système local-dist	ant			. 65

2.

	2.3.	Schém	a-bloc de l'encodeur	66
		2.3.1.	Phase d'initialisation	66
			2.3.1.1. Filtrage récursif du premier ordre	68
			2.3.1.2. Vérification d'hypothèses	68
			2.3.1.3. Limitations dues au modèle	69
			2.3.1.4. Paramètres de l'initialisation	69
		2.3.2.	Construction de la région d'intérêt	70
		2.3.3.	Image JPEG2000	71
			2.3.3.1. Suppression du fond \ldots \ldots \ldots \ldots \ldots \ldots	71
			2.3.3.2. Compression	72
		2.3.4.	Transmission d'images objets	72
	2.4.	Conclu	usion	73
3.	Déc	odage	et Réactualisation d'images de références	75
	3.1.	Schém	a-bloc du décodeur	75
		3.1.1.	Identification d'image	75
		3.1.2.	Reconstruction implicite du masque et décodage	76
		3.1.3.	Construction de l'image finale $\ldots \ldots \ldots \ldots \ldots \ldots$	76
			3.1.3.1. Notations \ldots	76
			3.1.3.2. Construction \ldots	77
	3.2.	Réactu	ualisation d'images de références	77
		3.2.1.	Rappel sur la logique floue	78
		3.2.2.	Mise à jour d'une image de référence par priorité	79
			3.2.2.1. Technique par image de référence prête	80
			3.2.2.2. Technique par coefficients de priorité	87
	3.3.	Conclu	usion	91
4.	Exp	ériment	tation et Résultats	93
	4.1.	Premi	ère expérimentation : vidéo enregistrée	94
		4.1.1.	Dispositif expérimental	94
			4.1.1.1. Aspects matériels et outils de développement	94
			4.1.1.2. Séquences vidéo utilisées	94
		4.1.2.	Résultats : Phase d'initialisation	95
			4.1.2.1. Construction d'une image de référence	95
			4.1.2.2. Evaluation de la qualité	96
			4.1.2.3. Temps de construction	100
		4.1.3.	Résultats : Gestion de la ROI	100
			4.1.3.1. Encodeur : la construction $\ldots \ldots \ldots \ldots \ldots$	100

			4.1.3.2.	Décodeur : la reconstruction implicite	100
		4.1.4.	Résultat	s : Construction de l'image finale	102
			4.1.4.1.	Reconstruction d'images finales	102
			4.1.4.2.	Evaluations	104
		4.1.5.	Résultat	s : Actualisation de l'image de référence distante .	109
			4.1.5.1.	Image prête	109
			4.1.5.2.	Stratégie de transmission et gestion des coefficients	
				de priorité	110
			4.1.5.3.	Conclusion 	116
		4.1.6.	Bilan su	r l'expérimentation	116
	4.2.	Deuxiè	ème expér	imentation : vidéo en ligne	117
		4.2.1.	Disposit	if expérimental	117
		4.2.2.	Résultat	s	117
		4.2.3.	Bilan su	r l'expérimentation	121
	4.3.	Conclu	usion		121
Co	onclus	ion gé	nérale et	perspectives	122
Bil	bliogr	aphie			126
Pu	blica	tions d	e l'auteu	r	133
Α.	Amé	lioratio	on du ren	udu visuel par LUX	137
В.	Sché	émas d	étaillés		141
	B.1.	Encod	eur		141
	В.2.	Décod	eur		142
С.	Arch	nitectur	re de l'im	plantation de l'algorithme	143
D.	JPE	G2000	: transm	issions progressives	145
Ε.	Impl	antatio	ons JPEG	52000	147
	E.1.	Logicie	elle		147
	E.2.	Matéri	ielle		147

Table des figures

0.1.	Fonctionnement du système	3
1.1.	Courbe de distorsion-débit	7
1.2.	Arbre binaire de Huffman.	9
1.3.	Schéma classique d'un système de compression avec perte	10
1.4.	Quantification scalaire.	11
1.5.	Ondelette de Haar.	16
1.6.	Ondelette de Daubechies	17
1.7.	Décomposition en ondelettes 2D en sous-bandes pour un niveau de	
	résolution.	18
1.8.	Les bancs de filtres : analyse et synthèse	19
1.9.	Exemple d'une décomposition en ondelettes à 2 niveaux de résolution.	19
1.10.	Processus de la technique lifting.	21
1.11.	Décomposition pyramidale gaussienne de niveaux 3	23
1.12.	Schéma-bloc de JPEG	24
1.13.	Dégradation d'image JPEG à un taux de compression élevé	25
1.14.	Schéma-bloc de JPEG2000	26
1.15.	Quantification uniforme avec zone morte	29
1.16.	Découpage en blocs rectangulaires; ici les deux résolutions ont la	
	même taille de blocs.	32
1.17.	Construction de la première étape Tier 1 du EBCOT	33
1.18.	Codage par plans de bits à 4 profondeurs	33
1.19.	Regroupement des blocs dans différentes couches de qualité	33
1.20.	Détermination du contexte du bit significatif	34
1.21.	Parcours des coefficients dans un bloc.	35
1.22.	Subdivision d'intervalle d'Elias du MQ-Coder	38
1.23.	Flux de sortie	39
1.24.	Technique pour la résistance aux erreurs du JPEG2000	40
1.25.	Disposition de la région d'intérêt;	40

1.26. Image JPEG2000 avec un taux de compression élevé t=250. $\ .$	41
1.27. Illustration de la technique du Maxshift et du décalage global. $\ .$.	42
1.28. Macrobloc, échantillonnage YUV $(4:2:2)$	44
1.29. Séquence d'images I, P, B de la MPEG	45
1.30. Hiérarchie des objets dans MPEG-4.	46
1.31. Mesure de taille du fichier encodé avec MPEG-4 en fonction du	
bitrate	49
1.32. Mesure de taille du fichier encodé avec JPEG2000 en fonction du	
taux de compression. \ldots	50
1.33. Conséquences du mouvement d'un objet dans la scène à fond fixe.	53
1.34. Histogramme de la valeur d'intensité lumineus e X_i	54
1.35. Filtre de morphologie mathématique [CHH+04]	55
1.36. Diagramme de la technique de soustraction de fond	56
1.37. Variation d'une valeur d'intensité lumineuse	58
2.1. Schémas-blocs du codec proposé	67
2.2. Grand objet mobile; $\Lambda = 20\%$	70
2.3. Illustration du résultat par l'opérateur ET logique	71
2.4. Mise en valeur uniforme des pixels appartenant au fond	72
2.5. En-tête de la transmission	73
3.1. Formalisation de la logique floue.	79
3.2. Construction de l'appartenance de la confiance C	81
3.3. Fuzzification de nz aux μ_{PG} et μ_{Null}	84
3.4. Evolution de la confiance C en fonction de nz	85
3.5. Schéma-bloc de la mise à jour par priorité	88
4.1. Séquence d'images testées	95
4.2. Construction d'une image de référence avec $L = 50$; séquence LAPS.	97
4.3. Construction d'une image de référence ; séquence MAGYS	98
4.4. Construction d'une image de référence ; séquence LABOINFO	99
4.5. Exemples des images JPEG2000	.01
4.6. Exemples d'un masque reconstruit implicitement au décodeur 1	.03
4.7. Exemples des images finales, reconstruites au décodeur à 9600 bit/s	
avec un taux de compression de $1:250$	05
4.8. Zoom sur la reconstruction d'images finales	.06
4.9. Diagramme de résultats des votes	.08
4.10. Variation de la confiance associée à l'image de référence prête de	
la Fig. 4.11-b	10

4.11. Mise à jour de l'image référence du système distant par image de	
référence prête	111
4.12. Réactualisation de l'image de référence en tenant compte des nou-	
velles informations incorporées dans la référence du système local.	112
4.13. Taux d'occupation de l'ensemble d'objets mobiles et les flags de	
mise à jour	113
4.14. Coefficients de priorité et mise à jour de l'image de référence du	
système distant selon la configuration pas d'objet mobile sur la	
scène. \ldots	114
4.15. Coefficients de priorité et mise à jour de l'image de référence du	
système distant selon la configuration beaucoup d'objets mobiles	
sur la scène.	115
4.16. L'encodeur : système matériel	117
4.17. Image de référence avec le système réel	118
4.18. Masques ROI	119
4.19. Images finales reconstruites avec le système réel	120
$4.20.$ Mise à jour d'une image de référence par coefficients de priorité. $\ .$	120
A.1. Amélioration par la transformée non-linéaire logarithmique LUX ;	
le feu arrière est d'un rouge plus vif.	139
B.1. Schéma détaillé de l'encodeur.	141
B.2. Schéma détaillé du décodeur.	142
C.1. Diagramme des processus pour implantation de notre algorithme	
en temps-réel	143
D.1. Transmissions progressives : résolutions et qualité	146

Liste des tableaux

1.1.	Code de Huffman	9
1.2.	Les coefficients des filtres d'analyse et de synthèse des ondelettes	
	de Daubechies $(5,3)$	28
1.3.	Les coefficients des filtres d'analyse et de synthèse des ondelettes	
	de Daubechies $(9,7)$	29
1.4.	Valeur de paramètres de mixture de gaussiennes	62
2.1.	Valeur de paramètres.	70
3.1.	Valeur des paramètres du modèle (image prête et coefficient de	
	priorité)	91
4.1.	PSNR de l'image de référence construite par rapport à une image	
	sans voitures dans la séquence	97
4.2.	Temps d'exécution de la construction d'une image de référence	100
4.3.	Différence entre le masque initial et le masque reconstruit	102
4.4.	PSNR de l'image finale construite	104
4.5.	Résultats des votes; les valeurs entre parenthèses correspondent	
	aux votes pour les séquences originelles	107
4.6.	Valeurs numériques des paramètres	111
4.7.	Temps d'exécution de l'ensemble du traitement, avec un taux de	
	$compression de 1: 188. \dots $	116
4.8.	Temps d'exécution de l'ensemble du traitement avec le système	
	réel. Certes, les codes ne sont pas optimisés mais déjà nous pouvons	
	analyser les résultats.	121
A.1.	Temps de compression : transformé é LUX et standard avec la sé-	
	quence LAPS	138
E.1.	Surcoûts de JPEG2000 par rapport à JPEG [LNR02]	147

Introduction

La vidéosurveillance, selon le dictionnaire Robert, est une surveillance par caméras vidéo. Elle peut se faire soit en local - les caméras sont alors installées et inamovibles, soit, dans le cas contraire à distance. Dans ce dernier cas, on doit transmettre des flux avec une fréquence d'acquisition constante, soit en tempsréel. Cette fréquence d'acquisition est déterminée en fonction de la dynamique des objets à surveiller : faible pour l'obervation d'une rue piétonne et élevée pour celle d'une autoroute. L'objectif temps-réel est atteint si l'on dispose d'une bande passante en adéquation avec la quantité de données et la fréquence d'acquisition. La bande passante de transmission peut rapidement nécessiter une infrastructure lourde et onéreuse.

Actuellement, un défi pour l'industrie est de développer un dispositif de vidéosurveillance au moyen d'équipements diminuant les contraintes d'installation (cablâge, alimentation, etc) mais efficaces. La flexibilité et la portabilité obligent à affronter deux difficultés : l'une est matérielle, pour les traitements, l'autre concerne la bande passante, pour la transmission. De nos jours, il existe déjà des équipements performants pour les traitements grâce notamment à des puces dédiées comme les processeurs DSP ("*Digital Signal Processor*") ou les unités logiques FPGA ("*Field Programmable Gate Array*"). Ce type d'équipement se trouve déjà dans la vie quotidienne dans nos téléphones portables, nos PDA, etc. Pour la transmission, les différents réseaux haut débit tels que *Internet, UMTS* (réseau sans fil troisième génération) laissent envisager une transmission vidéo à une fréquence proche des 25 images par seconde. La vidéosurveillance sans fil impose l'encodage de la source à envoyer et du canal de transmission afin d'obtenir une bonne robustesse aux erreurs et aux pertes.

L'étude menée dans cette thèse s'inscrit dans le contexte industriel de vidéosurveillance d'une scène routière. L'industriel MAGYS [MAG] est intéressé par l'étude et le développement d'un système embarqué pour une application de surveillance routière. Ce système doit permettre l'observation à distance des objets mobiles et doit pouvoir être mis en œuvre sans nécessiter une infrastructure lourde. Il est impératif d'utiliser une transmission sans fil ayant un réseau déjà déployé. Le système pourra avoir diverses applications :

- Sécurité, pour la surveillance de chantiers, de routes;
- Expertise à distance : constats de dégâts matériels;
- Communication : transmission en temps-réel d'événements importants, en évitant une infrastructure trop lourde.

Malgré l'arrivée de réseaux haut débit sans fil, seul le réseau sans fil à très bas débit : GSM est très répandu. A l'utilisation, on se heurte à divers problèmes, comme les erreurs en paquets, le temps d'attente assez long, etc, qui rendent la transmission d'une vidéo perfectible. La mise en œuvre d'une technique de codage est nécessaire afin d'améliorer la qualité de la transmission. Cette technique peut être intégrée, soit dans le protocole réseau, soit au niveau des données à transmettre.

Chez MAGYS, il existe actuellement un équipement permettant ce type de vidéosurveillance. L'image acquise à l'aide d'une caméra statique est encodée au format JPEG puis transmise vers un ordinateur distant via le réseau GSM. L'ordinateur distant décode et affiche l'image reçue (Fig. 0.1). Actuellement, la cadence d'affichage d'une image au décodeur varie entre 8 et 10 secondes. Limitée par la performance en compression de la norme JPEG, le système actuel ne permet pas d'envisager une augmentation de la cadence.

Notre objectif principal est donc de porter la cadence à une image par seconde (1 img/s) pour le canal GSM. Cela demande un gain de facteur 8 par rapport au système actuel. Le rendu visuel des images à la réception doit être identique et en tout cas acceptable par les utilisateurs. Ce dernier est mesuré à l'aide de critères subjectif (fourni par les industriels du domaine de la vidéosurveillance) et objectif (calcul du PSNR).

La cadence de 1 img/s permet d'envisager une cadence de 5 img/s sur le réseau GPRS (en pratique le réseau GPRS offre un gain de facteur 5 par rapport au réseau GSM) et une cadence de 25 img/s pour le réseau UMTS (5 fois plus performant que GPRS). La cadence idéale serait alors atteinte.

Pour mieux illustrer notre problématique, prenons le cas concret d'une image brute (non compressée) de 300 Ko. Pour envoyer cette image à travers le réseau GSM au débit maximal de 9600 bits/s (soit 1.2 Ko/s) en une seconde, celle-ci doit avoir une taille inférieure à 1.2 Ko. On doit alors compresser l'image avec



FIG. 0.1.: Fonctionnement du système.

un taux t = 250, ce qui est irréaliste avec la norme JPEG car l'image devient illisible.

Dans ce contexte industriel, la méthode et l'algorithme proposés doivent satisfaire les contraintes suivantes :

- Limitation de la capacité de mémoire et performance du calcul du processeur;
- Limitation de la bande passante de la transmission;
- Utilisation de normes de codage vidéo standard, disponibles sur le marché.

Ce mémoire s'articule autour de quatre chapitres dans lesquels nous présentons respectivement l'état de l'art (chapitre 1), notre approche (chapitre 2 et 3) et nos résultats expérimentaux (chapitre 4).

Chapitre 1 : nous effectuons un état de l'art sur les différentes techniques et les différentes normes de codage de vidéo dans la littérature. Nous étudions en particulier les performances des standards MPEG-4 et JPEG2000 dans le cadre de notre contexte applicatif. Les résultats nous amènent à choisir le standard JPEG2000 et à introduire un nouveau système de codage par la gestion de régions d'intérêt (option de JPEG2000 pour la visualisation progressive). Nous abordons les méthodes d'extraction automatique de ROI ainsi que les différentes techniques d'obtention d'une image de référence dans le contexte d'une caméra fixe.

Chapitre 2 : nous abordons le développement théorique de notre système de codage. Ce dernier contient différents blocs de traitement : la phase d'initialisation (permettant la construction de la première image de référence), la segmentation en régions (construction de la ROI par extraction des objets mobiles), l'encodage de données par JPEG2000, la transmission de la ROI et la mise à jour d'une image de référence. Nous développons ici ces différentes phases de traitement.

Chapitre 3 : nous présentons le développement théorique de notre système de décodage. Nous proposons ensuite deux techniques innovantes permettant la réactualisation de l'image de référence au niveau du décodeur en évitant l'effondrement de la cadence. La première, celle de l'*image prête,* consiste à trouver automatiquement d'une part le moment où l'image de référence a une qualité suffisante et d'autre part les régions à mettre à jour. La seconde, celle du *coefficient de priorité* est basée sur la subdivision en blocs et l'utilisation de coefficients de priorité par blocs. Un raisonnement par pile de priorité permet de choisir les blocs à mettre à jour.

Chapitre 4 : nous terminons notre étude par une série d'expérimentations pour chaque bloc de notre algorithme et par une présentation de nos résultats. Nous présentons nos dispositifs expérimentaux : séquences d'images enregistrées et vidéo en ligne avec une caméra industrielle. Nous qualifions dans ce chapitre nos résultats à l'aide des critères objectif et subjectif et nous montrons que les objectifs décrits précédemment sont atteints.

Finalement en conclusion, un bilan est dressé sur les aspects de notre contribution et nous proposons différentes pistes et perspectives de travail pour l'avenir tant sur l'aspect industriel que sur celui de la recherche.

1. Un état de l'art : Système de codage vidéo

Sommaire

1.1. Nécessité de la compression	•••	 ••	5
1.2. Redondances dans une vidéo	•••	 ••	6
1.3. Technique de codage	•••	 ••	6
1.4. Codage sans perte	•••	 ••	8
1.5. Codage avec perte	•••	 ••	10
1.6. Codage d'images fixes : standards JPEG	•••	 ••	23
1.7. Codage vidéo : standards du multimédia	•••	 ••	42
1.8. Choix stratégique : image fixe ou vidéo .	•••	 ••	48
1.9. Vers la compression par régions d'intérêt	•••	 ••	51
1.10. Conclusion	•••	 ••	63

1.1. Nécessité de la compression

L'avancée de la technologie de la communication, que ce soit pour les réseaux hétérogènes internet, les réseaux sans fil UMTS ou Wifi, améliore considérablement le débit de la transmission. Pour autant, les données dont on dispose ne cessent d'augmenter. Prenons l'exemple d'un cas concret : une seconde de vidéo codée en 24 bits possédant une résolution spatiale de CIF (352×288) et une résolution fréquentielle de 25 img/s contient au minimun 60 Mbits. Ainsi, un débit de 60 Mbits est nécessaire afin de transmettre cette vidéo en temps-réel. La nécessité d'une compression s'impose alors non seulement pour le stockage de données mais aussi pour la transmission de celles-ci à travers un réseau dont le débit est restreint.

1.2. Redondances dans une vidéo

La réduction des redondances dans la vidéo permet un meilleur taux de compression. Dans ce domaine, il existe différents types de redondances.

- 1. La redondance spectrale définit la corrélation entre les différentes composantes (longueurs d'onde) de la couleur.
- 2. La redondance spatiale est la corrélation entre les pixels voisins.
- 3. La redondance temporelle correspond à la corrélation entre les images de la séquence.
- 4. La redondance psycho-visuelle est l'exploitation de propriétés de la vue humaine.

Ces redondances sont présentes dans les images naturelles. La compression d'une image nécessite l'exploitation de la redondance spectrale, spatiale et psychovisuelle. On parle alors d'une compression spatiale. En ce qui concerne la vidéo, toutes les redondances sont étudiées, dans ce cas on parle d'une compression spatio-temporelle, spatio-fréquentielle ou hybride.

1.3. Technique de codage

Claude E. Shannon fonde la théorie de l'information dans le fameux article "A Mathematical Theory of Communication" en 1948 [Sha48]. Il s'agit d'une discipline fondamentale pouvant s'appliquer dans le domaine des communications. En effet, la théorie de l'information établit un ensemble de règles, en particulier de critères qualitatifs et quantitatifs pour déterminer si une communication est faisable ou non. Cette dernière est faite sur un support donné et dans un contexte bien défini. Le codage de source et le codage de canal sont les principes de base dans la théorie de l'information. Le codage de source consiste en l'élimination de la redondance (temporelle ou spatiale) afin d'y réduire le débit binaire. Le codage de canal, quant à lui, a pour objet la protection contre les erreurs dues aux multiples distorsions que subit le message dans le canal de transmission. La protection est obtenue en ajoutant de la redondance au message à envoyer.

Aussi, le codage de source et le codage de canal sont fondamentalement différents. On s'intéresse particulièrement au codage de source. Le codage de source est appelé aussi compression d'information ou compression de données. La transmission du contenu d'une source d'information nécessite des traitements. Les traitements peuvent être divisés en deux catégories : sans perte d'information ou



FIG. 1.1.: Courbe de distorsion-débit.

avec perte d'information. La compression d'information est dite sans perte lorsqu'il n'y a aucune perte de données sur l'information d'origine. Elle est dite avec perte dans le cas contraire.

D'une autre manière, la compression de données permet de diminuer la taille de stockage et rend possible leur transport à travers des réseaux de communication tels que le GSM, le UMTS ou l'internet. Certaines normes de compression disposent de techniques permettant la résistance aux erreurs en paquets lorsque les données sont transmises sur un réseau dont la qualité de service n'est pas garantie.

Pour une source donnée, la mesure permettant d'évaluer la capacité de codage est l'entropie [Sha48]. La définition de l'entropie H est donnée par l'expression :

$$H = -\sum_{n=0}^{M-1} \Pr(x_n) \times \log_2\left(\Pr(x_n)\right)$$
(1.1)

où $\Pr(x_n)$ est la probabilité de la valeur d'intensité x_n dans la source et M est la dynamique du signal (pour une image, typiquement M = 256).

Shannon a démontré qu'il est possible de diminuer le coût de codage d'une source donnée en regroupant les symboles à coder. Plusieurs techniques de codage sont inspirées par ce théorème, par exemple le codage de Huffman. On peut donc coder la source sans perte d'information (réversible) jusqu'à son entropie. En revanche, au delà de l'entropie une distorsion apparaît dans la source, codage irréversible (Fig. 1.1).

Deux techniques sont utilisées pour évaluer la distorsion.

1. Les méthodes subjectives, nécessitant des tests psychovisuels de l'œil hu-

main. Les tests sont réalisés à plusieures échelles avec des groupes de personnes, et doivent se dérouler selon la procédure des recommandations fournies par la CCIR [BT.95].

2. Les méthodes objectives qui utilisent le rapport signal crête sur bruit PSNR entre la source initiale et celle distordue.

Soient x_n la valeur initiale du signal et \hat{x}_n sa valeur codée ou distordue, N le nombre d'éléments. L'erreur quadratique moyenne EQM est :

$$EQM = \sqrt{\frac{1}{N} \sum_{n} (x_n - \hat{x}_n)^2}$$
 (1.2)

$$PSNR = 20 \times \log_{10}(\frac{M}{EQM}) \tag{1.3}$$

Les techniques de codage les plus utilisées sont :

- pour le codage réversible : codage par plage, Huffman, arithmétique, codage RLC, Lempel-Ziv;
- pour le codage irréversible : quantification, prédiction, utilisation des transformées.

Dans ce manuscrit, on s'intéresse seulement au codage d'image fixe et de vidéo.

1.4. Codage sans perte

1.4.1. Codage de Huffman

Le codage de Huffman consiste à coder les symboles par une représentation de bits à longueur variable. Les symboles ayant la probabilité d'apparition forte sont codés avec des chaînes de bits plus courtes, tandis que les symboles dont la probabilité d'apparition est faible sont codés par des chaînes plus longues. Le code d'un symbole ne doit pas être le préfixe d'un autre code. Cette propriété est admise afin que la reconnaissance soit possible. Pour représenter le codage de Huffman, on utilise l'arbre binaire.

Soit un message à coder "*ABBBBAAC*". La fréquence d'apparition ainsi que le code Huffman correspondant sont donnés dans le Tab. 1.1 et représentés par la Fig. 1.2.



FIG. 1.2.: Arbre binaire de Huffman.

TAB. 1.1.: Code de Huffman.

Symbole	А	В	С
Fréquence d'apparition	3	4	1
Code Huffman	01	1	00

1.4.2. Codage RLC (Run Length Coding)

Plutôt que de coder seulement le message lui-même, il est plus intéressant de coder un message contenant une suite d'éléments répétitifs par "un couple répétition et valeur". Le codage RLC consiste en effet à coder un élément du message par sa valeur de répétition. Considérons le message "AAAAAABBBBBBCCC", le code RLC correspondant est "6A5B3C", ce qui permet d'obtenir un gain de (14-6)/14, soit 57%. On s'aperçoit que plus la suite est longue, plus le débit est grand. Pour autant, s'il n'y a pas de répétition d'éléments, la technique ne donne pas de résultats satisfaisants. Voici par exemple un message à coder "ABCABC", le code RLC correspondant est "1A1B1C1A1B1C", ce qui conduit à un taux de compression négatif (6 - 16)/6, soit -166%. On s'aperçoit que le taux de compression négatif. Ainsi pour éviter cela, le codage RLC introduit un système de contrôle (bits) pour réaliser l'encodage. Il réalise le codage s'il y a répétition successive d'éléments (minimum égal à 4). Dans le cas contraire, il insert les bits contrôle (00).

1.4.3. Codage Lempel-Ziv

C'est une technique de codage qui utilise un dictionnaire. On cherche dans le fichier les chaînes qui se répètent, puis on mémorise dans le dictionnaire. Ensuite, le codage consiste à remplacer les chaînes mémorisées par leur adresse (ou indice) construite dans le dictionnaire. L'élaboration du dictionnaire ainsi que la



FIG. 1.3.: Schéma classique d'un système de compression avec perte.

recherche de chaîne répétée sont différentes selon la version de l'algorithme. Il en existe trois versions.

- LZ77, version originale, la recherche s'effectue par une fenêtre glissante;
- LZ78, la recherche s'effectue sur tout le fichier. La taille du dictionnaire est limitée en fonction du mode de codage (16, 32, ou 64 bits);
- LZW, introduite en 1984, qui est brevetée par la société Unisys, est une amélioration de la LZ78. Le dictionnaire, initialement construit, contient l'ensemble des codes ASCII. Il est élaboré au fur à mesure, ce qui permet de changer la taille du dictionnaire au cours du codage.

1.5. Codage avec perte

La Fig. 1.3 représente le schéma classique d'un système de compression avec perte.

Dans un premier temps, afin de mieux compacter l'information, la source est transformée en groupe de coefficients. Les transformations les plus utilisées, que ce soit pour les images fixes ou les séquences d'images, sont la Transformée en Cosinus Discrète (DCT), la Transformée en Ondelettes Discrète (DWT) ou la décomposition Pyramidale.

Dans un second temps, les coefficients obtenus après la transformation sont quantifiés (tronqués). La phase de quantification introduit l'erreur dans le système de codage.

La dernière étape consiste à coder les coefficients quantifiés par le codage entropique.

1.5.1. Codage par quantification

La quantification est l'une des sources de perte d'information dans le système de compression. Son rôle est en effet de réduire le nombre de bits nécessaire à la représentation de l'information. Elle est réalisée avec la prise en compte de l'aspect psychovisuel (l'œil humain), ce qui permet de déterminer la distorsion tolérable à apporter au signal à coder.



FIG. 1.4.: Quantification scalaire.

On distingue deux sortes de quantification : la quantification scalaire (QS) et la quantification vectorielle (QV).

1.5.1.1. Quantification scalaire (QS)

La quantification scalaire est réalisée indépendamment pour chaque élément. D'une manière générale, on peut la définir comme étant l'association de chaque valeur réelle x, à une autre valeur q qui appartient à un ensemble fini de valeurs. La valeur q peut être exprimée en fonction de la troncature utilisée : soit par l'arrondi supérieur, l'arrondi inférieur, ou l'arrondi le plus proche. On l'appelle le *"pas de quantification"* Δ . Δ est l'écart entre chaque valeur q. Arrondir la valeur x provoque une erreur de quantification, appelé le *"bruit de quantification"*. La valeur classique de ce dernier est $\frac{\Delta^2}{12}$.

La procédure suivante définit la réalisation d'une quantification scalaire. Soit X l'ensemble d'éléments d'entrée de taille N.

- 1. Echantillonner X en sous-intervalles $\{[x_n, x_{n+1}]/n \in \{0...N-1\}\}$
- 2. Associer à chaque intervalle $[x_n, x_{n+1}]$ une valeur q
- 3. Coder une donnée $x \in X$ par q si $x \in [x_n, x_{n+1}]$

Si Δ est constant, on parle d'une quantification uniforme. Sinon elle est dite non-uniforme. La Fig. 1.4 montre l'exemple d'une QS.

1.5.1.2. Quantification vectorielle (QV)

La quantification s'effectue sur un groupe d'éléments de la source, représenté par un vecteur \vec{x} de dimension n. La source est constituée par un ensemble fini de vecteurs \vec{x} . La QV consiste alors à remplacer le vecteur \vec{x} par un vecteur \vec{y} de même dimension appartenant à un dictionnaire [FRL94]. Le dictionnaire est un ensemble fini de vecteurs codes. Un vecteur \vec{x} codé, appelé classe est obtenu en faisant la moyenne itérative de vecteurs \vec{y} . La règle du plus proche voisin, au sens de la distance euclidienne entre deux vecteurs, est utilisée pour réaliser la quantification. La quantification vectorielle se décompose en général en deux parties : le processus de codage (codeur) et le processus de décodage (décodeur). Le processus de codage cherche l'adresse du vecteur \vec{y} correspondant dans le dictionnaire et l'envoie au récepteur. Le décodeur, quant à lui, dispose d'une réplique du dictionnaire et consulte celle-ci afin de reconstruire le vecteur code correspondant à l'adresse reçue. D'un point de vue mathématique, on peut définir la QV de la manière suivante :

Concernant le codeur, le processus de codage Q est défini par :

$$Q : \mathbb{R} \to I$$
$$\overrightarrow{x} \to Q(\overrightarrow{x}) \tag{1.4}$$

où I représente l'ensemble des indices correspondant au dictionnaire Y. Concernant le décodeur, le processus de décodage D est défini par :

$$D : I \to Y$$
$$i \to \overrightarrow{y} \tag{1.5}$$

L'élaboration du dictionnaire est donc une phase très importante pour la QV. Elle est faite à partir d'un processus d'apprentissage et peut être obtenue selon l'algorithme de LBG (Linde, Buzo, Gray). Dans le domaine de la compression d'image fixe, de nombreuses publications scientifiques ont été proposées pour élaborer le dictionnaire [PR01] [Del05].

1.5.2. Codage par prédiction

C'est la technique de compression la plus ancienne. On prédit la valeur du pixel à partir de la valeur précédemment codée. La prédiction peut se faire au moyen de l'histogramme de l'image. Seul l'écart entre la valeur réelle et la valeur prédite est quantifié puis codé et envoyé au décodeur. On peut réaliser la prédiction, au sein de l'image elle-même ainsi qu'entre images d'une séquence. Cette dernière est connue sous le nom de prédiction par compensation de mouvement. Le codage par prédiction est utilisé dans le codage Differential Pulse Code Modulation (DPCM).

1.5.3. Codage par transformée

La transformation des données d'entrée est faite afin de mieux compacter l'énergie de la transformée d'image sur un nombre faible de coefficients. La transformation a pour objet de décorréler les pixels d'image. On opère la transformation sur un bloc unitaire [Dav95] ou directement sur l'image entière.

1.5.3.1. Transformée de Karhunen-Loeve KLT

Soit $X = \{x_0, \dots, x_{N-1}\}$ le vecteur représentant un bloc. On modélise la corrélation entre les éléments dans ce bloc par sa matrice de covariance :

$$Y_{c} = E\left[(X - \mu_{X})(X - \mu_{X})^{T}\right]$$
(1.6)

où μ_X représente la moyenne du vecteur X. On note σ_{ij}^2 la covariance d'élément du vecteur X à la position (i, j). Un élément de la matrice de covariance n'est autre que σ_{ij}^2 :

$$Y_{c} = \begin{bmatrix} \sigma_{11}^{2} & \sigma_{12}^{2} & \cdots & \sigma_{1N}^{2} \\ \sigma_{21}^{2} & \sigma_{22}^{2} & \cdots & \sigma_{2N}^{2} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1}^{2} & \sigma_{N2}^{2} & \dots & \sigma_{NN}^{2} \end{bmatrix}$$
(1.7)

En normalisant la matrice Y_c et en supposant que la covariance ne dépend que de la distance entre les pixels, on peut trouver une expression, pour la matrice Y_c dont les valeurs propres λ_i forment une des fonctions de décomposition. On la note A. La transformation C = A.X définit la matrice de covariance diagonale par $Y = A.Y_c.A^T$

$$Y = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_N \end{bmatrix}$$

Trier les valeurs propres de la matrice par ordre décroissant permet la décorrélation totale des coefficients ainsi qu'une forte concentration d'énergie.

1.5.3.2. Transformée de Fourier Discrète DFT

La DFT permet de passer du domaine spatial au domaine fréquentiel. La DFT d'un signal discret x_n s'exprime par :

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(-i\frac{2\pi nk}{N}\right) \tag{1.8}$$

La fonction inverse permettant de remonter au signal original x_n connaissant sa transformée X_k est :

$$x_n = \sum_{k=0}^{N-1} X_k \exp\left(+i\frac{2\pi nk}{N}\right) \tag{1.9}$$

La DFT présente un coût de calcul très élevé. Ainsi, une variante de la transformée de Fourier dite rapide (FFT) a été développée pour assouplir ce coût.

1.5.3.3. Transformée en Cosinus Discrète DCT

La DCT est une variante de la DFT et il en existe plusieurs. Dans le cadre du signal mono-dimensionnel, la plus connue est celle-ci :

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right)$$
(1.10)

La DCT est très utilisée dans la compression d'image fixe, notament dans la norme JPEG. Dans cette application la DCT s'effectue sur un bloc de pixels 8×8 , et qui s'exprime par :

$$X_{u,v} = \frac{1}{4}C(u)C(v)\sum_{n=0}^{7}\sum_{m=0}^{7}x_{n,m}\cos\left(\frac{(2n+1)u\pi}{16}\right)\cos\left(\frac{(2m+1)v\pi}{16}\right) \quad (1.11)$$

avec

$$u, v = \{0 \cdots 7\}$$
 et $C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } w = 0\\ 1 & \text{sinon} \end{cases}$

1.5.3.4. Transformée en Ondelettes Discrètes DWT

Contrairement à la transformée de Fourier, la transformée en ondelettes permet de déterminer les différentes composantes fréquentielles d'un signal donné, ansi
que leur localisation spatiale ou temporelle. Par définition, les ondelettes sont des fonctions gérées à partir d'une ondelette mère Ψ , par dilatations et translations. Ainsi, la décomposition en ondelettes fait intervenir deux paramètres qui sont le facteur d'échelle *s* et le facteur de translation τ [Mal89] [Rio93] [Dau98].

Le paramètre d'échelle *s* permet d'obtenir des ondelettes à partir d'une ondelette mère, des ondelettes comprimées (support réduit) ainsi que des ondelettes dilatées (support étendu). Les ondelettes comprimées sont utilisées pour déterminer les composantes haute fréquence tandis que les ondelettes dilatées permettent de déterminer les composantes basse fréquence.

Le paramètre τ , quant à lui, permet d'analyser par translations successives le signal jusqu'à ce que celui-ci soit entièrement parcouru.

1.5.3.4.1. Cas d'un signal monodimensionnel

Dans le cas monodimensionnel, l'ondelette mère s'écrit :

$$\Psi_{s,\tau}(x) = \frac{1}{\sqrt{|s|}} \Psi\left(\frac{x-\tau}{s}\right) \tag{1.12}$$

Par cette transformée, on peut présenter l'information contenue dans une fonction notée f(x) de carré intégrable à une position τ et une échelle s. Dans le cas discret les valeurs de coefficients s et τ sont calculées de la manière suivante :

$$s = s_0^m \tag{1.13}$$

$$\tau = n\tau_0 s_0^m \tag{1.14}$$

avec $m, n \in \mathbb{Z}$ et $s_0 > 1, \tau_0 > 0$

Dans le cas particulier $s_0 = 2$ et $\tau_0 = 1$, il existe des ondelettes telles que s et τ peuvent être discrétisés de manière que les $\Psi_{m,n}(x)$ forment une base orthonormale.

$$\Psi_{m,n}(x) = 2^{-\frac{m}{2}} \Psi(2^{-m}x - n) \tag{1.15}$$

La décomposition en ondelettes d'une fonction f(x) peut s'effectuer selon la forme :

$$f = \sum_{m,n} C_{m,n}(f) \Psi_{m,n}$$
(1.16)

avec $C_{m,n}$, coefficient d'ondelette qui mesure les variations locales du signal. Son obtention est donnée par la relation suivante :



FIG. 1.5.: Ondelette de Haar.

$$C_{m,n}(f) = \int \Psi_{m,n}(x)f(x)dx \qquad (1.17)$$

De nombreuses sortes d'ondelettes ont été proposées dans la littérature [Dau92]. On peut recenser les plus utilisées. Les ondelettes de Morlet et de Sombrero sont des ondelettes continues, tandis que les ondelettes (orthogonales) de Haar, Shannon, Meyer, Spline et Daubechies sont des ondelettes discrètes.

1.5.3.4.2. Ondelette de Haar

C'est l'ondelette la plus simple. La fonction $\Psi(x)$ est définie par :

$$\Psi(x) = \begin{cases} 1 & x \in [0, \frac{1}{2}[\\ -1 & x \in [\frac{1}{2}, 1]\\ 0 & \text{ailleurs} \end{cases}$$
(1.18)

1.5.3.4.3. Ondelettes de Daubechies

Pour permettre une représentation complète et non redondante du signal, il faut que les ondelettes dilatées et translatées forment une base orthogonale de l'espace fonctionnel. I. Daubechies [Dau92] a élaboré une famille d'ondelettes dont le support est compact et orthogonal. Une ondelette est dite compacte (définie sur un intervalle donné) seulement si les filtres d'ondelettes associés ont une réponse impulsionnelle finie. Autrement dit, ce sont des filtres à p moments nuls. Ils sont réalisés avec un filtre d'échelle et un filtre d'ondelettes ayant une taille de 2p. Les coefficients des filtres sont appelés *coefficients de filtres en ondelettes*. Pour p = 2, on a 4 coefficients. On retrouve l'ondelette de Haar dans le cas où p = 1.



FIG. 1.6.: Ondelette de Daubechies.

Les fonctions d'ondelettes de Daubechies ne sont pas exprimables d'une façon analytique car la construction d'une fonction d'échelle et d'une fonction ondelette nécessite une itération. Dans le cas p = 2, les valeurs des coefficients des filtres sont :

$$c_{0} = \frac{1+\sqrt{3}}{4\sqrt{2}} \qquad c_{2} = \frac{3-\sqrt{3}}{4\sqrt{2}} c_{1} = \frac{3+\sqrt{3}}{4\sqrt{2}} \qquad c_{3} = \frac{1-\sqrt{3}}{4\sqrt{2}}$$
(1.19)

1.5.3.4.4. Cas d'un signal bi-dimensionnel

D'après Mallat [Mal89], dans le cas d'un espace à deux dimensions, la fonction d'ondelette monodimensionnelle Ψ avec la fonction d'échelle φ peut être séparée en trois fonctions d'ondelettes distinctes (Ψ_1, Ψ_2, Ψ_3) :

- $\Psi_1(x, y) = \varphi(x)\Psi(y)$ exprime les variations selon l'axe horizontal;
- $\Psi_2(x,y) = \Psi(x)\varphi(y)$ exprime les variations selon l'axe vertical;

 $-\Psi_3(x,y) = \Psi(x)\Psi(y)$ exprime les variations selon les deux axes (diagonal).

Il existe une technique appelée *"transformée standard"*, qui permet de réaliser la transformée en ondelettes à deux dimensions. En effet la technique consiste à calculer la transformée en ondelettes mono-dimensionnelle pour chaque élément de la ligne puis d'appliquer la transformée en ondelettes mono-dimensionnelle pour chaque élément de la colonne.

D'après l'Eq. 1.16, la transformation en ondelettes peut être considérée comme une projection du signal f sur ces différents sous-espaces. Cette décomposition peut s'obtenir simplement par des convolutions du signal avec des filtres miroirs en quadrature. A chaque niveau, on filtre le signal par des filtres d'analyse passe-



FIG. 1.7.: Décomposition en ondelettes 2D en sous-bandes pour un niveau de résolution.

haut (notés h_H) et passe-bas (notés h_L), et on sous-échantillonne avec un facteur de 2. Tendance (approximation du signal à un niveau) et fluctuations (information de différence de contenu, extraite entre deux niveaux) au niveau courant, ont un support deux fois moindre qu'au niveau précédent. Ainsi, après la décomposition, on obtient quatre matrices chacune de taille quatre fois plus petite. D'où l'obtention de la représentation multi-résolution dans laquelle on a quatre sous-bandes.

On les désigne par LL, LH, HL et HH. La lettre H correspond au filtrage passe-haut et la lettre L à celui du passe-bas appliqué de façon séparable sur les lignes et les colonnes (Fig. 1.7).

Ainsi, pour reconstruire le signal original, on applique une succession de filtres conjugués, filtres de synthèse, en sur-échantillonnant avec un facteur de 2 le signal décomposé (Fig. 1.8). Dans cette figure, le filtre d'analyse est représenté par le couple (h_L, h_H) tandis que le couple (g_L, g_H) représente le couple du filtre de synthèse.

Afin d'obtenir une reconstruction parfaite, les deux bancs de filtres d'analyse et de synthèse doivent satisfaire la relation ci-après.

Soit $H_L(z)$, $G_L(z)$, $H_H(z)$, et $G_H(z)$ les transformées en Z des filtres h_L , g_L , h_H , g_H respectivement :

$$H_L(z)G_L(z) + H_H(z)G_H(z) = 2 (1.20)$$

$$H_L(-z)G_L(z) + H_H(-z)G_H(z) = 0 (1.21)$$



FIG. 1.8.: Les bancs de filtres : analyse et synthèse.



ginale

(c) niveau 2

FIG. 1.9.: Exemple d'une décomposition en ondelettes à 2 niveaux de résolution.

En choisissant $G_L(z) = -cz^{-l}H_H(-z)$ et $G_H(z) = cz^{-l}H_L(-z)$, où l et c sont des constantes, on peut déterminer la condition permettant la reconstruction du signal par :

$$-cz^{-l}H_L(z)H_H(-z) + cz^{-l}H_H(z)H_L(-z) = 2$$
(1.22)

Cette condition est acquise lorsque h_L et h_H d'une part ainsi que g_L et g_H d'autre part sont orthogonaux entre eux. Ces sont des bancs de filtres bi-orthogonaux. La Fig. 1.9 montre l'exemple d'une décomposition dyadique à deux niveaux de résolution.

1.5.3.4.5. La technique lifting

La technique "lifting" [Swe95], dite ondelettes de deuxième génération [Swe98], permet d'effectuer la transformée en ondelettes sans le banc de filtres. En effet, le banc de filtres est remplacé par un certain nombre d'étapes : la prédiction, la mise à jour et la mise en échelle des coefficients [DS96]. Cette technique permet

non seulement de réduire la complexité de l'implantation comme la réduction de la quantité de mémoire ou le temps de calcul, mais permet aussi la reconstruction parfaite du signal. Le processus est effectué successivement par une procédure bien spécifique.

La première étape consiste à transformer le signal d'entrée x_i en séquences d'échantillonnage : échantillonnage pair et échantillonnage impair. Cette transformation est réalisée à l'aide d'une transformée paresseuse "Lazzy Transform". On utilise la même notation que celle de [RJ02], en notant respectivement s_i^n et d_i^n les séquences paire et impaire, où $n \in [1, N]$ est le nombre d'itérations effectuées, qui dépend du type d'ondelettes utilisées. On peut ainsi écrire :

$$s_i^n = x_{2i} \tag{1.23}$$

$$d_i^n = x_{2i+1} (1.24)$$

L'étape suivante consiste en la prédiction et la mise à jour de chaque échantillonnage. D'abord, les échantillons impairs d_i^n sont prédits à partir des échantillons pairs voisins s_i^n . La prédiction se fait avec une combinaison linéaire de s_i^n .

Ensuite la mise à jour de s_i^n est réalisée avec les séquences prédites précédemment par une pondération :

$$d_i^n = d_i^{n-1} + \sum_k P_n(k) s_k^{n-1}$$
(1.25)

$$s_i^n = s_i^{n-1} + \sum_k U_n(k) d_k^n$$
 (1.26)

où $P_n(k)$ et $U_n(k)$ sont respectivement des coefficients (poids) de prédiction et de mise à jour associés à l'itération n. Dans le cas d'ondelettes de Daubechies (9,7), on a N = 2, tandis que pour les ondelettes de Daubechies (5,3) on a N = n = 1. Pour les ondelettes de Daubechies (5,3), l'Eq. 1.25 et l'Eq. 1.26 peuvent s'écrire ainsi :

$$d_i^1 = d_i^0 - \frac{1}{2} \left(s_i^0 + s_{i+1}^0 \right) \tag{1.27}$$

$$s_i^1 = s_i^0 + \frac{1}{4} \left(d_{i-1}^1 + d_i^1 \right) \tag{1.28}$$

L'étape finale est la mise en échelle des coefficients obtenus précédemment par K_0 et K_1 . Celle-ci permet d'ajuster les amplitudes des coefficients. Dans le cas



FIG. 1.10.: Processus de la technique lifting.

d'ondelettes de Daubechies (5,3), on a $K_0 = K_1 = 1$, tandis que $K_1 = 1/K_0$ avec $K_0 = 1.230174104914001$ pour les ondelettes de Daubechies (9,7).

On obtient ainsi les filtres passe-bas et passe-haut correspondant au signal d'entrée. La Fig. 1.10 représente le diagramme du processus de la technique lifting.

1.5.3.5. Décomposition pyramidale

Introduite pour la première fois par Burt et Adelson [BA83] sous le nom de pyramide gaussienne, la décomposition pyramidale consiste en une représentation multi-échelle de l'image. On observe l'image à différentes résolutions. Le passage à une résolution supérieure est effectué par un filtrage passe-bas selon chaque direction ligne et colonne, ce qui permet d'obtenir une résolution de l'image réduite de moitié. Ce processus est appelé *réduction*. Soient $G_0, G_1 \dots G_n$ la pyramide gaussienne réalisant chaque résolution ; on a :

$$G_{l}(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) G_{l-1}(2i+m,2j+n) \qquad l = 1, N$$
(1.29)

où N représente le nombre de la pyramide. La fonction *noyau générateur*, réalisée à l'aide d'une fenêtre de taille 5×5 doit satisfaire les conditions suivantes :

$$w(m,n) = v(m)v(n) \tag{1.30}$$

$$\sum_{m=-2}^{2} v(m) = 1 \tag{1.31}$$

$$v(m) = v(-m)$$
 pour $m = 0, 1, 2$ (1.32)

En plus de ces trois contraintes, on a ajouté une nouvelle contrainte appelée égale contribution. Si v(0) = a, v(-1) = v(1) = b et v(-2) = v(2) = c. L'égale contribution se traduit par :

$$a + 2c = 2b \tag{1.33}$$

On satisfait cette condition en prenant :

$$v(0) = a \tag{1.34}$$

$$v(-1) = v(1) = 1/4 \tag{1.35}$$

$$v(-2) = v(2) = 1/4 - a/2 \tag{1.36}$$

L'opération inverse (expensée) qui permet de passer d'une résolution inférieure à une résolution supérieure se fait à l'aide d'une interpolation des pixels. Cette dernière consiste à grandir une image de taille (M + 1) à une image de taille (2M + 1):

$$G_{l}(i,j) = 4\sum_{m=-2}^{2}\sum_{n=-2}^{2}w(m,n)G_{l+1}\left(\frac{i+m}{2},\frac{j+n}{2}\right)$$
(1.37)

On note que dans cette expression (Eq. 1.37) la fonction w(.) ne contribue pas à l'élaboration de la somme.

Cette représentation en multi-échelle par la pyramide gaussienne sert à prédir la valeur d'intensité de pixels de l'image initiale (résolution G_0) aux différentes résolutions $G_1, G_2 \cdots G_N$. La compression d'image consiste alors à coder la différence de la valeur d'intensité de pixels entre deux résolutions successives, respectivement 2^l et 2^{l+1} que l'on appelle décomposition pyramidale laplacienne et définie par :

$$L_{l}(i,j) = G_{l}(i,j) - G_{l+1}(i,j)$$
(1.38)

où G_{l+1} est l'image expensée à la résolution 2^{l+1} .

La fonction poids associée à la pyramide laplacienne constitue donc une différence de deux fonctions gaussiennes, connue sous le nom DOG.

Le décodage ou reconstruction d'image finale est construite à l'aide d'une somme successive :

$$G_0(i,j) = \sum_{l=0}^{N} L_l(i,j)$$
(1.39)



FIG. 1.11.: Décomposition pyramidale gaussienne de niveaux 3.

La Fig. 1.11 montre l'exemple d'une décomposition pyramidale.

1.6. Codage d'images fixes : standards JPEG

De nombreux organismes internationaux s'occupent de la normalisation des systèmes de codage pour des images fixes : l'Organisation Internationale de Normalisation (ISO) et la Commission Electrotechnique Internationale (IEC). On s'intéressera aux travaux du groupe *Joint Photographic Experts Group* (JPEG) qui est chargé des spécifications, de l'évolution du standard d'image fixe de l'ISO/IEC. Les normes de cette famille sont connues sous les noms JPEG et JPEG2000.

La technique utilisée pour l'encodage d'images fixes peut être ramenée à deux grandes familles : méthode par transformée (cf. §1.5.3) et technique structurelle (recherchant l'homogénéité dans l'image comme : les textures, les contours, histogramme). L'approche structurelle emploie les techniques qui manipulent la valeur d'intensité du pixel dans l'image. Les deux grandes familles peuvent être utilisées ensemble pour un système de codage. Autrement dit, il n'existe pas de frontière entre l'approche par transformée et l'approche structurelle. On appelle taux de compression le rapport entre l'image brute et l'image compressée.

1.6.1. JPEG

Norme de compression d'image fixe, établie en 1991, JPEG permet la compression sans perte (JPEG-LS) et avec perte d'information. C'est une norme qui a eu beaucoup de succès depuis plus de dix ans et que nous utilisons toujours aujourd'hui. Elle s'impose surtout dans le domaine d'archivage d'images naturelles.



FIG. 1.12.: Schéma-bloc de JPEG.

1.6.1.1. Principe

L'encodage par JPEG est réalisé en quatre étapes successivement : transformée en cosinus discrète DCT, phase de quantification, réorganisation en zig-zag, suivie de codage RLC et codage de Huffman (Fig. 1.12).

1.6.1.2. DCT

La première étape consiste à découper l'image originale en blocs de taille 8×8 . Ensuite pour chaque bloc, on applique la DCT. On a donc une matrice dont les composantes sont les coefficients de la transformée DCT. Ainsi, on obtient deux sortes de coefficients, DC et AC. Le coefficient DC représente la moyenne des pixels appartenant au bloc courant (premier élément de la matrice transformée), les éléments restants sont des coefficients AC.

1.6.1.3. Quantification

C'est la deuxième étape. Cette phase consiste à donner les valeurs approximatives de la matrice précédente. Pour cela, une matrice de quantification est construite. Elle doit prendre en compte l'aspect psychovisuel, par exemple : l'œil humain est un filtre passe-bas.

1.6.1.4. Réorganisation en zig-zag

L'objet de cette troisième étape est d'obtenir le maximum de suites de zéros. En effet, la norme JPEG traite les valeurs zéro (coefficients haute-fréquence) de la matrice quantifiée en raison de leur nombre important. Ceci concerne seulement les coefficients AC du bloc. Le coefficient DC, quant à lui, est codé par le codage différentiel. Ce dernier consiste à coder la différence entre les deux coefficients DC successifs courant et précédent.

1.6.1.5. Codage entropique

La dernière étape est le codage de Huffman des coefficients obtenus précédemment. Ainsi, on a une image compressée en JPEG.



(a) image originale



(b) image JPEG; t=60

(c) image JPEG2000;t=60

FIG. 1.13.: Dégradation d'image JPEG à un taux de compression élevé.

1.6.1.6. Limite du standard JPEG

La norme JPEG ne permet pas un taux de compression élevé supérieur à t = 32. Au delà, des artefacts très gênants sont brusquements présents dans l'image codée (effets obtenus avec la DCT, voir Fig. 1.13-b).

1.6.2. JPEG2000

1.6.2.1. Performance de JPEG2000

JPEG2000 est une norme de compression d'image fixe. Introduite en mars 1997 par l'ISO/IEC, elle est devenue standard en décembre 2000. L'objectif de la norme est de compléter la performance du standard JPEG mais sans la remplacer [BS02]. Par rapport à JPEG, à la place de la transformée en cosinus discrète (DCT), le standard JPEG2000 utilise la transformée en ondelettes discrète (DWT). Au lieu



FIG. 1.14.: Schéma-bloc de JPEG2000.

du codage de Huffman, la norme JPEG2000 emploie un codage entropique basé sur le codage arithmétique par plan de bits (EBCOT) [Tau00][TOWS02]. Ce dernier permet l'accès aléatoire, la parallélisation, la flexibilité et la résistance aux erreurs. Tandis que la DWT permet la scalabilité, obtenue grâce à la représentation multi-résolution. Voici les principaux avantages de JPEG2000 par rapport à JPEG [CSE00] [Gro03] :

- Performances supérieures à même taux de compression. Celles-ci sont d'autant plus grandes que le taux de compression est très élevé (taux de compression supérieur à t = 32, Fig. 1.13-b);
- Possibilité d'avoir deux modes de compression : sans ou avec perte;
- Transmission et reconstruction progressives de l'image : spatiale (taille) ou qualité (visuelle);
- Robustesse aux erreurs en paquets pour applications mobiles à très bas débit;
- Scalabilité en résolution;
- Gestion de régions d'intérêt.

Ces avantages sont acquis au prix d'un coût d'implantation (calculs et mémoire) de 3 à 6 fois plus élevé par rapport au standard JPEG.

1.6.2.2. Architecture du standard

Le standard JPEG2000, ISO/IEC FCD15444 [1.000], comporte 12 parties. La première partie Part-1, cœur du système, décrit le fonctionnement de base du standard. C'est la partie qui nous intéresse. Le schéma-bloc JPEG2000 est donné par la Fig. 1.14. Chaque bloc est détaillé ci-après.

1.6.2.3. Découpage en tuiles

L'image originale est découpée en petits blocs rectangulaires appelés tuiles dont les dimensions correspondent à des puissances de 2 (16, 32, 64, etc). Ainsi obtenues, les tuiles ont les mêmes dimensions, mises à part, éventuellement, les tuiles constituant les bords à droite et en bas de l'image. Chaque tuile est codée indépendamment, ce qui réduit la quantité de mémoire nécessaire à l'encodage; en contre-partie, lorsque le taux de compression est assez fort, ce découpage provoque une dégradation de la qualité d'image codée. La tuile est donc l'unité d'information élémentaire traitée par l'encodeur standard JPEG2000. Dans notre application, on n'a utilisé qu'une seule tuile.

1.6.2.4. Transformées couleur

L'étape suivante est la transformée couleur. Cette phase permet de décorréler les composantes couleurs. La transformée couleur, étape optionnelle dans la norme JPEG2000, ne peut être réalisée que si les composantes couleurs ont les mêmes dynamiques. Comme on a 3 composantes couleurs (R,G,B), la transformée couleur s'applique à chacune de ces composantes. La norme propose deux modes de transformée : transformée couleur réversible ou irréversible.

1.6.2.4.1. Transformée réversible

La transformée couleur réversible permet la compression avec ou sans perte. En effet la transformée $RGB \rightarrow YUV$ se fait avec une transformée linéaire dont les coefficients des matrices directe et inverse sont des coefficients fractionnaires. Ainsi, la transformée $YUV \rightarrow RGB$ est l'exacte inverse de la transformée $RGB \rightarrow$ YUV. Les coefficients de la matrice T directe et de l'inverse $A = T^{-1}$ sont les suivants :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = T \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{avec} \quad T = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \quad (1.40)$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = A. \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \quad \text{avec} \quad A = \begin{bmatrix} 1 & \frac{3}{4} & -\frac{1}{4} \\ 1 & -\frac{1}{4} & -\frac{1}{4} \\ 1 & -\frac{1}{4} & \frac{3}{4} \end{bmatrix}$$
(1.41)

1.6.2.4.2. Transformée irréversible

La transformée couleur irréversible permet seulement la compression avec perte. En effet la transformée $RGB \rightarrow YCbCr$ se fait avec une transformée

TAB.	1.2.:	Les	coefficients	des	filtres	d'analyse	et	de	synthèse	des	ondelettes	de
		Dau	ubechies (5,3	3).								

i	Passe-bas $h_L(i)$	Passe-haut $h_H(i)$	Passe-bas $g_L(i)$	Passe-haut $g_H(i)$
0	6/8	1	1	6/8
±1	2/8	-1/2	1/2	-2/8
± 2	-1/8			-1/8

linéaire dont les coefficients des matrices directe et inverse sont des coefficients réels à virgule fixe. Ainsi, la transformée $YCbCr \rightarrow RGB$ n'est pas l'exacte inverse de la transformée $RGB \rightarrow YCbCr$. Les coefficients de la matrice directe T et de l'inverse A sont les suivants :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = T. \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \text{avec} \quad T = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix}$$
(1.42)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = A. \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \quad \text{avec} \quad A = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \quad (1.43)$$

1.6.2.5. Décomposition en ondelettes

La norme JPEG2000 utilise la technique *lifting*, et accepte seulement la décomposition dyadique à 5 niveaux de résolution, c'est-à-dire que pour chaque résolution on effectue la décomposition à partir de la sous-bande LL résultant des deux filtrages successifs passe-bas de la ligne et de la colonne du signal décomposé. Le standard utilise deux types d'ondelettes : les ondelettes de Daubechies (9,7) et de Daubechies (5,3). Le couple (9,7) veut dire qu'on a 9 coefficients du filtre passe-bas et 7 coefficients du filtre passe-haut (voir Tab. 1.3). Ces deux ondelettes sont choisies selon le type de la compression souhaitée, sans perte ou avec perte. Les ondelettes de Daubechies (5,3) sont utilisées pour la compression sans ou avec perte, tandis que les ondelettes de Daubechies (9,7) sont utilisées uniquement pour la compression avec perte. Ainsi, les ondelettes (5,3) sont associées à la transformée couleur réversible ($RGB \rightarrow YUV$) et la transformée couleur ($RGB \rightarrow YCbCr$) est associée à celle des ondelettes (9,7).

TAB. 1.3.: Les coefficients des filtres d'analyse et de synthèse des ondelettes de Daubechies (9,7).

i	Passe-bas $h_L(i)$	Passe-haut $h_H(i)$
0	0.6029490182363579	1.115087052456994
±1	0.2668641184428723	-0.5912717631142470
± 2	-0.07822326652898785	-0.05754352622849957
±3	-0.01686411844287495	0.09127176311424948
± 4	0.02674875741080976	

i	Passe-bas $g_L(i)$	Passe-haut $g_H(i)$
0	1.115087052456994	0.6029490182363579
±1	0.5912717631142470	-0.2668641184428723
± 2	-0.05754352622849957	-0.07822326652898785
± 3	-0.09127176311424948	0.01686411844287495
± 4		0.02674875741080976



FIG. 1.15.: Quantification uniforme avec zone morte.

1.6.2.6. Quantification

Dans le système de compression en général, la quantification est l'étape qui réalise la compression des données. Elle permet en effet la troncature de tous les coefficients d'ondelettes dans chaque sous-bande. La norme JPEG2000 opte pour une quantification scalaire uniforme (cf. Fig. 1.4-b) avec une zone morte "deadzone". La zone morte permet de mettre à zéro les coefficients de faible amplitude. En conséquence, le pas de quantification Δ_b est doublé (Fig. 1.15). La quantification $q_b(u, v)$ des coefficients $a_b(u, v)$ de chaque sous-bande b est effectuée selon la formule :

$$q_b(u,v) = sign(a_b(u,v)) \left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor$$
(1.44)

où $\lfloor a \rfloor$ représente l'arrondi inférieur. Le pas de quantification \triangle_b est différent d'une sous-bande à l'autre. La relation permettant de définir le pas de quantification est :

$$\Delta_b = 2^{R_b - \varepsilon_b} (1 + \frac{\mu_b}{2^{11}}) \tag{1.45}$$

où R_b représente la dynamique des coefficients de la sous-bande, et correspond au nombre de bits utilisés pour coder la composante. μ_b et ε_b sont la mentisse et l'exposant du pas de quantification. D'après la relation (Eq. 1.44), plus le pas de quantification est grand, plus la quantité de données est réduite, c'est-à-dire que la perte d'information est d'autant plus grande. Dans le cas d'une compression sans perte, on utilise une simple troncature par arrondi. Ceci est obtenu en fixant la valeur de $\Delta_b = 1$. Mais, dans la deuxième partie de la norme JPEG2000, le pas de quantification est modifiable au niveau du codeur afin de prendre en compte des phénomènes de masquage du système visuel humain, comme la fonction de sensibilité au contraste CSF, la sensibilité aux fréquences spatiales, la pondération fréquentielle, la quantification non uniforme. Aussi en plus de la troncature, il est possible d'utiliser une quantification scalaire par division pour effectuer une compression avec perte.

La dynamique maximale des coefficients dans les sous-bandes est définie par la relation suivante :

$$M_b = G + \varepsilon_b - 1 \tag{1.46}$$

où G est un bit de garde.

Le décodeur réalise la déquantification en utilisant le pas de quantification stocké dans le flux de données codées. En effet le pas de quantification est signalé dans l'en-tête concernant la quantification QCD ("Quantization Default"). Le stockage peut être réalisé par différents modes. Si le pas de quantification Δ_b est identique pour toutes les bandes, seule la valeur de Δ_b est signalée dans l'en-tête. En ce qui concerne la quantification implicite, la mentisse et l'exposant, respectivement μ_b et ε_b , sont signalés dans la bande *LL*. Le pas de quantification de la résolution suivante est obtenu en multipliant par deux le pas de quantification de la résolution en cours. Enfin si la quantification est effectuée de façon explicite, le couple (μ_b, ε_b) est signalé dans chaque sous-bande.

Au décodeur, l'opération permettant d'effectuer la déquantification pour la compression avec perte par les ondelettes de Daubechies (9,7) est :

$$Rq_b(u,v) = \begin{cases} (q_b(u,v) + \gamma) \triangle_b & \text{si } q_b(u,v) > 0\\ (q_b(u,v) - \gamma) \triangle_b & \text{si } q_b(u,v) < 0\\ 0 & \text{ailleurs} \end{cases}$$
(1.47)

où γ est un paramètre de reconstruction qui peut être choisi arbitrairement par le décodeur. Dans le cas d'une compression sans perte on a $Rq_b(u, v) = q_b(u, v)$.

1.6.2.7. Codage entropique

1.6.2.7.1. Principe

Chaque coefficient quantifié est représenté par un indice. Ainsi, le codage entropique consiste à exploiter ces indices par une technique basée sur l'utilisation de contextes. Le codage est réalisé par plans de bits en plusieurs passes. Mais il importe avant tout que les différentes sous-bandes des coefficients soient divisées en blocs rectangulaires, appelés "code-blocs" (comme le montre la Fig. 1.16), lesquels constituent les unités d'information traitées lors de la phase de codage. La largeur et la hauteur d'un bloc, qui sont paramétrables, doivent être comprises entre 4 et 1024. La taille typiquement utilisée est 64×64 . Le total des blocs ne doit pas dépasser 4096. Ensuite, les blocs sont codés indépendamment les uns des autres. Ceci permet d'une part une robustesse aux erreurs et d'autre part, au décodeur, d'effectuer en parallèle le décodage de tous les blocs codés.

Au cours de cette phase de codage, des points de troncature sont déterminés dans le flux de sortie. Ces points de troncature servent à indiquer quelles parties du flux de bits peuvent être supprimées afin de respecter le taux de compression spécifié, et ce en assurant une qualité d'image optimale [LNR02].

1.6.2.7.2. Le codage EBCOT

JPEG2000 utilise l'algorithme EBCOT (*Embedded Block Coding with Opti*mised Truncation) pour réaliser le codage entropique. Cet algorithme, cœur de la norme qui est développée dans [Tau00] et [TOWS02], est conceptuellement réalisé en deux étapes. La première étape, bloc Tier 1 (Fig. 1.14) effectue la modélisation du contexte et l'encodage entropique, tandis que la deuxième étape génère l'allocation de bits de sortie. On représente dans la Fig. 1.17 la schématique du bloc Tier 1. Le contexte est modélisé avec le codage par primitives (*Zero coding, Run-Length Coding, Sign Coding,* et Magnitude Refinement), et la propriété de voisinage (similarité statistique des coefficients voisins). D'une autre manière, on peut définir un contexte comme étant un groupe d'éléments possédant des propriétés statistiques identiques. Le codage arithmétique (MQ-Coder), quant à lui,



FIG. 1.16.: Découpage en blocs rectangulaires; ici les deux résolutions ont la même taille de blocs.

est un codage entropique basé sur une division récursive des intervalles de probabilité d'Elias [Imp04].

Etant divisés en blocs de coefficients, les coefficients d'ondelettes subissent ensuite le codage par plans de bits (Fig. 1.18). Le codage se fait en commençant par les bits de poids le plus fort MSB ("*Most Significant Bit*") vers les bits de poids le plus faible LSB ("*Least Significant Bit*"). Lors du parcours de chaque plan de bit, un coefficient d'ondelettes quantifié est dit significatif si son indice reste égal à 1.

Le codage par plans de bits est effectué en trois passes : passe de codage *si-gnificance propagation*, passe de codage *magnitude refinement* et passe de codage *cleanup*. Chaque passe constitue une unité de codage. Cette dernière est ensuite regroupée en différentes couches de qualité dans lesquelles est définie l'importance des coefficients (voir Fig. 1.19). Les différentes couches ainsi définies fournissent les informations permettant de fixer le nombre de bits à envoyer au décodeur, ce qui permet la transmission progressive (par exemple : voir en premier la région d'intérêt).

JPEG2000 emploie un concept de *contexte significatif* pour réaliser les trois passes de codage. Le concept signifie que pour un bit insignifiant (ou non signifiant) à la position j, le contexte associé est devenu significatif si $K^{sig}[j]$ est supérieur à zéro. Le contexte $K^{sig}[j]$, appelé état du contexte significatif (voir [TM02], page 355 pour plus détail), est obtenu en fonction de trois variables $K^h[j], K^v[j], K^d[j]$, Fig. 1.20.



Système de codage EBCOT

FIG. 1.17.: Construction de la première étape Tier 1 du EBCOT.



FIG. 1.18.: Codage par plans de bits à 4 profondeurs.



FIG. 1.19.: Regroupement des blocs dans différentes couches de qualité.



FIG. 1.20.: Détermination du contexte du bit significatif.

Pour chaque codage de passe on utilise les variables suivantes :

 $\nu[j]$, définit la valeur du bit j;

 $\sigma[j]$, définit la significance du bit j;

 $\pi[j]$, définit l'appartenance du bit j au codage de passe "significance propagation".

On désigne par MQ-Coder, le codeur arithmétique et CoderSign le codeur de signe.

Passe de codage "significance propagation"

C'est la première des trois phases de codage. Lors de cette phase, les coefficients (représentés par les indices de quantification) sont parcourus selon la Fig. 1.21. D'abord les coefficients sont groupés dans une bande composée de quatre lignes. Ensuite, on les parcourt de haut en bas. Dans le sens de la largeur les coefficients sont parcourus de gauche à droite. Le codage "significance" est réalisé si, et seulement si, le bit de la $j^{ème}$ position est insignifiant et que ses voisins sont significatifs c'est-à-dire $\sigma[j] = 0$ et $K^{sig}[j] > 0$. Ainsi après le codage, la variable $\pi[j]$ est mise à jour pour signaler que le coefficient en cours a été encodé par cette passe de code. Lorsque le bit est devenu significatif, on code le signe. Cette passe de code s'effectue pour chaque plan de bits et l'algorithme permettant l'encodage "significance propagation" est donné par l'Algo. 1.

Passe de codage "magnitude refinement"

La deuxième phase de codage est le "magnitude refinement". Elle est réalisée seulement pour les bits encodés précédemment et qui sont devenus significatifs.



FIG. 1.21.: Parcours des coefficients dans un bloc.

L'algorithme qui permet l'encodage de "magnitude refinement" est donné par l'Algo. 2. La relation qui permet d'obtenir le couple $(\sigma^{new}[j], K^{mag}[j])$ est donnée dans [TM02], page 360.

```
1: pour chaque position j faire
       si \sigma[j] = 0 et K^{sig}[j] > 0 alors
 2:
          MQ-Coder(\nu[j], K^{sig}[j])
 3:
 4:
           si \nu[j] = 1 alors
 5:
              \sigma[j] \leftarrow 1
              CoderSign()
 6:
           fin si
 7:
           \pi[j] \leftarrow 1
 8:
       sinon
9:
           \pi[j] \leftarrow 0
10:
11:
       fin si
12: fin pour
```

Algo. 1: Pseudo-code "significance propagation".

Passe de codage "cleanup"

Cette dernière phase de codage a pour objet de diminuer le nombre total de bits à coder. Elle s'applique sur le reste des coefficients du bloc qui n'ont pas été codés par les deux passes de codage précédentes. En principe, ce sont des coefficients qu'on souhaite rendre non significatifs. Pour cela on utilise deux modes de codage : mode normal ou mode "run-length". On les associe respectivement aux variables de signification du contexte K^{uni} et K^{run} . L'objectif consiste à regrouper les bits se trouvant sur la même colonne j_1 d'une bande, Fig. 1.21, dont les bits ne sont pas significatifs par rapport aux bits voisins qui également ne sont pas significatifs. j_1 et j_2 sont respectivement la largeur et la hauteur de la bande (groupes de coefficients, à ne pas confondre avec les sous-bandes de la transformée en ondelettes). L'algorithme permettant l'encodage dans la passe de codage "*cleanup*" est donné par l'Algo. 3.

Codage de signe CoderSign

Le signe est codé immédiatement si le bit est devenu significatif lors du codage de passe "significance propagation". Le codage s'appuie sur la configuration de ses 4 coefficients voisins. La configuration doit se trouver parmi les trois cas de figures suivants : configuration significative et positive; significative et négative; ou non significative. Soient, $\chi^h[j]$ et $\chi^v[j]$ qui représentent respectivement le signe des coefficients voisins en sens horizontal et en sens vertical :

$$\chi^{h}[j] = \chi[l, c-1] + \sigma[l, c-1] + \chi[l, c+1] + \sigma[l, c+1]$$
(1.48)

$$\chi^{v}[j] = \chi[l-1,c] + \sigma[l-1,c] + \chi[l+1,c] + \sigma[l+1,c]$$
(1.49)

où $\chi[j]$ représente le signe du coefficient courant.

1: pour chaque position j faire 2: si $\sigma[j] = 1$ et $\pi[j] = 0$ alors 3: Trouver $K^{mag}[j]$ 4: MQ-Coder $(\nu[j], K^{mag}[j])$ 5: $\sigma^{new}[j] \leftarrow \sigma[j]$ 6: fin si 7: fin pour

Algo. 2: Pseudo-code "magnitude refinement".

1.6.2.7.3. Codage arithmétique MQ-Coder

Le codage arithmétique proposé dans la norme JPEG2000 est un codage basé sur une subdivision d'intervalles. Etant initialement développé pour le standard JBIG pour une application à l'image binaire (possédant deux niveaux), le codeur arithmétique MQ-Coder comporte deux entrées : le contexte et la décision (voir Fig. 1.17). Chaque décision binaire, représentée par un bit, est divisée récursivement. Les divisions sont faites pour estimer la probabilité d'Elias : MPS *("Most Probable Symbol")* et LPS *("Less Probable Symbol")*. En notant A la longueur d'intervalle courante, l'estimation des deux sortes d'intervalles est donnée par les expressions suivantes :

Intervalle MPS :

```
1: si l \mod 4 = 0 et l \le j_1 - 4 alors
        r \leftarrow -1
 2:
        si K^{sig}[j_l + i, j_c] = 0 pour tout i \in \{0, 1, 2, 3\} alors
 3:
           r \leftarrow 0
 4:
           tant que r < 4 et \nu[j_1 + r, j_2] = 0 faire
 5:
              r \gets r+1
 6:
              si r = 4 alors
 7:
                  MQ-Coder(0, K^{run})
 8:
              sinon
 9:
                  MQ-Coder(1, K^{run})
10:
                  \begin{array}{l} \text{MQ-Coder}(\lfloor \frac{l}{2} \rfloor, K^{uni}) \\ \text{MQ-Coder}(l \ mod \ 2, K^{uni}) \end{array}
11:
12:
              fin si
13:
           fin tant que
14:
        fin si
15:
16: fin si
17: si \sigma[j] = 0 et \pi[j] = 0 alors
        si r \ge 0 alors
18:
           l \leftarrow r-1
19:
        sinon
20:
           MQ-Coder(\nu[j], K^{sig}[j])
21:
22:
        fin si
        si \nu[j] = 1 alors
23:
           \sigma[j] \gets 1
24:
           CoderSign()
25:
        fin si
26:
27: fin si
```

Algo. 3: Pseudo-code "cleanup".



FIG. 1.22.: Subdivision d'intervalle d'Elias du MQ-Coder.

$$A - A \times Qe \tag{1.50}$$

Intervalle LPS :

$$A \times Qe$$
 (1.51)

où Qe représente la probabilité LPS (Fig. 1.22).

Dans JPEG2000, la réalisation du codeur arithmétique est effectuée par l'intermédiaire d'une table d'index. La table représente l'estimation de la probabilité LPS (Qe). Pour chaque couple d'entrée (décision, contexte), on cherche le symbole le plus probable dans une variable comportant les différents états. Comme chaque état est représenté dans le tableau d'index, on peut associer le contexte à l'index du tableau. De son côté, le décodeur dispose de la réplique d'index du tableau, ce qui permet de réaliser le décodage.

1.6.2.8. Organisation du flux de sortie

Le codage entropique du standard JPEG2000 comporte deux étapes (voir Fig. 1.14). La première étape, abordée précédemment, permet de générer les flux binaires, tandis que la deuxième étape sert à organiser ces flux dans différents paquets. Les paquets ainsi obtenus sont ensuites insérés dans un flux de sortie unique que l'on appelle "codestream". Le codestream est constitué par de nombreux en-têtes et marqueurs. Les informations de codage (pas de quantification, nombre de tuiles, type d'ondelettes utilisées, etc), ainsi disponibles dans les entêtes, permettent au décodeur d'effectuer la reconstruction. L'organisation du flux de sortie est donnée dans la Fig. 1.23.

1.6.2.9. Résistance aux erreurs

JPEG2000 est un standard flexible permettant une résistance aux erreurs dues à la transmission via un canal dont la qualité de service n'est pas garantie, comme



FIG. 1.23.: Flux de sortie.

le réseau sans fil GSM [Ema02][Fag00]. Le code-bloc, codé indépendamment, ne suffit certes pas à résoudre ce problème. Pour résister aux erreurs, JPEG2000 opte pour une technique basée sur deux modes : mode SEGMARK et mode ERTERM.

Le premier mode SEGMARK, (Fig. 1.24-a), consiste à insérer un symbole spécial de quatre bits "1010" à la fin de chaque passe de codage cleanup. Cette opération se fait pendant l'encodage par plans de bits, ce qui permet d'avoir un seul plan de bits corrompu si un bit contenant l'erreur est détecté. Ainsi, lors du décodage, le décodeur identifie celui-ci en l'ignorant. Le fait de décoder un plan de bits contenant une erreur provoque un artéfact lors de la reconstruction des sous-bandes (transformée inverse en ondelettes).

Le deuxième mode, ERTERM (Fig. 1.24-b), consiste à prédire par un mot de code la fin de chaque passe de code; il y en a trois. Le décodeur peut détecter s'il y a eu un bit d'erreur à la fin de chaque passe de code. Le mot de code est réalisé selon un algorithme bien spécifique inclus dans le codeur arithmétique [Tau00].

1.6.2.10. Gestion de régions d'intérêt

JPEG2000 offre la possibilité d'une gestion de la Région d'Intérêt. L'option ROI est implantée entre la phase de quantification et l'encodage entropique (Fig. 1.14). L'utilisateur peut choisir une région dans l'image qu'il juge pertinente en



FIG. 1.24.: Technique pour la résistance aux erreurs du JPEG2000.



FIG. 1.25.: Disposition de la région d'intérêt; A gauche dans le domaine spatial; à droite dans les ondelettes.

codant cette région avec plus de précision. L'information du codage, en prenant en compte l'option ROI, est stockée dans le marqueur segment du codestream, ce qui permet au décodeur de réaliser le décodage souhaité comme : décoder en premier la région d'intérêt avant le fond. La Fig. 1.25 montre la disposition de régions d'intérêt dans différentes sous-bandes, tandis que la Fig. 1.26 représente l'exemple d'une image compressée avec la gestion de régions d'intérêt. Deux techniques sont proposées dans le standard : Maxshift et décalage global.

Technique du Maxshift

La méthode Maxshift consiste tout simplement à dégrader les coefficients appartenant au fond. Pour cela, on effectue un décalage binaire vers la droite des coefficients associés au fond, ce qui permet de placer les plans de bits des coeffi-



(a) Lena, image originale



(b) Codage sans la gestion ROI

(c) Codage avec la gestion ROI (masque rectangle)

FIG. 1.26.: Image JPEG2000 avec un taux de compression élevé t=250.

cients d'ondelettes appartenant à la ROI sur les plans de bits de poids plus fort (Fig. 1.27).

$$bg = c/2^s \quad \text{si} \quad c \in Fond \tag{1.52}$$

où c et bg sont des coefficients d'ondelettes.

Les étapes du codage sont les suivantes :

- 1. Générer le masque binaire permettant d'identifier les coefficients de la région d'intérêt ;
- 2. Calculer la valeur s du décalage binaire à appliquer;
- 3. Appliquer le décalage binaire aux coefficients du fond de l'image;



FIG. 1.27.: Illustration de la technique du Maxshift et du décalage global.

4. Spécifier dans le marqueur segment du codestream la valeur de s utilisée.

La difficulté de cette technique est la détermination de la valeur du bit de décalage s. En effet on risque d'avoir un dépassement de capacité de mémoire si on choisit une valeur de s assez élevée. Malgré tout cette valeur doit être choisie de telle sorte qu'après le décalage le coefficient le plus petit contenu dans une région d'intérêt soit plus grand que le plus grand des coefficients appartenant au fond.

L'intérêt de la méthode Maxshift est de ne pas avoir à transmettre au décodeur d'informations spatiales telles que les coordonnées spatiales du masque. On peut donc décoder l'image sans avoir besoin du masque utilisé lors de l'encodage.

Technique du décalage global

C'est une généralisation de la technique Maxshift. On effectue le même décalage binaire à droite aux coefficients d'ondelettes du fond mais, cette fois, la valeur du bit de décalage ainsi que le masque sont tous les deux envoyés au décodeur, ce qui rend plus complexe le décodage. Le décalage peut être réalisé différemment pour chaque sous-bande. Et une tuile peut avoir des régions d'intérêt multiples (décalage binaire avec des valeurs différentes), contrairement à la méthode Maxshift (une seule valeur de décalage). Vu la complexité du décodage, la norme accepte seulement 2 formes de régions : rectangle ou ellipse. Cette méthode est décrite dans la deuxième partie de la norme JPEG2000 [Dra00].

1.7. Codage vidéo : standards du multimédia

De nombreux organismes internationaux s'occupent de la normalisation des systèmes de codage vidéo : l'Union Internationale de Télécommunications-Télécommunications (UIT-T), l'Organisation Internationale de Normalisation (ISO) et la Commission Electrotechnique Internationale (IEC). L'UIT-T produit principalement des recommandations techniques. Les normes de codage vidéo de la famille de noms H.x (par exemple H.264) proviennent de l'UIT-T.

Evidemment, le codage vidéo exploite toutes les redondances qu'on a évoquées au §. 1.2, en utilisant deux modes : codage *"intra-image"* et codage *"inter-image"*. Etant codée indépendamment, l'image intra est compressée soit par la transformée en ondelettes (image JPEG2000), soit par la transformée en cosinus discrète (image JPEG). Elle sert ensuite comme image de référence. L'image inter, quant à elle, est réalisée par une prédiction. La prédiction se fait avec l'estimation et la compensation de mouvement.

1.7.1. Norme MPEG

Le groupe d'experts MPEG a pour objet de développer des normes permettant la compression, la décompression, le traitement et le codage des images animées et des données audio. Le principe de codage MPEG est basé sur la compensation de mouvement. Dans un premier temps, on convertit les composantes couleurs présentes dans l'image en luminance et chrominances (que l'on note YUV). Ainsi, les composantes UV contiennent deux fois moins d'information que la luminance Y. Dans un second temps, chaque composante est découpée en *macroblocs* (MB) de taille 16×16 pixels. Chaque macrobloc est ensuite découpé en blocs de 8×8 pixels. Le nombre de chrominances UV associé à chaque macrobloc est défini par le sous-échantillonnage choisi qui peut être de trois types : 4 : 2 : 2 (Fig. 1.28), 4 : 2 : 0 ou 4 : 4 : 4. Le format 4 : 2 : 2 signifie ceci : pour chaque pixel de Y, est échantillonné un pixel sur deux pour U et un pixel sur deux pour V; c'est le format le plus répandu dans la norme MPEG. Le choix du format est effectué en fonction de l'application visée.

La norme MPEG utilise deux modes de compression, spatial et temporel. Ces modes ont conduit à la définition de trois images appelées : image *intra* I (correspondant à la compression spatiale), image *prédite* P et image *bidirectionnelle* B (correspondant à la compression temporelle).

1.7.1.1. Principe de la norme MPEG

1.7.1.1.1. Image intra I

Appelée aussi image-clé, l'image intra est une image JPEG. C'est une image de référence à partir de laquelle les images P et B sont prédites. Dès lors l'image I est intercalée afin que le décodeur puisse décoder l'image en choisissant l'image



FIG. 1.28.: Macrobloc, échantillonnage YUV (4:2:2).

I la plus proche dans le flux reçu. Grâce à cette fonctionnalité, le décodeur peut faire une lecture rapide (avant et retour). Comme c'est une image JPEG, la transmission d'une image I au récepteur demande beaucoup de bande passante. Elle est donc très coûteuse en termes de débit de transmission.

1.7.1.1.2. Image prédite P

L'image P est une image prédite par rapport à une image I ou à une image P codée précédemment. C'est une image codée avec la compensation de mouvement. En effet, seule la différence entre le bloc en cours et le bloc d'image précédente est codée. On cherche dans l'image précédente un macrobloc identique ou semblable pour optimiser la compression. Ainsi la différence obtenue est codée spatialement, comme une image I. L'opération effectuée par la technique d'estimation de mouvement fournit les informations nécessaires au décodage. Ces informations qui sont le vecteur de mouvement et l'erreur de prédiction sont transmises au décodeur afin que celui-ci puisse reconstruire le macrobloc final en y ajoutant ces informations.

1.7.1.1.3. Image bidirectionnelle B

C'est une image prédite, interpolée par rapport à deux images : l'une correspondant à une image précédente (image I ou P) et l'autre correspondant à une image suivante (image P). Les images I et P peuvent servir de référence, les images B ne servent jamais de référence (pour éviter la répercussion des erreurs de prédiction). L'image B permet le meilleur taux de compression, mais en contre-partie elle comporte une certaine complexité ainsi qu'un coût de calcul assez important.

Le codage vidéo selon la norme MPEG constitue une succession d'images I, P et B (Fig. 1.29). L'insertion des images B dans le flux nécessite de changer l'ordre des images dans le flux vidéo. En effet, le décodeur ne peut pas décoder une image B sans avoir obtenu l'image de référence correspondant si cette dernière se situe



FIG. 1.29.: Séquence d'images I, P, B de la MPEG.

plus loin dans la séquence. Dans ce cas un retard est forcément introduit afin que le décodeur puisse décoder l'image B.

1.7.1.2. MPEG-1

La norme MPEG-1 (ISO/IEC-11172) créée en 1992, permet l'encodage de vidéo avec un débit de l'ordre de 1.5 Mbits/s. La norme est destinée à l'archivage de données. Elle comporte plusieurs parties dont la partie vidéo et la partie audio. Cette dernière a donné naissance au format de compression audio MP3.

1.7.1.3. MPEG-2

La MPEG-2 (ISO/IEC-13818), introduite en 1994, permet un débit de 20 Mbits/s pour la vidéo petit format, un débit de 3 à 6 Mbits/s pour la vidéo de qualité télévision (720 × 576). Ses résultats sont de meilleure qualité qu'avec son prédécesseur MPEG-1. La norme est utilisée pour le DVD, VCD et SVCD. Elle est également utilisée dans la diffusion de la télévision numérique notamment la Télévision Numérique Terrestre (TNT) et la Télévision Haute Définition (TVHD). La norme MPEG-2 permet d'avoir quatre résolutions telles que haute résolution (1920 × 1152), haute résolution 1140 (1440 × 1152), résolution normale (720 × 576) ainsi que basse résolution (352 × 288) ce qui n'est pas le cas de la norme MPEG-1 car elle ne définit qu'une seule résolution CIF (352 × 288), comme pour l'image d'un magnétoscope.

1.7.1.4. MPEG-4

La MPEG-4 (ISO/IEC-14496), introduite en 1998, est une norme de codage basée sur une approche orientée objet. Une scène devient alors une composition



FIG. 1.30.: Hiérarchie des objets dans MPEG-4.

d'objets média hiérarchisés (Fig. 1.30). Dans chaque hiérarchie, on trouve la description de la scène comportant l'arrière plan, les objets en mouvement séparés avec le fond ainsi que les objets audio. On peut interagir avec la scène en manipulant les objets. Par exemple, on peut modifier la forme géométrique ou la couleur d'un objet, un objet peut être supprimé ou ajouté dans la scène. Un concept de *"objets médias"* (Audio-Video Object) a été défini afin de réaliser ces fonctionnalités. En ce qui concerne les composants vidéo, le VOP (Video Object Plan) est ainsi utilisé. Le code VOP se fait de la même manière que les normes MPEG-1 et MPEG-2. On retrouve les images I, P et B mais avec un autre nom, comme I-VOP, P-VOP et B-VOP.

La norme MPEG-4 a été définie pour recouvrir trois domaines : l'informatique, les télécommunications ainsi que la télévision. De ce fait, beaucoup de domaines d'application sont visés : par exemple, la télévision numérique, la production vidéo, le multimédia embarqué, le streaming vidéo. La norme MPEG-4 est optimisée pour trois gammes de débits : inférieur à 64 Kbits/s, de 64 à 384 Kbits/s ainsi que de 384 Kbits/s à 4 Mbits/s.

1.7.1.5. MPEG-7

La MPEG-7 (ISO/IEC-15938), introduite en 1997, est une norme de représentation du contenu des documents multimédias. La norme est une description standardisée des données permettant une interprétation informatique (métadonnées). Les outils de description du contenu visuel sont groupés en diverses catégories : couleur, texture, forme, contour, mouvement. La norme est très utile pour le moteur de recherche, l'indexation, la reconnaissance des images par le contenu. MPEG-4 et MPEG-7 sont complémentaires. En effet, pour une séquence donnée, la MPEG-4 code la séquence tandis que MPEG-7 y ajoute une couche d'informations.

1.7.1.6. MPEG-21

La MPEG-21 (ISO/IEC-18034), introduite en 2000, est une norme permettant de gérer la protection, l'identification et la propriété intellectuelle des fichiers multimédias. Comme la norme MPEG-7, la norme MPEG-21 ne définit pas le système de codage.

1.7.2. Norme UIT-T

1.7.2.1. H.261

C'est une norme élaborée par l'UIT-T. Approuvée en 1993, la norme H.261 vise les applications de visiophonie pour le réseau RNIS à des débits multiples de 64 Kbits/s [LF03]. La procédure de codage est semblable à la norme MPEG, notamment le découpage en macroblocs, la DCT, la quantification, le codage entropique ainsi que la compensation de mouvement. L'estimation de mouvement d'un bloc (limité à +/- 15 pixels) se fait en direction horizontale et verticale. La norme accepte deux formats de vidéo : QCIF (176 × 144) et CIF (352 × 288).

1.7.2.2. H.263

Après l'apparition de la norme H.261, une nouvelle norme, H.263, a été élaborée en 1995. La norme est destinée à des applications à très bas débit. C'est une norme qui reprend le principe de base de la recommandation H.261 mais en y ajoutant quelques précisions telles que la compensation de mouvement. Désormais, l'image prédite B est possible; en outre on peut réaliser l'estimation et la compensation de mouvement avec une précision d'un demi-pixel. Elle s'applique à une vidéo au format CIF.

1.7.2.3. AVC/H.264

La norme AVC (Advanced Video Coding)/H.264 ou MPEG-4 part-10 est une norme élaborée conjointement par le MPEG et l'UIT. La réalisation technique de la norme (appellée aussi H.264 tout court) a été effectuée au sein du Groupe JVT (Joint Video Team). La norme inclut plusieurs profils [LF03] [Sun05]. Parmi ces profils, celui qui concerne les applications mobiles et e-streaming est le Profil X, celui qui concerne les applications de radiodiffusion en définition standard est le Profil principal. Plusieurs améliorations ont été apportées par rapport à ses prédécesseurs (MPEG-4, H.263) telles que l'utilisation de la transformée entière au lieu de la DCT, la possibilité d'utiliser les deux images : SI et SP. Les descriptions techniques de la norme sont développées dans [SWS03].

1.8. Choix stratégique : image fixe ou vidéo

Aujourd'hui pour la compression vidéo, la norme MPEG-4 est la plus utilisée, que ce soit dans le domaine de la transmission de vidéo (Internet ou 3G), ou dans le domaine de l'archivage. Actuellement de nombreux codecs (codeur-décodeur) sont disponibles sur le marché, ce qui témoigne du succès de la norme. Nous nous sommes intéressés à la performance de la norme MPEG-4 dans une application à très bas débit.

Pour la compression d'image fixe, le standard JPEG2000 est le plus adapté à notre problématique car il combine un fort taux de compression et la gestion de régions d'intérêt. De plus des implantations matérielles commencent à voir le jour [BBMP04].

Nous allons maintenant décrire les tests et les observations qui nous ont permis de comparer les performances de la norme MPEG-4 et celles JPEG2000.

1.8.1. Objectif des tests

Ces tests doivent permettre de faire un choix objectif entre les deux stratégies citées précédemment. De ce fait, l'étude comparative de MPEG-4 et JPEG2000 se fera dans le cadre de l'application industrielle présentée dans l'introduction.

1.8.1.1. MPEG-4 à très bas débit

L'industriel, la société MAGYS, dispose d'un équipement d'encodage MPEG-4 dont le codec est implanté en hardware. Dans son fonctionnement, on peut régler deux paramètres : la bande passante de la transmission et le taux de compression ("*bitrate*"). Nous combinons des tests en ne faisant varier qu'un paramètre à la fois. D'abord, nous fixons la bande passante du réseau IP à 9600 bits/s afin de simuler le bas débit, ensuite nous faisons varier le paramètre de codage *bitrate* jusqu'à ce que la taille de la séquence passe sous la limite de 1.2 Ko. La vidéo



FIG. 1.31.: Mesure de taille du fichier encodé avec MPEG-4 en fonction du bitrate.

reçue au décodeur sera alors étudiée d'un point de vue subjectif afin de vérifier la qualité par rapport au contexte.

1.8.1.1.1. Dispositif expérimental

Notre dispositif est constitué de :

- Codec VideoJet 10 MPEG-4. Ce codec est constitué d'un codeur et d'un décodeur MPEG-4 avec un serveur web intégré,
- Caméra fixe externe.

1.8.1.1.2. Fonctionnement

Le système capture et encode la vidéo, puis l'envoie vers un PC distant (décodeur) à travers le réseau IP. A l'arrivée, le décodeur décode et affiche la vidéo reçue. Le système permet aussi d'encoder une séquence d'images. Dans le cadre de cette expérience, afin de satisfaire notre application finale, nous utilisons une séquence d'images typiquement routière et qui a le format (320×240). Nous faisons ensuite varier la fréquence d'acquisition à 25, 15 et 8 img/s. Afin de satisfaire notre contexte d'étude, nous fixons la durée de la séquence à encoder à une seconde.

1.8.1.1.3. Résultats et conclusions

La Fig. 1.31 montre la courbe taille-bitrate de notre mesure. Cette expérience montre que nous n'avons pas la taille du fichier 1.2 Ko. En effet, au-delà d'un bitrate de 3 Kbits/s (correspondant à une taille de fichier 9.3 Ko), la séquence



FIG. 1.32.: Mesure de taille du fichier encodé avec JPEG2000 en fonction du taux de compression.

encodée est devenue illisible.

1.8.1.2. JPEG2000 à fort taux de compression

1.8.1.2.1. Dispositif expérimental

Pour compresser l'image, qu'on a extraite avec la séquence utilisée précédemment pour tester MPEG-4, nous utilisons le codec Kakadu [TM02] [Kak02] pour l'encodage JPEG2000. C'est la version la plus rapide qui existe actuellementet et permet la gestion de régions d'intérêt. Nous réalisons les tests avec les deux modes suivants : avec et sans ROI.

1.8.1.2.2. Résultats et conclusions

La Fig. 1.32 montre la courbe taille-taux de compression de ces deux modes de codage. Pour spécifier les régions d'intérêt qui sont généralement constituées par des véhicules en mouvement, on construit le masque relatif à celles-ci à la main. On retrouve dans les deux techniques de codage avec et sans ROI la taille d'image inférieure à 1.2 Ko.

1.8.2. Conclusion

Les résultats des tests sur la norme MPEG-4 nous amènent à conclure que celle-ci ne nous permet pas d'atteindre l'objectif souhaité car, après l'encodage, on ne parvient pas à obtenir la taille 1.2 Ko. Alors que JPEG2000 avec la gestion
de la ROI, nous permet d'obtenir cette taille. Ainsi notre étude s'est focalisée sur l'encodage image par image, la norme d'image fixe "*Motion JPEG2000*", pour encoder notre vidéo capturée par une caméra fixe. Notre système de codage est ainsi basé sur la compression d'image fixe.

1.9. Vers la compression par régions d'intérêt

L'option de la gestion de la région d'intérêt de la norme JPEG2000 présente beaucoup d'intérêt pour notre problématique. On détourne en effet cette fonctionnalité de son objectif initial, qui au départ sert à réaliser une visualisation progressive (voir annexe D) et à compresser davantage (voire éliminer) la partie arrière-plan d'image. Cela permet de minimiser l'information à envoyer. Ainsi, on extrait les objets pertinents dans la scène (pour nous, les régions mobiles), on encode puis on envoie au décodeur [BPBSS97][Bea03][FG03][MPL+05]. Pour le décodage, selon la technique Maxshift (cf. Eq. 1.52), le décodeur n'a pas besoin d'information spatiale relative au masque de régions d'intérêt pour décoder. Ainsi, on peut implicitement reconstruire la masque de mouvement pendant le décodage.

Dans cette approche la réception d'une image de référence ou arrière-plan avant les objets mobiles est donc nécessaire afin de reconstruire l'image finale au décodeur, ce qui demande donc une phase d'initialisation dans laquelle une première image de référence préalablement construite est envoyée au décodeur. L'approche proposée nécessite de traiter deux régions :

- 1. La première concerne les objets mobiles;
- 2. La seconde concerne les parties non-mobiles, c'est-à-dire le fond.

Tout d'abord, nous abordons les différentes techniques proposées dans la littérature pour déterminer les deux régions mentionnées ci-dessus. Ensuite nous choisirons la technique la plus adaptée à notre cas expérimental. Enfin nous développerons un système complet de codage et de décodage utilisant une image de référence.

On peut représenter la séparation en deux régions d'intérêt à l'aide d'un étiquetage binaire [CBC01]. Cela consiste à étiqueter tous les pixels dans l'image à l'instant t. L'étiquette e du pixel p définit l'appartenance du pixel au fond ou à une zone mobile, ce qui permet d'obtenir une carte binaire des changements temporels :

$$e(p,t) = \begin{cases} 1 & \text{si le pixel} \in \text{objet mobile,} \\ 0 & \text{si le pixel} \in \text{fond fixe.} \end{cases}$$
(1.53)

1.9.1. Extraction de régions de mouvement

La détection de mouvement dans le contexte d'une caméra fixe est un domaine de recherche très actif. Elle est utilisée dans beaucoup de domaines : la compression ou l'archivage, la transmission de vidéo, la robotique, dans le domaine médical, etc. En faisant l'hypothèse d'une illumination quasi-constante de la scène observée, on peut extraire en chaque pixel une observation o(p,t) portant sur la variation temporelle de l'intensité lumineuse I du pixel p.

Une approche statistique probabiliste à l'aide de la théorie des champs aléatoires de Markov MRF ("Markov Random Fields") [CB90] a été développée afin de détecter les mouvements. L'idée est d'introduire une modélisation a priori du champ des étiquettes connaissant les observations et le modèle. Cela permet ainsi de trouver la configuration la plus probable du champ des étiquettes. Cette approche donne de bons résultats [CBC01], mais malheureusement est souvent très lente. Cette lenteur est causée par les itérations nécessaires pour chercher la configuration la plus problable. Une implantation matérielle en temps-réel de MRF a été proposée [DLC99]. On retrouve le développement de cette technique dans [KZB93] [Lié98]. Compte tenu de nos contraintes matérielles, nous n'utiliserons pas la détection de mouvement par MRF.

1.9.1.1. Différence temporelle d'images

Pour cela, on utilise comme observation la valeur absolue (pour être invariant au contraste) de la différence temporelle d'intensité lumineuse entre deux instants successifs.

$$o(p,t) = |I(p,t) - I(p,t-1)|$$
(1.54)

D'une part cette observation, peu coûteuse en calcul, est très bruitée (bruits provenant de l'acquisition de la caméra et de la quantification), ce qui ne permet pas d'obtenir une segmentation correcte. D'autre part on ne détecte pas le mouvement lent. Néanmoins, elle est facile à réaliser. Dans le cas idéal, pour un mouvement non complètement uniforme, les pixels appartenant au fond sont donnés par une variation nulle (o(p, t) = 0), tandis que les pixels qui appartiennent à une région mobile correspondent à une variation non nulle.



FIG. 1.33.: Conséquences du mouvement d'un objet dans la scène à fond fixe.

Aussi, dans une scène à caméra fixe, on peut distinguer quatre zones occupées par l'objet en mouvement (Fig. 1.33) : le fond, la zone de glissement, la zone de recouvrement (région recouverte par l'objet à l'instant courant) et la zone d'écho (zone de fond découverte par l'objet à l'instant courant). Le traitement de ces zones se fait à l'aide d'un post-traitement de type filtre de morphologie mathématique (par exemple l'Eq. 1.59) sur chaque valeur d'étiquette (Eq. 1.53). Pour obtenir une segmentation correcte de l'objet mobile, il faut d'abord éliminer la zone d'écho, puis reconstruire la zone de glissement. Un état de l'art approfondi pour réaliser la détection de mouvement est développé dans [Amb00][PMCM01][Cha03].

Pour étiqueter le pixel, on choisit d'appliquer un seuillage adaptatif sur l'Eq. 1.54, comme celui de l'Eq. 1.57 afin de suivre la dynamique de l'observation.

1.9.1.1.1. Seuillage entropique

L'objectif du seuillage est d'éliminer les observations qui n'appartiennent ni à l'objet mobile ni au fond. L'obtention de la valeur θ du seuil est très délicate car il doit prendre en compte l'information incorporée dans l'image (valeur d'intensité lumineuse), donc être adaptatif en fonction de l'évolution de la scène. De nombreuses techniques sont proposées dans la littérature [Abu85][VMP98][WA03]. Parmi celles-ci se trouve le seuillage basé sur l'entropie, qui est développé dans [Lié98][LLF04]. L'idée est d'introduire la puissance entropique dans le calcul du seuil. La puissance entropique est donnée par :

$$N = \frac{1}{2\pi} \exp(2H_{nat}) \tag{1.55}$$

où H_{nat} est l'entropie de la source, exprimée en nats : $H_{nat} = \log(\sigma \sqrt{2\pi e})$, où σ est l'écart-type entropique de la source (distribution gaussienne équivalente). On exprime la valeur du seuil par une multiplication de σ avec un coefficient



FIG. 1.34.: Histogramme de la valeur d'intensité lumineuse X_i

multiplicatif définissant le pour centage de distribution seuillée que l'on note α :

$$\theta = \alpha.\sigma \tag{1.56}$$

Pour une application à la détection de mouvement, on peut trouver une valeur de α permettant une extraction efficace des pixels mobiles dans la scène (voir histogramme, Fig. 1.34). Pour cela on prend $\alpha = 4$:

$$\theta = 4\sigma \approx 2^{H_{bit}} \tag{1.57}$$

Le seuil est calculé globalement pour chaque trame de la séquence car il dépend seulement de l'entropie de l'image (cf. Eq. 1.58), ce qui ne nécessite pas de mémoriser une séquence d'images. Ensuite, il a été démontré que pour une application routière, ce seuil donne des résultats satisfaisants [LLF04].

$$H_{bit} = -\sum_{i=0}^{255} \Pr(x_i) \times \log_2(\Pr(x_i))$$
(1.58)

1.9.1.1.2. Amélioration de l'image des étiquettes

Après le seuillage, nous obtenons une image binaire correspondant à la séparation en deux régions. Il faut alors traiter cette image afin d'en améliorer la définition : élimination des zones de glissement, des zones d'écho, des zones d'ombre, etc.

Filtres de morphologie mathématique

Une technique pour améliorer le résultat du seuillage est d'utiliser les filtres de morphologie mathématique [Mar96]. Ils permettent de regrouper les régions connexes présentes dans le masque de mouvement. On distingue deux cas : la



c) Masque corrigé, obtenu avec l'algorithme

FIG. 1.35.: Filtre de morphologie mathématique [CHH+04].

morphologie binaire s'appliquant sur une image binaire et la morphologie s'appliquant sur une image en niveaux de gris. On se positionne principalement dans le premier cas et les opérations de bases sont la dilatation et l'érosion. Il s'agit d'une technique basée sur la théorie des ensembles.

L'application d'une ouverture ou d'une fermeture améliore le masque du mouvement en supprimant les zones de glissement et d'écho. Le filtre de morphologie mathématique, calculé à partir de la différence entre une dilatation et une érosion, et développé dans [CHH⁺04], permet d'atténuer l'ombre. Il consiste à appliquer l'Eq. 1.59 sur deux images successives à niveaux de gris (voir Fig. 1.35-a). Selon les auteurs, comme le montre la Fig. 1.35-c, le filtre permet d'effacer l'ombre du mouvement.

$$GRA(I) = (I \oplus E) - (I \ominus E)$$
(1.59)

où \oplus représente l'opérateur dilatation, \ominus représente l'opérateur érosion et E est un élément structurant.

Approches au niveau couleur

Pour supprimer l'ombre d'un objet en mouvement, des techniques qui exploitent la couleur sont plus adaptées que les approches ci-dessus. En effet, pour une image couleur, le changement d'un point éclairé à un point ombré se traduit par une chute nette de la luminance alors que les autres composantes chrominances sont conservées. A partir de ce constat, en utilisant l'espace couleur :



FIG. 1.36.: Diagramme de la technique de soustraction de fond.

luminance, chrominance rouge et chrominance bleue il sera possible de détecter si un pixel appartient à l'objet mobile ou à son ombre. [IBBD03] utilise trois paramètres de seuillage relatifs aux trois composantes de l'espace couleur. Le seuillage est réalisé relativement aux rapports des composantes de l'espace couleur de l'image de référence (le fond) et celles de l'image courante. Cette technique est performante et peu coûteuse en temps de calcul mais les seuils sont déterminés empiriquement.

Un état de l'art complet dans le cadre de la détection et de la suppression de l'ombre d'objet en mouvement est évalué dans [PMCM01], où les auteurs ont sélectionné une vingtaine de méthodes.

1.9.1.2. Différence temporelle du fond

Dans cette technique, on utilise comme une observation la valeur absolue de la différence temporelle d'intensité lumineuse entre l'image courante I et l'image de référence I_{ref} :

$$o(p,t) = |I(p,t) - I_{ref}(p,t)|$$
(1.60)

On applique ensuite le seuillage et le post-traitement, respectivement présentés aux §1.9.1.1.1 et §1.9.1.1.2 pour réaliser l'étiquettage.

Cette technique (Fig. 1.36) est largement présente dans la littérature et son niveau de performance dépend fortement de la qualité du fond [KCHD05] [SG99] [RKJB00] [Yos04] [AM05]. La difficulté principale est l'obtention du fond mais surtout de son actualisation temporelle.

1.9.1.2.1. Estimation d'une image de référence

Obtenir une image de référence est une tâche très complexe [VMP98]. On distingue deux cas d'applications : scène intérieure ou extérieure. Dans le premier cas d'application, d'une manière générale, les changements intervenant sur la scène sont l'illumination (éclairage), le bruit d'acquisition de la caméra puis les zones affectées par le mouvement de l'objet. Tandis que pour les scènes extérieures, en plus des changements mentionnés précédemment, de nombreux phénomènes peuvent provoquer une perturbation de l'image de référence. On peut les répartir en deux catégories : les changements provoqués par les phénomènes naturels et le changement causé par l'objet lui même.

Pour la première catégorie, comme le montre la Fig. 1.37-b, la valeur de l'intensité lumineuse des pixels p_1 et p_2 , qui sont extraits de la séquence d'images Fig. 1.37-a, varie de façon significative en seulement 5 secondes d'acquisition (fréquence d'acquisition 8 img/s). Cela se vérifie notamment dans le cas d'événements naturels comme le vent, le coucher du soleil, le mouvement de l'arbre au bord de la route, un nuage, des phénomènes météorologiques, etc. La deuxième catégorie concerne le changement provoqué par l'objet mobile lorsque celui-ci se déplace lentement, s'arrête et repart ; il s'agit donc de la caractéristique de l'objet mobile (taille, vitesse, texture, etc).

Les techniques permettant d'extraire une image de référence peuvent être regroupées en deux familles : techniques non-récursives ou récursives.

1.9.1.2.1.1. Techniques non-récursives

Utilisée pour des applications disposant de ressources suffisantes, la technique non-récursive utilise de nombreuses images bufferisées pour modéliser une image de référence. Il faut stocker la séquence sur une longueur importante pour notre contexte applicatif, au moins 10 secondes, et cela demande beaucoup de mémoire.

Pour estimer la référence, on cherche la variation temporelle de chaque pixel dans la séquence bufferisée. Par conséquent, la fréquence d'acquisition et la longueur de la séquence sont des facteurs très importants quant à la qualité du résultat de cette technique. Un objet qui se déplace lentement dans la scène nécessite une longueur de séquence d'autant plus importante que sa vitesse est faible afin d'obtenir une référence correcte. D'où la difficulté de la mise en œuvre de la méthode si la longueur a été fixée initialement. Nous présentons ci-après les techniques les plus développées dans la littérature.

Méthode par valeur maximale-minimale

Haritaoglu *et al.* [HHD00] utilisent 3 modèles pour modéliser le fond. La valeur d'intensité lumineuse de chaque pixel est modélisée temporellement par la valeur maximale, la valeur minimale et la valeur maximale de la différence entre deux images successives. Pour construire l'image de référence, leur système nécesPixel p₂



Pixel p₁





(b)

FIG. 1.37.: Variation d'une valeur d'intensité lumineuse.

site une mémorisation de 10 à 20 secondes afin de pouvoir estimer une première image de référence par un filtre médian. Ainsi, un pixel p appartient au fond si la valeur d'intensité lumineuse vérifie :

$$|I(p,t) - max| > D \qquad \text{ou} \qquad |I(p,t) - min| > D \tag{1.61}$$

où max et min représentent respectivement la valeur maximale et minimale d'intensité lumineuse dans l'image de référence; D est la valeur absolue maximale de la différence entre deux images successives.

Méthode non-paramétrique

Développée par Elgammal *et al.* [EHD99], la méthode modélise l'historique des valeurs d'intensité lumineuse sur un nombre d'images L par une fonction de densité de probabilité. Soit x_{t-L} , x_{t-L+1} , \cdots x_{t-1} , l'historique des valeurs d'intensité d'un pixel, la probabilité qu'un certain pixel ait une intensité x_t à l'instant t est :

$$Pr(x_t) = \frac{1}{L} \sum_{i=t-L}^{t-1} K(x_t - x_i)$$
(1.62)

où K est une distribution normale de variance nulle $N(0, \sum)$ et \sum est la matrice de covariance de la composante couleur :

$$K(x_t - x_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (x_t - x_i)^T \sum^{-1} (x_t - x_i)\right)$$
(1.63)

On suppose que les différentes composantes sont indépendantes. Ainsi, pour une image ayant d = 3 composantes couleurs (R,G,B) de variance σ_j^2 $(j = 1 \cdots 3)$, on peut définir la matrice de covariance ainsi que la probabilité de l'Eq. 1.62 comme suit :

$$\sum = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}$$
(1.64)

$$Pr(x_t) = \frac{1}{L} \sum_{i=1}^{L} \prod_{j=1}^{3} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2} \frac{\left(x_{t_j} - x_{i_j}\right)^2}{\sigma_j^2}\right)$$
(1.65)

La variance σ_j^2 , calculée indépendamment pour chaque composante couleur j, est obtenue en fonction du médian de deux trames consécutives $|x_i - x_{i+1}|$.

1.9.1.2.2. Techniques récursives

Contrairement aux techniques précédentes, les techniques récursives ne nécessitent pas de mémorisation d'images. L'estimation d'une image de référence est réalisée soit avec un filtre récursif soit avec une approche probabiliste prenant en compte les informations du passé. L'image de référence est actualisée à chaque instant après traitement de l'image courante. Dans la suite, nous présentons les méthodes les plus souvent mises en œuvre dans la littérature.

Filtre de Kalman

Le filtre de Kalman, conçu pour suivre la trajectoire d'un objet en mouvement, est très utilisé dans le domaine du suivi d'un objet mobile dans une scène. C'est un filtre récursif basé sur une représentation d'état. L'objectif est d'estimer pour chaque instant l'état caché d'un système dynamique. On cherche ainsi l'estimation optimale. Celle-ci est considérée comme atteinte lorsque certaines conditions sont vérifiées [WB01]. La technique récente pour estimer une image de référence I_{ref} avec sa fonction dérivée temporelle I'_{ref} par le filtre de Kalman est :

$$\begin{bmatrix} I_{ref}(p,t) \\ I'_{ref}(p,t) \end{bmatrix} = A \cdot \begin{bmatrix} I_{ref}(p,t-1) \\ I'_{ref}(p,t-1) \end{bmatrix} + K_t \cdot \begin{bmatrix} I(p,t) - H \cdot A \cdot \begin{bmatrix} I_{ref}(p,t-1) \\ I'_{ref}(p,t-1) \end{bmatrix} \end{bmatrix}$$
(1.66)

où A est la dynamique associée au fond, H la matrice de représentation de la mesure et K_t est un scalaire qui représente le gain du filtre de Kalman. En prenant les valeurs numériques :

$$K_t = \alpha. \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & \beta \\ 0 & \beta \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

on peut réécrire ainsi l'Eq. 1.66 :

$$I_{ref}(p,t) = \alpha I(p,t) + (1-\alpha)I_{ref}(p,t-1) + \alpha(1-\beta)I'_{ref}(p,t-1)(1.67)$$

$$I'_{ref}(p,t) = \alpha (I(p,t) - I_{ref}(p,t-1)) + \beta(1-\alpha)I'_{ref}(p,t-1)$$
(1.68)

Le modèle (Eq. 1.67) montre que nous avons deux paramètres (α, β) à ajuster où $0 \le \alpha, \beta \le 1$ pour régler l'apprentissage d'une image de référence. Dans la pratique, la valeur de α est très inférieure à celle de β et les valeurs testées dans [CK05] sont $\alpha = 0.001$ et $\beta = 0.7$.

Mixture de gaussiennes

Cette méthode est basée sur une approche probabiliste. La technique consiste en l'estimation de la valeur d'intensité pixel par de multiples distributions gaussiennes. On cherche, parmi ces modèles de distribution, la plus probable par rapport à l'hypothèse d'appartenance au fond. Friedman et Russel [FR97] ont proposé, dans le cas d'une application routière, une méthode pour modéliser par un triplet de distribution gaussienne : la route, l'ombre et le véhicule. Chaque composante (route, ombre, véhicule) est estimée à l'aide d'une fonction de densité : distribution normale. Le modèle est paramétrique et utilise la moyenne, la variance et un poids. Ces grandeurs sont estimées à l'aide de l'algorithme EM [KGV83] [GG84] [Bes86].

[FR97] utilise trois distributions; ensuite Stauffer et Grimson [SG99] ont généralisé ces techniques par de multiples distributions K où $3 \leq K \leq 5$. On peut obtenir selon les auteurs un fonctionnement en temps-réel pour les images à basse résolution QCIF ou CIF. Comme pour toutes les méthodes d'extraction d'un fond fixe, cette méthode est confrontée au problème de traitement d'un très grand objet se déplaçant très lentement dans la scène. En effet, la mise à jour des paramètres de chaque composante gaussienne ne parvient pas à effacer instantanément ce type d'objet. Plusieurs images sont donc nécessaires pour les supprimer de l'image de référence.

Pour contrer cela, l'hypothèse suivante doit être vérifiée pour chaque pixel : "le temps d'observation du fond doit être très supérieur au temps d'observation d'un objet mobile". Nous présentons ci-dessous les grandes lignes de cette technique.

Soit la probabilité $Pr(x_t)$ qu'un certain pixel ait une intensité x_t à l'instant t, celle-ci est estimée par une somme de distributions gaussiennes :

$$\Pr(x_t) = \sum_{j=1}^{K} \omega_{j,t} \times \eta(x_t, \mu_{j,t}, \sum_{j,t})$$
(1.69)

où $\eta(x_t, \mu, \Sigma)$ est la densité de probabilité gaussienne, définie par :

$$\eta(x_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{d}{2}} |\sum|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x_t - \mu_t)^T \sum^{-1} (x_t - \mu_t)\right)$$
(1.70)

où d = 3 (3 composantes couleurs).

Pour chaque nouvelle trame, les paramètres (μ, σ^2, ω) des distributions sont réactualisés à l'aide de fonctions récursives :

TAB. 1.4.: Valeur de paramètres de mixture de gaussiennes.

Paramètres	K	α_m	σ_m	θ_b	σ_0	ω_0
Valeur	3	0.01	2.5	0.60	12	0.02

$$\omega_{j,t} = (1 - \alpha_m)\omega_{j,t-1} + \alpha_m \tag{1.71}$$

$$\mu_{j,t} = (1 - \rho_m)\mu_{j,t-1} + \rho_m x_t \tag{1.72}$$

$$\sigma_{j,t}^2 = (1 - \rho_m)\sigma_{j,t-1}^2 + \rho_m (x_t - \mu_{j,t})^2$$
(1.73)

où α_m , ρ_m , coefficients d'adaptation, permettent de régler la vitesse d'apprentissage, avec $0 \leq \alpha_m \leq 1$. [SG99] ont proposé que la valeur ρ_m soit obtenue par :

$$\rho_m = \alpha_m \eta(x_t \mid \mu_j, \sigma_j) \tag{1.74}$$

Tandis que [PS02] l'a critiqué et définit la valeur ρ_m de la manière suivante :

$$\rho_m \approx \frac{\alpha_m}{\omega_{j,t}} \tag{1.75}$$

Désormais, il nous faut définir la combinaison de distributions correspondant au fond. Pour cela, un seuil θ_b relatif à une somme de poids est introduit et permet de définir les distributions appartenant au fond. Après un tri des distributions selon la variance, les *B* premières distributions vérifiant l'Eq. 1.76 sont considérées comme modèle du fond.

$$B = argmin_b(\sum_{j=1}^b \omega_j > \theta_b) \tag{1.76}$$

Le choix des différents paramètres à l'initialisation est très important car il impacte le temps d'obtention d'un fond de qualité. Le Tab. 1.4 représente la valeur des paramètres de la mixture à l'initialisation. Ces valeurs sont choisies empiriquement après une série de tests avec des séquences typiques. Cette technique est la plus robuste dans le cadre d'une application routière [CK05].

1.9.1.2.3. Techniques retenues pour l'expérimentation

Dans le cadre de notre étude pour la segmentation en régions, nous combinerons pour la détection du mouvement la technique utilisant la différence entre deux images successives et celle utilisant la différence entre l'image courante et l'image de référence [CBC01]. Le résultat obtenu sera binarisé par seuillage entropique, afin de suivre la dynamique de l'image, et amélioré en combinant les opérateurs de morphologie mathématique et le traitement dans l'espace couleur Y, Cr, Cb.

L'actualisation de l'image de référence sera obtenue à l'aide de la technique par mixture de gaussiennes, version Stauffer et Grimson [SG99]. Nous disposerons alors à chaque instant d'une image caractérisant le fond au niveau de l'encodeur.

1.10. Conclusion

Nous avons abordé dans ce chapitre un état de l'art sur les différentes techniques de compression de vidéo et les descriptions de différentes normes de codage existantes, élaborées au sein de l'UIT-T, de MPEG et de ISO/IEC.

Les tests réalisés sur le standard MPEG-4 dans notre contexte d'application, nous ont conduits à élaborer un nouveau système de codage par la gestion de régions d'intérêt du standard JPEG2000, ce que nous développerons dans le prochain chapitre. Au lieu du codage vidéo, on utilisera l'encodage d'images fixes.

Le choix de la norme JPEG2000 est validé par le fait que, d'une part elle permet de compresser à un taux très élevé (comparativement à ses prédécesseurs JPEG ou JPEG-LS), et d'autre part JPEG2000 permet la gestion de régions d'intérêt dans l'image.

Les régions d'intérêt, caractérisées par un ensemble d'objets mobiles, sont obtenues à l'aide de la détection automatique de mouvement qui utilise également une image de référence. Cette dernière est estimée à l'aide de la méthode basée sur l'utilisation de mixtures gaussiennes.

2. Codage par le contenu pour la compression d'images mobiles

Sommaire

2.1.	Description générale du système	65
2.2.	Définition du système local-distant	65
2.3.	Schéma-bloc de l'encodeur	66
2.4.	Conclusion	73

Dans ce chapitre, nous présentons l'aspect théorique de notre système. L'encodage d'image fixe est abordé en précisant les choix d'organigramme et d'algorithme afin de produire des images d'une taille mémoire inférieure à 1.2 Ko.

2.1. Description générale du système

Le système de codage proposé consiste à encoder puis envoyer au décodeur une image contenant seulement les objets mobiles. Afin de reconstruire l'image finale à la réception, le décodeur doit disposer d'une image de référence sur laquelle les objets mobiles seront apposés. Cela nécessite ainsi une phase d'initialisation de l'encodeur et du décodeur afin d'obtenir une première image de référence qui est ensuite envoyée au décodeur.

2.2. Définition du système local-distant

Nous adoptons les définitions suivantes :

 L'encodeur est le système local. D'une part, il est composé de dispositifs matériels (CPU, acquisition vidéo, caméra, module de transmission) et d'autre part il contient l'algorithme de l'encodage; Le décodeur est le système distant. Il dispose tout simplement d'un équipement d'affichage d'images reconstruites : un PC, par exemple, doté d'un module de réception.

2.3. Schéma-bloc de l'encodeur

La Fig. 2.1-a représente le schéma-bloc de notre système. Il comprend 5 blocs de traitements qui sont implantés dans le système local :

- 1. Phase d'initialisation : Construction de la première image de référence, Envoi au décodeur;
- 2. Segmentation en régions : Construction de la ROI par extraction des objets mobiles ;
- 3. Encodage de données par JPEG2000;
- 4. Transmission de la ROI;
- 5. Mise à jour d'une image de référence en local.

Le fonctionnement de chaque bloc sera détaillé ci-après.

2.3.1. Phase d'initialisation

Il s'agit d'extraire une image de référence à partir d'une courte séquence d'images afin de permettre le démarrage du système. Il est inévitable d'avoir sur cette première estimation du fond des traces d'objets mobiles incomplètement effacées. Certaines hypothèses : comme la connaissance a priori de tous les pixels appartenant au fond **Hypothèse** A, la stabilité de la valeur d'intensité lumineuse des pixels appartenant au fond **Hypothèse** B doivent être admises si l'on veut garantir une qualité minimale [GTCS⁺01] [WS06]. Ainsi on suppose que, pendant la construction d'une référence, l'objet mobile ne peut stationner que durant une période très courte afin de ne pas être intégré dans la référence.

La méthode basée sur le filtrage de Kalman [WB01] est difficile à maîtriser pour une courte séquence, ce qui est dû au fait qu'on doit ajuster le gain du filtre α et la dynamique du fond β avec suffisamment de temps.

La méthode non-paramétrique [EHD99], la technique moyenneur ou la technique médian ne sont pas utilisables, vu la nécessité de la mémorisation de la séquence en vue de l'implantation expérimentale sur PC104.

La mixture de distributions gaussiennes [SG99] présente un défaut majeur du fait qu'on ne maîtrise pas le temps exact, pour l'effacement correct de l'objet



(a) Encodeur



(b) Décodeur

FIG. 2.1.: Schémas-blocs du codec proposé.

mobile dans la scène. Ceci s'obtient en réglant le temps d'apprentissage de la mise à jour du paramètre du modèle.

2.3.1.1. Filtrage récursif du premier ordre

Le compromis entre la qualité d'image construite et le temps nécessaire à la construction, ainsi que la capacité de mémoire disponible, nous conduit à choisir un filtre récursif du premier ordre dont la fonction de transfert et l'équation temporelle sont les suivantes :

$$G(z) = \frac{b}{1 + az^{-1}} \tag{2.1}$$

où a et b sont deux réels. En choisissant $(b = \alpha, a = \alpha - 1)$, on obtient la réponse du filtre par :

$$I_{refInit}(p, t+1) = \alpha I(p, t+1) + (1-\alpha)I_{refInit}(p, t)$$
(2.2)

où $I_{refInit}(p,t)$ et I(p,t) sont les valeurs d'intensité du pixel p respectivement dans l'image de référence et l'image courante à l'instant $t. \alpha \in [0,1]$ règle la vitesse de l'apprentissage.

Ce modèle est simple à appréhender et ne nécessite pas de mémorisation d'images de la séquence. On peut obtenir facilement le nombre minimum L d'images nécessaires à la construction. L est lié à la constante de temps du filtre. Cette dernière est réglée par le coefficient d'adaptation α .

2.3.1.2. Vérification d'hypothèses

Pour améliorer le résultat précédent et prendre en compte l'*hypothèse* A, en particulier l'effacement de l'historique du mouvement de l'objet dans la référence, les valeurs du coefficient d'adaptation α ont été partagées en deux catégories :

$$\alpha = \begin{cases} 0 & \text{si } p \in \text{un objet mobile} \\]0,1] & \text{si } p \in \text{référence à construire} \end{cases}$$
(2.3)

Nous définissons alors une carte binaire de stabilité temporelle avec deux images successives I(t-1) et I(t), qui est obtenue à l'aide de l'information de contour dont on atténue le bruit d'acquisition par un filtre moyenneur (3×3) .

$$D(p,t) = |I_g(p,t-1) - I_g(p,t)|$$
(2.4)

où $I_g(p,t)$ est le gradient spatial d'un pixel p d'une image à l'instant t de la

séquence. La différence D(p, t) est utilisée pour vérifier le respect de l'**hypothèse B**. On utilise ensuite un seuil entropique θ afin de binariser l'image de différence D(p, t):

$$M(p,t) = \begin{cases} 1 & \text{si } (D(p,t) > \theta) \\ 0 & \text{dans les autres cas} \end{cases}$$
(2.5)

où M est un masque de mouvement. Nous obtenons alors l'Eq. 2.3 par :

$$\alpha = \begin{cases} 0 & \text{si } M(p,t) = 1\\]0,1] & \text{si } M(p,t) = 0 \end{cases}$$
(2.6)

Pour combler les trous isolés, on utilise un filtrage de morphologie mathématique fermeture, défini par l'Eq. 2.7 :

$$(I(p,t) \oplus E(3,3)) \ominus E(3,3)$$
 (2.7)

où E(3,3) est l'élément structurant, formé par une matrice de taille (3×3)

2.3.1.3. Limitations dues au modèle

Afin d'étudier les limites de notre modèle, nous avons construit deux vidéos synthétiques. La première vidéo contient un petit objet se déplaçant dans la scène et la seconde un grand objet. Notre modèle (Eq. 2.2) donne un bon résultat avec la première vidéo : objet mobile petit. Par contre pour la seconde, des traces de l'objet persistent dans l'image de référence. En effet lors de la construction, un grand objet mobile est difficilement supprimé notamment à cause de la taille de la zone de glissement.

Dans notre contexte, un objet est considéré de grande taille si le taux d'occupation Λ de celui-ci est supérieur à un seuil prédéfini noté Λ_{min} . Λ est déterminé par le rapport entre le nombre de pixels qui sont égaux à 1 dans le masque de l'objet et la taille d'image. Le seuil Λ_{min} est déterminé expérimentalement en prenant en compte l'objet le plus grand d'une scène routière : un semi-remorque, qui est représenté dans la Fig. 2.2, ici $\Lambda = 20\%$.

2.3.1.4. Paramètres de l'initialisation

Le modèle de filtrage contient deux paramètres (α, L) . Pour une entrée échelon, notre système étant du premier ordre, la valeur finale de la sortie atteint 95% de la valeur de l'entrée à l'instant $t = 3\tau$ où τ est la constante de temps du filtre. Si on fixe L, alors $L \times Te$ (Te période d'acquisition) correspond au temps de réponse



FIG. 2.2.: Grand objet mobile; $\Lambda = 20\%$.

TAB. 2.1.: Valeur de paramètres.

Paramètres	α	L	Λ_{min}
Valeur	0.01	50	15%

à 95% et on déduit la constante de temps τ . Alors α peut être déterminé. Dans notre application, nous prenons la valeur des paramètres représentée dans le Tab. 2.1.

A présent nous considérons qu'une première image de référence est disponible et peut être envoyée au décodeur. Le système peut alors commencer à fonctionner en régime permanent c'est-à-dire à transmettre des données.

2.3.2. Construction de la région d'intérêt

La région d'intérêt, ensemble d'objets mobiles, est obtenue par la détection de mouvement, cf. §1.9.1. Pour réaliser l'étiquetage d'un pixel appartenant à l'objet mobile ou au fond, on a opté pour deux observations :

– La première est la différence entre deux images successives I aux instants t-1 et t:

$$o_{dt}(p,t) = |I(p,t-1) - I(p,t)|$$
(2.8)

– La deuxième observation est la différence entre l'image de référence I_{ref} estimée pour chaque trame et l'image courante :

$$o_{dr}(p,t) = |I_{ref}(p,t) - I(p,t)|$$
(2.9)

Nous utilisons ensuite la technique moins coûteuse en calcul, du seuillage entropique spatial adaptatif θ , suivi d'un ET logique (Fig. 2.3). Ainsi, tous les pixels étiquetés mobiles sont des candidats au masque ROI que l'on note M_{ROI} :

$$M_{ROI}(p,t) = \begin{cases} 1 & \text{si } (o_{dt}(p,t) > \theta) \text{ ET } (o_{dr}(p,t) > \theta)) \\ 0 & \text{dans les autres cas} \end{cases}$$
(2.10)



FIG. 2.3.: Illustration du résultat par l'opérateur ET logique.

2.3.3. Image JPEG2000

Le bloc Encodage JPEG2000 (voir Fig. 2.1-a) consiste à utiliser le codec JPEG2000 standard avec la prise en compte de la ROI. Toutefois, nous apportons des modifications à l'image à encoder pour réduire la taille des données sans pour autant utiliser le taux de compression maximal.

2.3.3.1. Suppression du fond

Etant donné que le système distant dispose d'une image du fond, les informations relatives à cette zone peuvent ne pas être transmises. Dans le domaine spatial, attribuer une valeur constante à l'intensité lumineuse des composantes couleurs conduit à la suppression des coefficients d'ondelettes pour cette région. Cette modification d'intensité lumineuse appliquée à tous les pixels du fond permet de choisir un taux de compression inférieur (gage de qualité) tout en vérifiant la contrainte de 1.2 Ko. Par contre, il est nécessaire que la valeur d'intensité du fond soit égale à la dynamique d'image à encoder avec JPEG2000 afin d'éviter les dégradations au niveau des contours de la ROI. Dans notre cas, l'image à trois composantes (R,G,B) est codée en 24 bits donc la dynamique est ramenée à 128.

Cette opération permet d'une part de contrôler la transition entre les coefficients de la transformée en ondelettes appartenant au fond et la ROI, et diminue d'autre part les données à coder. La Fig. 2.4 représente la répartition des données.



FIG. 2.4.: Mise en valeur uniforme des pixels appartenant au fond.

2.3.3.2. Compression

Ensuite l'image contenant seulement les objets mobiles est compressée à un taux t=250 avec prise en compte de la ROI par la mise en œuvre de la technique Maxshift pour obtenir la taille inférieure à 1.2 Ko. Il s'agit donc ici d'exploiter au maximum la performance du standard en utilisant l'option de la gestion de la ROI.

2.3.4. Transmission d'images objets

Après l'encodage, l'image est maintenant prête à l'envoi. Deux modes de transmission sont mis en place afin que le décodeur puisse identifier l'image reçue : l'image de référence initiale ou l'image de mouvement (ROI).

A l'initialisation, l'image de référence est encodée sans ROI avec un taux moyen (typ. 1 : 60) puis transmise au décodeur une seule fois. Un indicateur du type FLAG est nécessaire afin que le décodeur puisse identifier l'image reçue : image du fond ou image d'objets mobiles. La Fig. 2.5 représente un en-tête de données à envoyer. Chaque image envoyée est précédée par un en-tête. Le contenu de ce dernier peut être écrit directement dans le codestream d'image JPEG2000 encodée.

Les champs de l'en-tête sont :

- SIZE : indique la taille de données à envoyer ;

SIZE FLAG DATA

FIG. 2.5.: En-tête de la transmission.

- FLAG : indique la catégorie d'images à envoyer et peut avoir deux états :
 0 : images objets (contenant la ROI du mouvement), 1 : images de référence (contenant la ROI des régions à mettre à jour);
- DATA : contient les données.

2.4. Conclusion

Nous avons introduit dans ce chapitre une technique de codage (codec) basée sur la prise en compte de la ROI liée au mouvement. Ce dernier est déterminé par l'utilisation combinée d'une image de référence et de la différence entre des images successives. L'image de référence est estimée pour chaque trame selon la technique mixture de gaussiennes.

Après la phase d'initialisation permettant la construction d'une première image de référence, l'image courante est ensuite codée à l'aide du standard JPEG2000 avec un fort taux de compression et avec gestion de la région d'intérêt. Dans le chapitre suivant, nous présenterons notre système de décodage et la mise à jour de l'image de référence du décodeur.

3. Décodage et Réactualisation d'images de références

Sommaire

3.1.	Schéma-bloc du décodeur	75
3.2.	Réactualisation d'images de références	77
3.3.	Conclusion	91

Dans ce chapitre, nous présentons l'aspect théorique de notre système de décodage et de réactualisation d'images de références. Dans la première partie, le décodeur est abordé en précisant les choix d'organigramme et d'algorithme afin de reconstruire l'image finale. Dans la seconde partie, la technique de mise à jour de l'image de référence par morceaux est présentée.

3.1. Schéma-bloc du décodeur

Le décodeur (Fig. 2.1-b) reçoit l'image via le canal de transmission et procède au décodage JPEG2000. Des modifications doivent être apportées à ce processus afin de construire implicitement le masque binaire utilisé lors de la segmentation en régions. Cette détermination implicite permet de se passer d'informations additionnelles concernant les données spatiales des zones en mouvement. En combinant le masque, l'image reçue et l'image de référence, l'image finale est reconstruite.

3.1.1. Identification d'image

Le décodeur reçoit les images objets, vérifie leur catégorie et les décode. Si l'image reçue est une image de référence, le décodeur la mémorise comme telle (phase d'initialisation). Dans le cas contraire, la nouvelle image reçue est utilisée pour contruire l'image finale présentée à l'utilisateur.

3.1.2. Reconstruction implicite du masque et décodage

D'après la propriété de la méthode Maxshift, pour connaître si un coefficient de la transformée en ondelettes c appartient à la zone d'intérêt, il suffit de le comparer avec la valeur de 2^s où s est le nombre de décalages de bits effectués lors de l'encodage. s est signalée dans le codestream de JPEG2000.

$$\begin{cases}
c \in ROI & \text{si } c \ge 2^s \\
c \in Fond & \text{dans les autres cas}
\end{cases}$$
(3.1)

Toutefois, un coefficient c de la transformée en ondelettes appartenant à la ROI peut être nul et donc invariant à la technique Maxshift. Donc, seuls les coefficients à valeur non nulle sont identifiables comme étant des éléments de ROI avec la prise en compte du décalage s. Un masque binaire imparfait est donc obtenu et nous utilisons un post-traitement pour améliorer le masque reconstruit. Il s'agit du filtrage de morphologie mathématique (fermeture) de l'Eq. 2.7.

La partie concernant la décompression de l'image est maintenue complètement standard et finalement nous disposons du masque reconstruit et des données relatives à l'image courante.

3.1.3. Construction de l'image finale

3.1.3.1. Notations

A l'instant t pour un pixel p, on pose les notations suivantes, qui seront utilisées dans le reste de ce manuscrit.

A l'encodeur

- $-I_{refInit}$, désigne l'image de référence construite pendant la phase d'initialisation (filtrage récursif du premier ordre);
- I, représente l'image courante;
- $-I_{ROI}$, désigne l'image courante envoyée contenant seulement la ROI;
- $-M_{ROI}$, représente le masque de la ROI (image binaire);
- $-I_{ref}$, désigne l'image de référence maintenue par mixture gaussienne.

Au décodeur

- $-\hat{I}$, représente l'image finale reconstruite;
- \hat{I}_{ROI} , désigne l'image courante reçue décompressée;
- $-\hat{M}_{ROI}$, représente le masque reconstruit implicitement (image binaire);
- $-\hat{I}_{refInit}$, désigne l'image de référence.

3.1.3.2. Construction

Théoriquement, la construction d'une image finale peut se faire dans le domaine d'ondelettes mais elle est difficile à implanter parce que nous avons deux catégories de coefficients d'ondelettes : ceux qui appartiennent à la référence et ceux qui appartiennent à la région d'intérêt. Nous utilisons le domaine spatial pour reconstruire l'image finale.

Nous disposons d'une image courante décodée (images objets), une image de référence et un masque reconstruit. La construction se fait par un remplacement de pixel à pixel :

$$\hat{I}(p,t) = \begin{cases} \hat{I}_{refInit}(p,t) & \text{si } \hat{M}_{ROI}(p,t) = 0\\ \hat{I}_{ROI}(p,t) & \text{si } \hat{M}_{ROI}(p,t) = 1 \end{cases}$$
(3.2)

3.2. Réactualisation d'images de références

Dans notre système, deux images de références co-existent : celle du système local et celle du système distant. La mise à jour d'une image de référence au décodeur doit être effectuée aussi régulièrement que possible afin de maintenir la cohérence de l'image finale. Celle de l'encodeur est mise à jour à chaque trame traitée selon la mixture de gaussiennes.

La technique intuitive de réactualisation de l'image de référence au décodeur consiste à transmettre celle-ci avec une fréquence spécifique, comme par exemple toutes les 6 ou 15 images selon [FG03] ou [MPL⁺05]. C'est la méthode couramment présentée dans la littérature. Prenons, par exemple, la méthode proposée dans [MPL⁺05], les auteurs utilisent un canal pour envoyer les objets en mouvements et un deuxième canal pour assurer la mise à jour de la référence au décodeur. Dans leur système, l'image de référence du décodeur est rafraîchie toutes les 15 images. Mais cette technique n'est pas applicable dans notre système car nous ne disposons que d'un seul canal de transmission. En effet, la transmission d'une image de référence va ralentir considérablement la cadence de transmission des données de mouvement. Cette transmission nécessite un temps plus long que pour les images de mouvements, car elle doit être encodée sans la prise en compte de la ROI et avec un taux moyen (taille très supérieure à 1.2 Ko).

Pour contrer cela, nous proposons une méthode originale utilisant un masque de référence permettant de coder une partie du fond et d'utiliser également la ROI. Nous mettons à jour l'image de référence par morceaux. Cela permet de garder constant le débit de la transmission car le masque de référence est sensiblement identique en taille par rapport au masque de mouvement moyen. En utilisant le même processus de compression, on obtient une image de référence ayant globalement la même taille en octets c'est-à-dire inférieure à 1.2 Ko. De ce fait, la durée de transmission est également d'une seconde pour la réactualisation d'un morceau de l'image de référence. Le choix de ces morceaux doit être réalisé de façon à prioriser les régions pertinentes. Nous proposons d'associer à chacun de ces morceaux un coefficient caractérisant cette notion de priorité. Le domaine de variation de ce dernier est [0, 1]. La valeur 1 correspond à la priorité maximale et la valeur 0 à la plus basse. Les valeurs initiales des coefficients de chaque morceau sont déterminées durant la phase d'initialisation et leur évolution est réalisée à l'aide d'opérateurs flous.

3.2.1. Rappel sur la logique floue

Introduit par Lotfi Zadeh en 1965 [Zad65], la logique floue permet contrairement à la logique booléenne d'obtenir toutes les valeurs entre 0 et 1. La logique booléenne en effet ne fournit que deux états distincts : vrai ou faux, correspondant respectivement aux valeurs 1 ou 0. Le passage de l'espace réel à l'espace flou se fait par détermination des appartenances de la variable réelle aux différents sous-ensembles flous caractérisant le domaine de variation de la variable réelle. Les sous-ensembles flous sont caractérisés par une fonction d'appartenance μ telle que :

$$\mu: x \in V \quad \to \quad \mu(x) \in [0, 1] \tag{3.3}$$

où V est un référentiel.

Par exemple, si l'on s'intéresse à la taille d'une personne, on peut découper la variation de ce paramètre en trois catégories : petite taille, taille moyenne et grande taille. En logique classique, une personne ne peut appartenir qu'à une seule catégorie à la fois (cf. Fig. 3.1). Par contre en logique floue, on peut dire qu'une



FIG. 3.1.: Formalisation de la logique floue. Pour une personne de taille $l=l_0$, on peut dire que cette personne peut avoir la taille petite et la taille normale.

personne dont la taille est proche de la limite entre deux catégories (par exemple petite taille et taille moyenne) appartient partiellement aux deux catégories. De ce fait, la logique floue représente mieux la perception et le raisonnement humain.

Elle est très utilisée dans l'informatique décisionnelle et dans l'automatique pour le contrôle de systèmes complexes. En logique floue, on peut définir différents opérateurs existants en logique booléenne, comme les ET, OU, Addition, etc (pour plus de détails sur la logique floue, voir les travaux de [Zad65]).

3.2.2. Mise à jour d'une image de référence par priorité

Notre technique de mise à jour de l'image de référence prend en compte le taux d'occupation d'objet mobile dans la scène afin de décider si une mise à jour est possible ou justifiée. Le taux d'occupation est divisé en trois catégories :

- Taux nul : aucun objet mobile dans la scène;
- Taux moyen : peu d'objets mobiles dans la scène;
- Taux élevé : beaucoup d'objets mobiles.

Nous avons proposé et développé deux méthodes distinctes pour effectuer cette mise à jour basées sur ces trois cas. La première méthode utilise un critère de confiance sur la qualité globale de l'image de référence : technique par image de référence prête et la seconde utilise par contre des coefficients de qualité par morceaux de l'image de référence : technique par coefficients de priorité. Après expérimentation, on constate que la deuxième méthode est plus robuste.

3.2.2.1. Technique par image de référence prête

Lors de la phase d'initialisation, la technique de construction de l'image de référence ne permet pas d'effacer correctement les zones de glissement. La zone de plus mauvaise qualité doit être identifiée et sa mise à jour doit être enclenchée seulement si la référence estimée par mixture gaussiennes est jugée stable et correcte.

Nous proposons ci-après une méthode pour qualifier l'image de référence en : référence prête avec un critère de stabilité qui permet de vérifier sa netteté.

3.2.2.1.1. Obtention d'une référence prête

Pour déterminer la stabilité de la référence, nous avons développé une méthode basée sur un critère de confiance C où $0 \le C \le 1$. Ce dernier indique la qualité d'image de référence courante. A l'initialisation, on affecte C = 0 et durant le processus de construction de l'image de référence, la valeur de C est mise à jour en fonction de l'amélioration de cette image.

Le processus d'évolution de C est lié au calcul de la différence entre deux images de référence successives puis cette dernière est binarisée avec un seuil entropique θ et on la note par d:

$$d(p,t) = \begin{cases} 1 & \text{si } |I_{ref}(p,t) - I_{ref}(p,t-1)| > \theta \\ 0 & \text{dans le cas contraire} \end{cases}$$
(3.4)

On évalue ensuite le pourcentage de changement constaté, à l'aide d'opérateur flou :

$$nz = NonZero\left(d(p,t)\right) \tag{3.5}$$

où NonZero indique le taux de pixels non nul. nz représente le pourcentage de pixels qui ont changé dans l'image de référence courante. Chaque fois que nz est proche de zéro, on augmente la confiance C. Ne connaissant pas la loi optimale permettant de faire accroître ou décroître la confiance durant le processus de construction, nous proposons une méthode basée sur la logique floue.



FIG. 3.2.: Construction de l'appartenance de la confiance C

Cette stratégie permet de mettre en place un raisonnement calqué sur l'approche humaine.

Pour le domaine de variation de nz, nous définissons deux sous-ensembles flous :

-nz appartient au sous-ensemble : "Null",

-nz appartient au sous-ensemble : "Positif Grand".

Nous définissons les règles de raisonnement qui permettent de faire évoluer le critère C (voir Fig. 3.2).

$$\begin{cases} Si nz \text{ est NULL alors } C \text{ est augment} \\ Si nz \text{ est POSITIF GRAND alors } C \text{ est diminu} \end{aligned}$$
(3.6)

Pour augmenter ou diminuer la confiance C, nous faisons donc appel à des opérateurs de la logique floue. Le raisonnement utilisé est basé sur le mécanisme de Sugeno [Pat94], lequel à son tour est fondé sur la moyenne pondérée des sorties des règles. La prémisse "nz est NULL" est caractérisée par l'appartenance μ_{Null} de nz au sous-ensemble "NULL". De même, μ_{PG} représente l'appartenance de nzau sous-ensemble "Positif Grand".

Les conclusions de la première règle sont définies par α_{Null+} pour "C est augmenté fortement" et α_{Null-} pour "C est augmenté faiblement". La valeur de sortie β_{Null} est la suivante :

$$\beta_{Null} = \frac{\mu_{Null} \times \alpha_{Null+} + \mu_{PG} \times \alpha_{Null-}}{\alpha_{Null+} + \alpha_{Null-}}$$
(3.7)

où $\alpha_{Null+} = 0.9$ et $\alpha_{Null-} = 0.1$ pour l'expérimentation.

Comme on souhaite augmenter la confiance, l'opérateur le mieux adapté est l'opérateur OU optimiste qui est défini par l'Eq. 3.8.

$$C_{NULL} = \beta_{Null} + C - \beta_{Null} \times C \tag{3.8}$$

En effet, le OU neutre ne permet pas d'obtenir une augmentation de C si β_{Null}

est égal à C car il donne la valeur maximale. Dans ce cas, l'information contenue dans β_{Null} n'est pas prise en compte.

Les conclusions de la seconde règle sont définies par α_{PG+} pour "C est diminué fortement" et α_{PG-} pour "C est diminué faiblement". La valeur de sortie β_{PG} est la suivante :

$$\beta_{PG} = \frac{\mu_{Null} \times \alpha_{PG+} + \mu_{PG} \times \alpha_{PG-}}{\alpha_{PG+} + \alpha_{PG-}}$$
(3.9)

où $\alpha_{PG+} = 0.1$ et $\alpha_{PG-} = 0.9$ pour l'expérimentation.

Comme, on souhaite diminuer la confiance, l'opérateur mieux adapté est l'opérateur ET pessimiste ("And connective") qui est défini par l'Eq. 3.10.

$$C_{PG} = \beta_{PG} \times C \tag{3.10}$$

En effet, le ET neutre ne permet pas d'obtenir une diminution de C si β_{PG} est égal à C car il donne la valeur minimale. Dans ce cas, l'information contenue dans β_{PG} n'est pas prise en compte.

La valeur de C est déterminée à partir du raisonnement utilisant les sorties vraies :

- Si C_{Null} est vraie alors la sortie est "Augmenter" soit +1
- Si C_{PG} est vraie alors la sortie est "Diminuer" soit -1

La valeur de C est obtenue à l'aide la formule suivante :

$$C = \frac{C_{Null} \times (+1) + C_{PG} \times (-1)}{C_{Null} + C_{PG}}$$
(3.11)

Illustration de la méthode

Nous représentons dans les figures Fig. 3.3 et Fig. 3.4 l'illustration de l'obtention du raisonnement. Afin d'observer la variation de la confiance C, nous faisons varier le pourcentage de pixels nz qui ont changé dans l'image de différence d(p,t). Les variations de nz représentent respectivement :

- image de différences perturbée : présence de bruit $(nz \text{ supérieur à } nz_0)$,
- amélioration de l'image de différences : diminution du bruit $(nz \text{ décroît et devient inférieur à } nz_0)$,
- dégradation de l'image de différences : départ d'un objet intégré au fond (nz augmente),
- amélioration de l'image de différences : objet intégré au fond quitte l'image $(nz \text{ décroit et devient inférieur à } nz_0).$

La Fig. 3.3 représente l'étape de fuzzification : passage d'une valeur réelle aux

valeurs floues. Cette étape est caractérisée par la notion d'appartenance aux sousensembles flous respectivement NULL et POSITIF GRAND. L'axe des abscisses correspond à l'index des images. En mettant en œuvre le raisonnement proposé, l'évolution de nz implique celles de C_{Null} et de C_{PG} . Avec notre application numérique, la confiance C suit une évolution bornée entre 0.1 et 0.84, respectivement associés à $nz \ge nz_0$ et à nz = 0 (voir Fig. 3.4).

Lorsque C dépasse le seuil 0.80, en ayant ignoré les variations de régime transitoire au démarrage, la référence est alors considérée comme fiable et prête à être envoyée au décodeur. Nous l'appelons alors "*image de référence prête*" et la notons I_{refR} ("Ready").

La possibilité d'envoi au décodeur est visible sur la Fig. 3.4-c) par le passage à 1 du drapeau "Transmission de I_{refR} " (courbe bleue). Après cet instant, les valeurs de C supérieures au seuil sont ignorées et ne provoquent pas de transmission. Il faut attendre la diminution et le passage de C en dessous du seuil de réarmement 0.2 (Fig. 3.4-c, courbe verte) pour réactiver la possibilité d'une nouvelle transmission. Le passage sous le seuil de réarmement indique que de grands changements sont intervenus dans l'image de référence (dans notre simulation : mise en mouvement d'un objet initialement intégré au fond). De ce fait dès que C passera de nouveau au-dessus du seuil de 0.8 l'image I_{refR} sera transmise (second passage à 1 de "Transmission de I_{refR} ").

3.2.2.1.2. Régions de mauvaise qualité

A présent, nous disposons d'une image I_{refR} de meilleure qualité que $I_{refInit}$ déjà transmise. Nous devons maintenant identifier les régions à mettre à jour en priorité dans l'image $I_{refInit}$. Ces régions sont différentiables car soit elles étaient occupées par de grands objets pendant la phase d'initialisation (**cas 1**) soit des objets mobiles se sont arrêtés et ils sont de ce fait incorporés dans la référence (**cas 2**). Nous présentons ci-après une méthode permettant de les identifier.

Cas 1:

Lors de l'obtention de la première référence, on construit et mémorise l'historique du mouvement de tous les objets de grande taille. Pour déterminer cet historique, on met en œuvre un algorithme de détection de contour, par exemple celui de Canny-Deriche, afin de qualifier la taille des objets (liée à la longueur du contour). L'image $M_H(p,t)$ contenant l'historique de mouvement à l'instant test une image binaire obtenue par addition pixel à pixel du masque des objets de grande taille M_{Gr} à l'instant t avec l'image précédente $M_H(p, t - 1)$.



FIG. 3.3.: Fuzzification de nz aux μ_{PG} et μ_{Null} .



FIG. 3.4.: Evolution de la confiance C en fonction de nz. La mise à jour de l'image de référence du système distant est enclenchée lorsque C dépasse un seuil de 0.80.

$$M_H(p,t) = M_H(p,t-1) + M_{Gr}(p,t)$$
(3.12)

Cas 2 :

Les objets qui se sont arrêtés durant la mise à jour par mixture de gaussiennes, correspondent à des informations complémentaires devant être incorporées dans l'image $I_{refInit}$ à transmettre au décodeur. Nous proposons de définir un masque M_i contenant cette information et de le construire comme suit :

$$M_i(p,t) = \begin{cases} 1 & \text{si } |I_{refR}(p) - I_{refInit}(p)| > \theta_i \\ 0 & \text{dans le cas contraire} \end{cases}$$
(3.13)

où θ_i est un seuil, obtenu à partir d'une comparaison de l'histogramme de I_{refR} et $I_{refInit}$.

Dès lors, la prochaine zone à mettre à jour est une partie de l'image I_{refR} calculée à l'aide du masque M_{Ref} qui résulte d'un ET bit à bit entre les deux masques définis précédemment.

$$M_{Ref}(p,t) = M_H(p,t) \wedge M_i(p,t) \tag{3.14}$$

3.2.2.1.3. Stratégie de la mise jour

La mise à jour de l'image de référence au décodeur est activée dès que le critère de confiance C est supérieur au seuil de 0.8. L'image transmise est encodée au format JPEG2000 avec prise en compte de la ROI. Cette dernière correspond au masque M_{Ref} . Le décodeur doit pouvoir faire la différence entre un masque de mouvement et un masque de référence (fond). Nous utilisons pour cela le FLAG défini au §2.3.4. La valeur 0 correspondant à l'information de mouvement, nous choisissons la valeur 1 afin d'identifier l'information relative à l'image de référence.

La reconstruction de l'image de référence au décodeur utilise la même technique que pour l'image de mouvement : construction implicite du masque, substitution des pixels de l'image de référence par ceux de l'image reçue et qui appartiennent au masque.

Après la première mise à jour, la valeur C évoluera selon le processus décrit précédemment. Elle devra dans un premier temps passer en dessous du seuil de réarmement, puis dans un second temps repasser le seuil, forçant la mise à jour. La zone à mettre à jour est déterminée à l'aide du masque M_{Ref} , basé cette fois-ci seulement sur les informations complémentaires.
$$M_{Ref}(p,t) = M_i(p,t) \tag{3.15}$$

3.2.2.1.4. Bilan

Dans cette approche, nous forçons la mise à jour d'une image de référence du décodeur dès que l'image de référence du codeur est jugée correcte. Cette technique provoque la perte d'une image de mouvement mais comme elle met en œuvre la gestion de la ROI, la taille de l'image de référence transmise est très inférieure à celle sans prise en compte de la ROI [MPL⁺05]. La cadence de rafraîchissement au niveau du décodeur ne sera pas trop perturbée. Il se peut que le volume des données transmises soit supérieur aux 1.2 Ko du fait de la nonmaîtrise de la taille de la ROI. La gestion de cette taille permettrait une maitrise de la cadence.

3.2.2.2. Technique par coefficients de priorité

Nous présentons ci-après une évolution de la technique précédente avec la définition de masques de ROI qui permettent de borner la taille maximale de la zone à réactualiser.

3.2.2.2.1. Description de l'approche

Nous proposons de découper le masque de ROI en blocs dont la taille conduira à une image JPEG2000 inférieure à 1.2 Ko. Soit Nb, le nombre total de blocs et Szb, la taille du bloc (typ. 32×32). On peut réécrire l'image de référence de cette façon :

$$I_{ref} = \bigcup_{i=0}^{Nb-1} I_{refB}(i) \qquad M_{ROI} = \bigcup_{i=0}^{Nb-1} M_B(i)$$
(3.16)

où I_{refB} et M_B représentent respectivement la sub-image de référence découpée et la sub-image du masque de la région d'intérêt découpée, *i* est l'index du bloc traité. On utilise ces notations pour le reste de ce manuscrit.

La Fig. 3.5 montre la chaîne du traitement pour obtenir la priorité des blocs à mettre à jour. Pour chaque image de référence estimée, on calcule dynamiquement le coefficient de priorité ϑ ($0 \le \vartheta \le 1$) de chaque bloc. Ce coefficient prend en compte l'amélioration de l'image de référence dans chaque bloc et il est similaire au critère de confiance défini au §3.2.2.1. Les régions ayant subi une forte perturbation, généralement affectées par le mouvement de l'objet mobile, sont ainsi localisées.



IRD: image de référence du système distant coefficients de priorité pile de priorité Régions à mettre à jour

FIG. 3.5.: Schéma-bloc de la mise à jour par priorité.

3.2.2.2.2. Calcul du coefficient de priorité

Soit ϑ_i le coefficient de priorité d'un bloc *i*. Ce coefficient est le complémentaire du coefficient de confiance C_i de ce bloc :

$$\vartheta_i = 1 - C_i \tag{3.17}$$

 C_i est calculé de la même façon que l'Eq. 3.11 à la différence que l'opérateur NonZero est appliqué au niveau du masque de mouvement du bloc i:

$$nz_i = NonZero\left(M_B(i)\right) \tag{3.18}$$

On utilise les mêmes sous-ensembles flous et le même mécanisme de raisonnement pour calculer les différentes sorties de règles :

- Pour la première règle, C_i évolue suivant la relation suivante :

$$C_{iNull} = \beta_{iNull} + C_i - \beta_{iNull} \times C_i \tag{3.19}$$

– Pour la deuxième règle, C_i évolue suivant la relation suivante :

$$C_{iPG} = \beta_{iPG} \times C_i \tag{3.20}$$

-La valeur finale de ${\cal C}_i$ est obtenue à l'aide de la formule suivante :

$$C_{i} = \frac{C_{iNull} \times (+1) + C_{iPG} \times (-1)}{C_{iNull} + C_{iPG}}$$
(3.21)

Etant estimée à l'aide de mixture de gaussiennes, l'image de référence est fortement perturbée dans les zones affectées par le mouvement. Ces régions, par conséquent, présentent un degré de confiance assez faible et cela conduit à un coefficient de priorité élevé (proche de 1). La mise à jour par morceaux d'une image de référence est effectuée selon la priorité utilisée (ϑ plus élevé ou ϑ plus faible). Inspirée par la technique d'une pile d'instabilité qui a été développée par Chou et Brown [CB90], nous empilons les blocs de telle sorte que les coefficients de priorité les plus élevés se retrouvent en haut de la pile.

3.2.2.3. Prise de décision

On effectue la mise à jour d'une partie de l'image de référence en fonction du taux d'occupation d'objet mobile θ_{Etat} . Ce dernier est déterminé à l'aide de l'opérateur flou *NonZero* appliqué au masque ROI :

$$\theta_{Etat} = NonZero(M_{ROI}) \tag{3.22}$$

A partir de ce paramètre, on distingue trois états de configurations : aucun objet mobile, peu ou beaucoup d'objets mobiles. On introduit deux seuils θ_F et θ_E pour caractériser ces configurations. θ_F est le seuil en dessous duquel on considère qu'aucun objet mobile n'est présent, tandis que θ_E est le seuil pour lequel beaucoup d'objets mobiles sont détectés. Le choix des deux paramètres (θ_F, θ_E) est très important et les trois configurations résultantes sont :

- 1. $\theta_{Etat} < \theta_F$: on considère qu'on n'a aucun objet mobile dans la scène;
- 2. $\theta_F < \theta_{Etat} < \theta_E$: on considère qu'on a une occupation d'objets mobiles moyenne dans la scène;
- 3. $\theta_{Etat} > \theta_E$: on considère qu'on a beaucoup d'objets mobiles.

Choisir le seuil θ_F trop faible conduit à ne pas activer la première configuration. La transmission d'un morceau de l'image de référence n'est pas activée et on ne réactualise pas l'image de référence du décodeur, alors que l'on dispose de la bande passante (pas de ROI de mouvement à transmettre).

Prendre le seuil θ_E trop proche de θ_F implique que la troisième configuration est souvent active. Dans ce cas, la mise à jour de l'image de référence est très restreinte, voire impossible car la ROI de mouvement est très importante et prioritaire dans notre contexte.

Lorsque la configuration médiane est active, il est alors difficile de choisir une stratégie, car la ROI de mouvement est importante (a priori pas de mise à jour de la référence) mais la perte d'une transmission de la zone de mouvement n'est pas pénalisante (taille ROI moyenne).

Pour respecter la contrainte de taille après encodage JPEG2000, le masque M_{ROI} ne doit pas dépasser 20%. On choisit une définition de la surface du bloc

de façon à respecter cette contrainte et à pouvoir transmettre jusqu'à cinq blocs simultanément. Ce choix permet d'obtenir une meilleure qualité lors de l'encodage par la gestion de la région d'intérêt du JPEG2000 et d'actualiser des zones nonconnexes de la référence.

La stratégie de réactualisation d'un morceau, en lien avec le coefficient de priorité, est réalisée suivant chacun des cas précédents.

3.2.2.2.4. Région à mettre à jour

Aucun objet mobile sur la scène $\theta_{Etat} < \theta_F$

C'est une situation **favorable**. Les blocs prioritaires sont ceux dont le coefficient de priorité est le plus élevé. Si nous avons plusieurs ex aequo, nous effectuons un tirage aléatoire parmis les candidats à la mise à jour. Cette opération permet l'amélioration de régions de mauvaise qualité affectées par les mouvements principaux. Dans notre contexte applicatif, cette région correspond à la route.

Beaucoup de mouvement $\theta_{Etat} > \theta_E$

C'est une situation **cruciale**, en principe aucune mise à jour du fond ne peut être enclenchée. Si cette configuration dure longtemps, notre stratégie est de forcer une mise à jour toutes les n images (typ. n = 15) [MPL⁺05]. Dans cette configuration, les blocs sélectionnés sont ceux dont le coefficient de priorité ϑ_i est proche de zéro. Ces zones correspondent aux régions non affectées par les mouvements : le bas-côté. La technique mixture de gaussiennes est très perturbée par la ROI de mouvement et seule, la partie non affectée par cette ROI est de bonne qualité. Cette stratégie permet de combiner, pour la visualisation au décodeur, les informations contenues dans la ROI de mouvement (actualisées très souvent car $\theta_{Etat} > \theta_E$) et celles de la mise à jour forcée. Cela conduit à une amélioration de la qualité globale de l'image reconstruite.

Mouvement moyen sur la scène $\theta_F < \theta_{Etat} < \theta_E$

C'est une situation **délicate**. La prise de décision étant difficile, nous utilisons la technique intuitive avec une mise à jour enclenchée toutes les n images (typ. n = 15) [MPL⁺05]. Les blocs à mettre à jour sont ceux ayant un coefficient de priorité maximal (idem cas $\theta_{Etat} < \theta_F$).

3.2.2.2.5. Algorithme de mise à jour par blocs

Le pseudo-code de l'Algo. 4 montre l'étape de la mise à jour par blocs de l'image de référence du décodeur. Après chaque mise à jour les coefficients de

TAB. 3.1.: Valeur des paramètres du modèle (image prête et coefficient de priorité).

Paramètre	θ_F	θ_E	n	Seuil sur C	Seuil pour réarmement	Szb
Valeur	1%	20%	15	0.64	0.4	32

priorité sont remis à zéro. Le Tab. 3.1 montre la valeur des paramètres utilisés dans le cadre de cette étude.

3.3. Conclusion

Nous avons introduit dans ce chapitre notre système de décodage permettant la reconstruction de l'image finale. Cette dernière est déterminée par l'utilisation combinée d'une image de référence, d'une image JPEG2000 décodée et d'un masque reconstruit implicitement. De plus, nous avons proposé une stratégie de mise à jour de l'image de référence au niveau du système local (mixture de gaussiennes) et au niveau du système distant (transmission par blocs-ROI avec notion de priorité).

La mise à jour de l'image de référence du distant est effectuée en fonction de l'évolution du mouvement dans la scène. Elle est réalisée par blocs de forme carrée. On définit une priorité pour chaque bloc constituant un morceau d'image de référence de l'encodeur. Les priorités sont calculées à l'aide d'un coefficient flou : *coefficient de priorité*. Ce dernier est défini selon le raisonnement de Sugeno appliqué à un système de règles floues. Un groupe de blocs forme alors la région d'intérêt pour la compression JPEG2000. Ainsi, l'image de référence est codée selon le même principe qu'une image de mouvement (fonctionnement nominal de notre système). Cette technique permet de garder constant le débit de la transmission, car la taille des données à transmettre est identique dans les deux cas de figure (mouvement ou fond). Ceci constitue une des contributions originales de ce travail de recherche. Dans le chapitre suivant, nous présenterons les applications et les résultats expérimentaux de notre méthode.

1: Définir les seuils de décision θ_F et θ_E 2: Définir une variable pour chaque cas case 3: Déterminer nz 4: Découper en blocs l'image de référence Nb 5: **pour** chaque bloc *i* **faire** 6: Calculer les coefficients de priorité ϑ 7: fin pour 8: Empiler les coefficients de priorité ϑ 9: Evaluer le taux d'occupation totale d'objet dans la scène θ_{Etat} 10: si $(\theta_{Etat} < \theta_F)$ alors Chercher les 5 blocs dont les ϑ sont les plus hauts de la pile 11: 12:Construire le masque correspondant en remplissant le carré par une valeur de 1 Mettre à zéro la valeur des ϑ sélectionnés 13:14: $case \leftarrow 1$ 15: **sinon** si $(\theta_F < \theta_{Etat} < \theta_E)$ alors 16:Chercher les 5 blocs dont les $\vartheta > 0$ 17:18: Refaire 12 : Refaire 13: 19: $case \leftarrow 2$ 20: sinon 21: si $(\theta_{Etat} > \theta_E)$ alors 22: Chercher les 5 blocs dont les $\vartheta = 0$ 23:Refaire 12 : 24:25:Refaire 13: 26: $case \leftarrow 3$ fin si 27:28:fin si 29: fin si 30: si (case == 0) alors 31: Enclencher la mise à jour 32: fin si 33: si (case == 1) alors Lancer le compteur n par $n \leftarrow (n+1)$ 34: si (n = 15) alors 35: 36: Enclencher la mise à jour 37: fin si 38: fin si 39: si (case = 2) alors Lancer un timer 40: si $timer = 15 \times T_e$ (T_e ; période d'échantillonnage) alors 41: Enclencher la mise à jour 42: fin si 43: 44: **fin si**

Algo. 4: Pseudo-code pour la mise à jour par bloc d'une image de référence

4. Expérimentation et Résultats

Sommaire

4.1.	Première expérimentation : vidéo enregistrée 94
4.2.	Deuxième expérimentation : vidéo en ligne 117
4.3.	Conclusion 121

L'application typique visée par cette étude est la transmission de vidéo à très bas débit pour un système embarqué, par exemple, la réception d'une vidéo avec un PDA ou téléphone portable. Nous présentons dans ce chapitre les résultats expérimentaux de notre système de codage. Nous effectuons notre expérimentation selon deux modes.

Le premier concerne l'utilisation de séquences vidéos pre-enregistrées, car cela nous permet d'étudier les différents cas de figure de notre contexte applicatif. Il sera alors plus simple de contrôler et de mesurer la qualité de nos résultats à l'aide de critères objectif et subjectif. La mise en œuvre sur des ordinateurs de développement permet de s'affranchir des problèmes directement liés aux restrictions matérielles. Cela conduira à la validation des choix et des paramètres des algorithmes présentés précédemment.

Le second mode utilise un dispositif expérimental industriel développé à partir de composants au format PC104. C'est le standard retenu par l'industriel. Nous nous plaçons alors dans les conditions normales de fonctionnement et l'acquisition des données d'entrée est réalisée à l'aide d'une caméra également au standard industriel. Nous pourrons alors conclure sur les performances de notre approche sous les contraintes matérielles.

La transmission devra s'appuyer sur un réseau sans fil (type GSM), mais pour valider notre système nous utilisons une liaison série bridée pour ramener la bande passante à une valeur proche de celle constatée sur le réseau sans fil.

4.1. Première expérimentation : vidéo enregistrée

Après une présentation rapide des aspects concernant le matériel et le logiciel, nous présentons les résultats obtenus pour chaque bloc de traitement de notre algorithme présenté dans la Fig. 2.1.

4.1.1. Dispositif expérimental

4.1.1.1. Aspects matériels et outils de développement

Les deux ordinateurs qui jouent respectivement le rôle du système local et du système distant, utilisent le système d'exploitation Windows XP \bigcirc et l'environnement de développement retenu est Microsoft Visual Studio 6.0 \bigcirc . Le langage de programmation est le C++. Ces choix sont liés aux contraintes de marché de l'industriel.

Au niveau des ressources matérielles, les ordinateurs sont de type :

- Processeur : AMD 1.6 GHz,
- RAM : 512 Mo.

Le réseau sans fil est simulé à l'aide d'une liaison série de type RS232 dont le débit est fixé à 9600 bit/s.

Lors des développements, nous ferons particulièrement attention à la gestion de la mémoire et nous veillerons à dissocier les éléments d'interface graphique et le corps de notre système.

4.1.1.2. Séquences vidéo utilisées

Nous utilisons pour les tests trois séquences différentes, dont deux sont issues de l'état de l'art et elles ont fait l'objet de publications; ce qui nous permet d'envisager un étalonnage de nos résultats :

- Une séquence du laboratoire LAPS Bordeaux-1¹, Fig. 4.1-a : séquence prise avec une illumination quasi-constante pour représenter les trois cas (pas, peu et beaucoup) du taux d'occupation de l'ensemble des objets mobiles dans la scène;
- 2. Une séquence fournie par l'industriel MAGYS [MAG], Fig. 4.1-b : séquence prise depuis un pont d'autoroute pour représenter un léger mouvement de la caméra (vibration du pont). Cette séquence contient de grands objets mobiles au début de la scène. Elle représente les cas de peu ou beaucoup d'objets mobiles dans la scène.

¹Merci à MM. Izquierdo et Berthoumieu du LAPS de nous avoir fourni la séquence.



(a) Séquence d'images LAPS

(b) Séquence d'images MA-GYS



(c) Séquence d'images LA-BOINFO

FIG. 4.1.: Séquence d'images testées.

 Une séquence d'images LABOINFO de l'université de Karlsruhe², Fig. 4.1c : séquence prise avec la pluie pour représenter la forte variation d'illumination.

4.1.2. Résultats : Phase d'initialisation

4.1.2.1. Construction d'une image de référence

Nous avons implanté deux méthodes de la littérature : filtrage récursif du premier ordre sans recherche d'objet mobile, mixture de gaussiennes pour la construction d'une image de référence. Ces méthodes sont comparées à notre approche : filtre récursif du premier ordre avec recherche d'objet mobile. Les Fig. 4.2, 4.3 et 4.4 montrent respectivement le résultat obtenu pour les séquences LAPS, MAGYS et LABOINFO.

²Disponible sur http://i21www.ira.uka.de/image_sequences/; séquence d'images avec droits réservés.

Technique par filtre récursif sans recherche d'objet mobile

Comme le montrent les Fig. 4.2-b, 4.3-b et 4.4-b (voir les images agrandies), les régions affectées par le mouvement de l'objet mobile sont encore présentes dans la référence. De ce fait, cette technique ne peut être retenue.

Technique par mixture de gaussiennes

Représentée dans les Fig. 4.2-c, 4.3-c et 4.4-c, cette technique ne permet pas d'obtenir un bon résultat avec une courte séquence (nombre d'images L = 50) dès qu'il y a un objet en mouvement dans la scène. De ce fait, cette technique ne peut être retenue.

Technique par filtre récursif avec recherche d'objet mobile

Selon les Fig. 4.2-d, 4.3-d et 4.4-d, cette technique donne un meilleur résultat pour les trois séquences. Cela est particulièrement significatif si la scène contient de grands objets qui se déplacent très lentement, comme c'est le cas sur la séquence MAGYS, Fig. 4.3-d. Pour cette séquence, notre approche donne un résultat semblable à celui obtenu par la première technique sur la séquence LAPS, identifiée comme *simple* (peu d'objets mobiles).

4.1.2.2. Evaluation de la qualité

L'obtention d'une image de référence correcte dépend également de la fréquence d'acquisition des images. Dans cette étude, nous avons une fréquence d'acquisition de 8 img/s et les résultats montrent que c'est largement suffisant pour notre application.

Pour mesurer la qualité de l'image de référence, nous utilisons la mesure objective PSNR (défini par l'Eq. 1.3). On calcule le PSNR de l'image contenant la différence entre l'image de référence issue de notre technique et une image sans objets mobiles extraite de la séquence. Dans la séquence LABOINFO, il n'existe pas d'image sans objets mobiles : nous ne pouvons pas déterminer le critère PSNR. Le Tab. 4.1 montre le PSNR des séquences LAPS et MAGYS. Pour la séquence LAPS, on obtient un PSNR supérieur à 20 dB ce qui est significatif de bonne qualité alors que pour la séquence MAGYS on a un PSNR légèrement inférieur à 20 dB indiquant une qualité limite. Cela s'explique par le fait que cette séquence contient de grands objets mobiles pendant la construction de la référence. Cependant, les résultats sont satisfaisants pour notre application.



FIG. 4.2.: Construction d'une image de référence avec L = 50; séquence LAPS. a) image courante (une image dans la séquence); b) image de référence obtenue par le filtre récursif du premier ordre sans recherche d'objet mobile; c) image de référence obtenue par mixture de gaussiennes; d) image de référence obtenue par le filtre récursif du premier ordre avec recherche d'objet mobile (c'est notre modèle).

TAB. 4.1.: PSNR de l'image de référence construite par rapport à une image sans voitures dans la séquence.

Séquences	LAPS	MAGYS	LABOINFO
PSNR (en dB)	35.90	19.31	Non disponible

La séquence MAGYS contient de grands objets mobiles. On constate que notre modèle permet la meilleure construction d'une référence, voir figure zoom sur d).



FIG. 4.3.: Construction d'une image de référence; séquence MAGYS.

Dans cette séquence, les trois techniques présentées ici donnent pratiquement le même résultat ; les wagons du train (grands objets) sont intégrés dans l'image de référence, ce qui présente la limite de notre modèle dans une condition extrêmement difficile.



FIG. 4.4.: Construction d'une image de référence; séquence LABOINFO.

Тав. 4.2.: Temps	d'exécution	de la	construction	d'une	image	de référence.
------------------	-------------	-------	--------------	-------	-------	---------------

Séquences	LAPS	MAGYS	LABOINFO
Temps construction (ms)	2700	2450	2750

4.1.2.3. Temps de construction

Nous avons fixé la longeur de la séquence à 50 images et la fréquence d'acquisition à 8 img/s. Le Tab. 4.2 présente la durée de construction de l'image de référence avec notre dispositif expérimental. Les deux catégories vidéos (LAPS et LABOINFO) ont la même résolution spatiale de 320×256 . Tandis que la séquence MAGYS a une résolution spatiale de 320×240 . Les séquences sont codées en 24 bits et chaque composante couleur est codée en 8 bits. Ces temps seront à mettre en relation avec ceux obtenus avec le dispositif industriel.

4.1.3. Résultats : Gestion de la ROI

4.1.3.1. Encodeur : la construction

L'obtention correcte de la région d'intérêt est très importante dans notre système. Elle est réalisée avec la détection de mouvement. Cette dernière est effectuée à l'aide de la technique de soustraction de fond, qui est combinée avec l'observation obtenue à partir de l'information de la différence entre deux images successives à l'instant précédent et à l'instant courant. Cette étape permet d'obtenir un masque de régions mobiles dans la scène. L'entrée du codeur JPEG2000 est constituée d'une part par le masque ROI et d'autre part par l'image dont les pixels n'appartenant pas au masque ont été affectés à la valeur médiane de la dynamique. Le codage est effectué avec activation de l'option ROI et avec un taux de compression très élevé 1 : 250. Ces paramètres conduisent à l'obtention d'une taille de données inférieure à la limite 1.2 Ko. La Fig. 4.5 montre les résultats d'image JPEG2000 obtenues à partir des scènes LAPS, MAGYS et LABOINFO. On constate pour la dernière séquence que les wagons du train ne sont pas détectés correctement car ils sont quasiment uniformes avec le fond et leur vitesse de déplacement relativement faible.

4.1.3.2. Décodeur : la reconstruction implicite

En théorie, la technique de codage de la gestion de région d'intérêt dans la norme JPEG2000 permet la reconstruction du masque sans connaissance a priori



Image courante 320×256 24 bits



Masque du mouvement

(a) Séquence LAPS



Image JPEG2000 à 1:204 Taille obtenue: 1288 octets



Image courante 320×240 24 bits

Masque du mouvement

(b) Séquence MAGYS



Image JPEG2000 à 1:188



Image courante 320×256 24 bits



Masque du mouvement



Image JPEG2000 à 1:204 Taille obtenue: 1260 octets

(c) Séquence LABOINFO

FIG. 4.5.: Exemples des images JPEG2000.

Séquences	LAPS	MAGYS	LABOINFO
$\Gamma(\%)$	0.9	1.9	1.02

TAB. 4.3.: Différence entre le masque initial et le masque reconstruit.

des informations spatiales utilisées lors de l'encodage. Mais la reconstruction implicite du masque ne peut plus être parfaite si un coefficient d'ondelettes est nul après la quantification du fond ou du masque (cf. §3.1.2). L'erreur entre le masque initial et celui reconstruit est minimale car les coefficients d'ondelettes de la ROI sont codés par plans de bits. Ce codage, étant réalisé en commençant par les bits de poids fort vers les bits de poids faible, permet d'obtenir un nombre maximum de coefficients de la ROI non nuls. La méthode Maxshift est cohérente avec cette hypothèse car il faut que le maximum de coefficients d'ondelettes appartenant au fond soit inférieur au minimum des coefficients d'ondelettes appartenant à la ROI. Cette condition conditionne le choix de la valeur du décalage binaire s.

Ce problème nous a conduit à mettre en place le post-traitement du masque implicite à l'aide d'opérateur de morphologie mathématique afin de limiter les écarts. Cependant, il est nécessaire de connaître la différence entre le masque initial et le masque reconstruit afin de valider l'utilisation de cette technique. Pour cela, nous utilisons le critère suivant :

$$\Gamma = \frac{1}{N} \sum_{p=0}^{N-1} \left| M_{ROI}(p) - \hat{M}_{ROI}(p) \right|$$
(4.1)

où N représente la taille d'image et p est un pixel courant.

La Fig. 4.6 représente l'exemple du masque reconstruit implicitement du côté du décodeur. Le Tab. 4.3 montre le pourcentage d'écart. D'après ces résultats, pour les trois séquences LAPS, MAGYS et LABOINFO, on trouve une variation de ce critère de 0.5% à 2% entre le masque construit initialement et celui reconstruit implicitement.

4.1.4. Résultats : Construction de l'image finale

4.1.4.1. Reconstruction d'images finales

L'image finale est construite par substitution des pixels du fond par ceux de l'image reçue validés par le masque reconstruit (Eq. 3.2). La Fig. 4.7 (voir zoom Fig. 4.8) présente une image reconstruite pour les séquences LAPS, MAGYS et LABOINFO. Afin de mesurer les effets de notre approche, nous utilisons pour la



FIG. 4.6.: Exemples d'un masque reconstruit implicitement au décodeur. a) le masque initial; b) le masque reconstruit; c) la différence entre les deux masques. On trouve essentiellement l'erreur de la recontruction sur le bord de la région d'intérêt. De haut en bas : séquences LAPS, MAGYS et LABOINFO.

TAB. 4.4.: PSNR de l'image finale construite.

Séquences	LAPS	MAGYS	LABOINFO
PSNR (dB)	32.6	28.5	30.1

reconstruction de l'image le fond utilisé pour l'obtention de la ROI à l'encodeur. Cela permet de mettre en avant les défauts liés au masque implicite.

4.1.4.2. Evaluations

Nous utilisons deux stratégies pour évaluer les résultats basées sur des critères objectif et subjectif.

4.1.4.2.1. Critère objectif PSNR

Le premier critère consiste à utiliser le PSNR pour mesurer la distorsion entre l'image initiale et l'image reconstruite. Le Tab. 4.4 présente la valeur moyenne du PSNR pour chaque séquence et pour les trois images (Fig. 4.7). On trouve une moyenne supérieure à 28 dB. Pour certaines images, on constate un léger artefact dans l'image reconstruite. Ce sont des artefacts liés aux contours externes du masque reconstruit.

4.1.4.2.2. Critères subjectifs

Nous procédons de deux manières différentes pour évaluer notre résultat (image construite) : évaluation par des experts industriels spécialisés dans le domaine de la vidéosurveillance routière et évaluation par un groupe de personnes qui ne sont pas spécialistes de traitement et d'analyse d'images. La procédure d'évaluation subjective est réalisée selon les recommandations du CCIR [BT.95]. Nous présentons aux personnes pour chaque série d'images d'une part les images originelles et d'autre part les images reconstruites sans leur indiquer quelle séquence elles observent.

Experts Industriels

Dans le cadre de notre application industrielle, nous avons soumis six images reconstruites aux experts des industriels spécialisés dans la vidéosurveillance, pour les expertiser. Ces experts sont MAGYS et certains de ses clients. Ils nous ont fourni ensuite leurs expertises.



PSNRY=31.43 dB

PSNRY=30.23 dB



a) image courante (à l'encodeur); b) l'image finale reconstruite (au décodeur); le débit de transmission est de 1 img/1.2 s.







FIG. 4.8.: Zoom sur la reconstruction d'images finales. a) image courante; b) image reconstruite. De haut en bas : séquences LAPS, MAGYS et LABOINFO.

Séquence LAPS					
Images	Illisible	acceptable	Correcte	Très bonne	
1		5(3)	8(10)		
2	1	7(8)	4(5)	1	
3	5(2)	6(10)	2(1)		
		Séquence MAGYS			
1		5(4)	8(9)		
2	2	9(11)	2(2)		
3	3(3)	9(10)	1		
		Séquence LABOINFO			
1	11(10)	2(3)			
2	8(9)	5(4)			
3	11(13)	2			

TAB. 4.5.: Résultats des votes; les valeurs entre parenthèses correspondent aux votes pour les séquences originelles.

Un groupe de personnes

On procède selon plusieurs échelles d'évaluation. Dans notre cas, pour évaluer la qualité d'image reconstruite, nous utilisons les quatre échelles suivantes : illisible, acceptable, correcte et très bonne. L'évaluation se fait par un vote parmi ces quatre propositions. De plus pour les trois séquences testées, nous avons extrait une série de trois images pour le vote.

Les participants sont des enseignants chercheurs et des doctorants au laboratoire LIPSI-ESTIA. Le groupe est composé de 13 personnes. Celles-ci expriment leur vote sans avoir eu connaissance des autres votes. Le Tab. 4.5 montre le résultat des votes.

4.1.4.2.3. Conclusion

Les résultats de ce test subjectif montrent que d'une part la qualité de l'image issue de notre processus n'est que très peu altérée et que d'autre part la qualité est au standard des dispositifs de vidéosurveillance actuellement opérationnels. Le cas de la séquence LABOINFO est très spécifique car la qualité originelle est déjà en-dessous du standard usuel.



FIG. 4.9.: Diagramme de résultats des votes.

4.1.5. Résultats : Actualisation de l'image de référence distante

Dans notre système, la mise à jour d'une image de référence du système distant joue un rôle très important. Nous avons proposé et développé deux méthodes pour cette mise à jour au décodeur :

- Méthode par image prête;
- Méthode par coefficients de priorité.

4.1.5.1. Image prête

La qualification de l'image de référence en image prête est représentée sur la Fig. 4.10 et est obtenue par le méchanisme flou basé sur le degré de confiance C. Les valeurs numériques des paramètres de ce raisonnement sont présentées dans le Tab. 4.6. Nous observons d'une part que l'évolution de C est bien cohérente avec celle de nz et d'autre part que le comportement des différents paramètres est idendique à celui observé en simulation. Sur la Fig. 4.10 la valeur de C passe une première fois le seuil de qualification image prête pour l'index 35, ce qui enclenche la possibilité de réactualisation de la référence et qui invalide le drapeau de mise à jour. Lorsque C passe en dessous du seuil de réarmement à l'index 45, le drapeau de mise à jour est validé, ce qui permet la seconde possibilité de mise à jour à l'index 85.

Le résultat pour la séquence MAGYS est représenté dans la Fig. 4.11. Le masque d'historique mémorisant le passage d'objet durant l'initialisation est présenté et la mise à jour de l'image de référence distante également (index 35 - Fig. 4.10). L'amélioration des régions impactées par de grands objets est clairement visible (Fig. 4.11-d). Cette expérimentation met en évidence la prise en compte des zones de mauvaise qualité identifiées durant le processus de création de la première image de référence.

Nous représentons dans la Fig. 4.12 une réactualisation de l'image de référence distante lorsque des objets se sont immobilisés (nous avons modifié la séquence afin de les immobiliser). Cela correspond à l'apparition d'informations complémentaires dans le fond. Ils sont donc incorporés dans l'image de référence de l'encodeur (Fig. 4.12-b). Nous constatons que notre méthode permet bien de construire le masque d'informations complémentaires qui caractérise les régions affectées par les objets immobilisés (Fig. 4.12-c). La mise à jour (index 85 - Fig. 4.10) a permis d'incorporer ces nouveaux objets dans l'image de référence du système distant (Fig. 4.12-d). Cette expérimentation met en évidence la prise en



FIG. 4.10.: Variation de la confiance associée à l'image de référence prête de la Fig. 4.11-b.

compte de l'incorporation d'objets dans le fond durant le processus de maintenance de l'image de référence.

4.1.5.2. Stratégie de transmission et gestion des coefficients de priorité

Pour cette expérimentation, nous avons appliqué à chaque bloc la méthode présentée précédement ; la détermination de la confiance C_i par bloc permet d'obtenir le coefficient de priorité ϑ_i . La Fig. 4.14 permet de visualiser la priorité des différents blocs.

Seule la séquence LAPS permet l'enclenchement de la mise à jour de l'image de référence du distant pour les trois cas proposés §3.2.2.2.4. Les séquences MAGYS et LABOINFO activent seulement le troisième cas. Le taux d'occupation de la

Région délaissée par un grand objet lors de l'initialisation





(a) Image de référence à l'encodeur à l'initialisation

(b) Image de référence prête



(c) Masque de la région à mettre à jour

Région mise à jour



(d) Référence du décodeur réactualisée

FIG. 4.11.: Mise à jour de l'image référence du système distant par image de référence prête.

Paramètres	nz_0	Seuil sur C	Seuil pour réarmement
Valeur	0.0006	0.64	0.4

TAB. 4.6.: Valeurs numériques des paramètres.



(a) Ancienne image de référence prête

(b) Nouvelle image de référence prête



(c) Masque (sans posttraitement) de nouvelles informations incorporées

(d) Image de référence réactualisée du système distant

FIG. 4.12.: Réactualisation de l'image de référence en tenant compte des nouvelles informations incorporées dans la référence du système local.



FIG. 4.13.: Taux d'occupation de l'ensemble d'objets mobiles et les flags de mise à jour.

séquence LAPS est donné par la Fig. 4.13; on y représente également les Flags correspondant aux trois cas (trait vert : seuil définissant le cas *pas d'objet* et le trait bleu définissant le cas *beaucoup d'objets*).

La configuration pas d'objet est enclenchée à l'index image 1, 10 ou 31. Cela correspond au taux d'occupation inférieur au seuil θ_F (Fig. 4.13, trait vert). La mise à jour est effectuée à ces instants (Fig. 4.13, courbe violette). Les blocs prioritaires pour réactualiser l'image de référence du décodeur sont représentés dans la Fig. 4.14-c). Ces blocs correspondent aux régions affectées par le mouvement, c'est-à-dire la route.

La configuration beaucoup d'objets mobiles est activée à l'index image 47 de la Fig. 4.13. Cela correspond au taux d'occupation supérieur à un seuil θ_E (Fig. 4.13, trait bleu). La mise à jour est donc possible à cet instant (Fig. 4.13, courbe noire). Pour notre test, nous forçons la mise à jour mais la stratégie normale est d'observer cette situation avec au moins n images (typ. n = 15). Les blocs prioritaires pour réactualiser l'image de référence du décodeur sont représentés dans la Fig. 4.15-c). Ces blocs correspondent aux régions non affectées par le mouvement c'est-à-dire le bas-côté.

Sur la Fig. 4.13, la configuration *peu d'objets mobiles* n'est pas matérialisée par un flag. Le fonctionnement normal est d'observer cette situation sur au moins nimages et d'enclencher une mise à jour similaire au cas pas d'objet mobile. Ces résultats ne sont pas présentés car ils sont comparables à ceux de la Fig. 4.14-c).



FIG. 4.14.: Coefficients de priorité et mise à jour de l'image de référence du système distant selon la configuration *pas d'objet mobile* sur la scène.



FIG. 4.15.: Coefficients de priorité et mise à jour de l'image de référence du système distant selon la configuration *beaucoup d'objets mobiles* sur la scène.

4.1.5.3. Conclusion

Nous avons présenté dans cette section notre stratégie de mise à jour de l'image de référence. Notre méthodologie diffère de celle habituellement mise en œuvre car d'une part, nous choisissons de travailler par partie d'image en intégrant la notion de blocs prioritaires et d'autre part, nous choisissons d'actualiser les régions de bonne qualité (de priorité faible) lorsque la quantité d'objets mobiles est élevée. Cette démarche permet l'obtention d'une image courante de qualité optimale vis à vis des utilisateurs.

4.1.6. Bilan sur l'expérimentation

Cette expérimentation a permis de mettre en œuvre dans un contexte d'environnement de développement l'ensemble des approches théoriques que nous avons proposées et de valider la cohérence et la qualité des images produites. Ces résultats nous ont permis de passer à la phase suivante d'expérimentation visant des tests sur un matériel industriel. Toutefois, en évaluant le temps de calcul pour chaque phase du traitement (voir Tab. 4.7), on constate que c'est la transmission qui est la plus coûteuse si l'on ne prend pas en compte la phase d'initialisation qui n'est activée qu'une seule fois au démarrage de notre système. Le coût de calcul est un point sensible de ce développement et une attention toute particulière doit être portée à la qualité de la programmation. La transmission dépasse la seconde car nous n'avons pas bridé la taille de la ROI des objets en mouvement.

TAB. 4.7.: Temps d'exécution de l'ensemble du traitement, avec un taux de compression de 1 : 188.

Séquences	LAPS	MAGYS	LABOINFO
Traitements	(ms)	(ms)	(ms)
Initialisation	2700	2450	2750
Estimation référence	90	250	230
Encodage	400	450	400
Recherche de priorité par blocs	90	80	92
Transmission	1200	1100	1200



FIG. 4.16.: L'encodeur : système matériel.

4.2. Deuxième expérimentation : vidéo en ligne

4.2.1. Dispositif expérimental

Pour le développement du système local (encodeur), nous disposons d'un matériel basé sur des cartes industrielles au format PC104. Ce prototype est composé d'une CPU Celeron 800 MHz avec une mémoire RAM 128 Mo, un disque dur compact flash (128 Mo de capacité), une carte d'acquisition vidéo et une caméra, Fig. 4.16. Le système d'exploitation retenu est Linux dont le noyau sera adapté au niveau des ressources et des besoins de notre applicatif. La transmission est toujours basée sur une liaison série RS232 bridée en terme de débit ; l'environnement logiciel et matériel du système distant est identique à celui utilisé pour l'expérimentation précédente.

4.2.2. Résultats

Nous avons implanté notre stratégie sur ce matériel et pour cela l'ensemble des algorithmes a été adapté au nouvel environnement d'exécution. La caméra industrielle offre une résolution spatiale de 352×288 . L'image couleur est codée sur 24 bits avec 8 bits par composante au standard RGB. La fréquence d'acquisition vidéo théorique est de 25 img/s mais pour notre application nous la ramenons à 8 img/s, ce qui est suffisante car il nous faut une seconde pour transmettre 1.2 Ko de données.

Nous conduisons les tests comme précédemment et nous analysons les résultats pour chaque phase du traitement (initialisation, segmentation d'objets mobiles, encodage par JPEG2000, reconstruction implicite du masque du mouvement et



FIG. 4.17.: Image de référence avec le système réel.

image finale reconstruite).

Les traits noirs présents dans l'image sont des capteurs graphiques implantés dans la caméra industrielle.

La Fig. 4.17 montre l'image de référence obtenue durant la phase d'initialisation. Nous constatons que la durée de construction est très élevée : 31 s. Cela est dû à l'écriture de l'image sur le disque dur car cette partie utilise un programme livré avec le matériel.

La Fig. 4.18 représente le masque ROI construit par l'encodeur, le masque ROI reconstruit implicitement par le décodeur et la différence entre les deux masques. Nous retrouvons les résultats semblables avec ceux de l'expérimentation précédente. L'écart, déterminé à l'aide de l'Eq. 4.1, est de 1.2%.

La Fig. 4.19 montre les images finales reconstruites et la moyenne du PSNR vaut 30 dB ce qui est bien supérieur au seuil de qualité utilisé précédemment.

La mise à jour de l'image de référence est représentée dans la Fig. 4.20. Pour ce positionnement de la caméra, la configuration *beaucoup d'objets* ne peut pas être activée et seules celles correspondant aux configurations *pas* ou *peu d'objets mobiles* le peuvent. Mais il est difficile de comparer l'image de référence à jour du décodeur avec celle de l'encodeur car aucune information complémentaire n'est incorporée.

La durée totale de l'ensemble du traitement est donnée dans le Tab. 4.8 et l'on observe globalement des temps supérieurs principalement imputables à la puissance du matériel .



(a) ROI construit par l'encodeur

(b) ROI reconstruite implicitement par le décodeur



- (c) Image de différence
- FIG. 4.18.: Masques ROI.



FIG. 4.19.: Images finales reconstruites avec le système réel. Taux de compression 1:204. Débit de la transmission : 1 image toutes les 1.2 s



(a) Image de référence du système local

(b) Image de référence mise à jour du système distant

FIG. 4.20.: Mise à jour d'une image de référence par coefficients de priorité.

TAB. 4.8.: Temps d'exécution de l'ensemble du traitement avec le système réel. Certes, les codes ne sont pas optimisés mais déjà nous pouvons analyser les résultats.

Vidéo en ligne (352x288 en 24 bits)	Durée (ms)
Initialisation	31000
Estimation référence	300
Encodage	500
Recherche de priorité par blocs	200
Transmission	1350

4.2.3. Bilan sur l'expérimentation

Cette expérimentation a permis de mettre en œuvre dans un contexte d'environnement industriel l'ensemble des approches théoriques que nous avons proposées. La cohérence et la qualité des images produites sont toujours à un niveau acceptable. Par contre, nous avons bien identifié les effets induits par la diminution sensible des performances globales du matériel industriel. Ce paramètre n'est pas trop gênant car les nouvelles générations de PC au format PC104 progressent de façon significative sur ce point. Il est toutefois très important d'être vigilant sur la qualité du code.

4.3. Conclusion

Nous avons montré dans les expériences le bien-fondé de notre stratégie quant au choix de l'utilisation du standard JPEG2000 et de la prise en compte de la région d'intérêt dans le cadre d'une application de vidéosurveillance de scène routière. La gestion de la région d'intérêt permet un gain important en terme de taille de données à transmettre grâce à la combinaison d'un fort taux de compression et d'une uniformisation du fond. Ainsi l'encodage proposé permet d'augmenter la qualité des informations transmises sans nuire au débit de la transmission.

La stratégie de mise à jour du fond par transmission au système distant d'un nombre fini de portions d'image permet d'une part de garantir le maintien de la cohérence du fond vis-à-vis des objets mobiles sur une longue période de fonctionnement et d'autre part de ne pas pénaliser la cadence de rafraîchissement des objets mobiles sur le poste distant.

Les images affichées sur le poste distant sont reconstruites par substitution des

pixels entre la ROI de l'image reçue et le fond disponible à cet instant. Nos résultats ont été évalués selon deux critères : objectif (PSNR) et subjectif (réalisé chez l'industriel MAGYS et avec une population d'utilisateurs potentiels). La qualité des images affichées est tout à fait comparable à celle des systèmes actuellement opérationnels.
Conclusion générale et perspectives

Nous avons abordé dans ce mémoire un problème important en encodage d'image dans un contexte de vidéosurveillance, celui de la compression et de la transmission. L'objectif souhaité dans cette étude est la transmission d'une vidéo à travers un réseau à très bas débit. La cadence souhaitée est de l'ordre d'une image par seconde avec le GSM. Le rendu visuel de la vidéo à la réception doit être acceptable sur la base de critères définis par l'industriel.

Dans le premier chapitre, nous avons abordé la problématique du codage, avec ou sans perte, d'image fixe ou de séquence. Cet état de l'art a permis de dégager l'orientation principale de notre étude. Après un test comparatif, le standard JPEG2000 a été retenu et de ce fait nous nous sommes focalisés sur le codage avec perte d'images fixe.

La fonctionnalité de la gestion de région d'intérêt (ROI) intégrée au standard JPEG2000 est en principe destinée à la visualisation progressive durant la décompression. Nous avons détourné cette fonctionnalité afin d'obtenir un taux de compression élevé (1 : 250), sans pour autant dégrader fortement la qualité des informations présentes dans la ROI. Dans notre cas, la région d'intérêt est constituée par l'ensemble des objets mobiles présents dans la scène. Le masque de ROI est obtenu par détection de mouvement basée sur la combinaison des informations issues de l'étude, d'une part, de la différence temporelle d'images et, d'autre part, de la différence temporelle entre l'image courante et le fond.

Dans le second chapitre, nous avons abordé la description théorique de notre système de codage. Ce dernier est décomposé en cinq blocs principaux : phase d'initialisation, estimation d'une image de référence, segmentation d'objet mobile, encodage et transmission.

L'obtention d'une image de référence pendant la phase d'initialisation doit permettre de combiner deux objectifs contradictoires : rapidité d'obtention et qualité d'image. Nous avons opté pour l'utilisation d'un filtre récursif du premier ordre que nous avons amélioré en introduisant la recherche d'objet mobile. Cette première image de référence est transmise au système distant avec une faible compression.

Le choix retenu pour la détermination du masque ROI impose une actualisation permanente de l'image de référence. L'estimation de la référence à l'encodeur est effectuée à l'aide de mixture de gaussiennes. Ce modèle permet de prendre en compte les changements liés d'une part à l'environnement naturel et d'autre part à l'intégration dans la scène d'un objet mobile qui change d'état, selon qu'il s'arrête ou repart. Afin de diminuer la taille des données à transmettre, nous avons choisi d'uniformiser les pixels n'appartenant pas au masque ROI à la valeur médiane de la dynamique des composantes couleurs. L'image ainsi obtenue est encodée au standard JPEG2000 avec gestion de la région d'intérêt puis transmise via une liaison série RS232 à 9600 bauds.

Dans le troisième chapitre, nous avons abordé la description théorique de notre système de décodage et notre stratégie pour la réactualisation d'images de références.

La première partie de ce chapitre est dédiée au décodeur qui englobe : le décodage de l'image reçue, la reconstruction implicite du masque ROI et l'affichage de l'image courante. L'apport essentiel dans cette partie réside dans la stratégie de la reconstruction implicite du masque. Cette reconstruction a été rendue possible grâce à une analyse fine du standard JPEG2000 ayant conduit au choix de la technique Maxshift.

La deuxième partie de ce chapitre est consacrée à la réactualisation de la référence du distant. Nous avons proposé une technique à deux niveaux. Le premier niveau correspond à la stratégie de déclenchement de la réactualisation. Celle-ci est fonction du taux de pixels mobiles dans l'image et est découpée en trois configurations : sans mouvement, peu de mouvement ou beaucoup de mouvement. Dans les deux premiers cas, nous avons choisi de mettre à jour principalement les régions affectées par le mouvement dans l'image de référence. Pour le dernier cas, nous avons décidé d'actualiser les régions non affectées par le mouvement car cela conduit à l'amélioration de la lisibilité de l'image courante à l'affichage. Le second niveau concerne le choix des régions à mettre à jour. Nous avons découpé l'image de référence en blocs carrés et chacun de ces blocs possède un coefficient de priorité. L'évolution de ce coefficient s'appuie sur une modélisation et un raisonnement basé sur la logique floue. Cela permet la mise en œuvre d'un mécanisme intuitif proche du raisonnement d'un expert en vidéosurveillance. Suivant la configuration du critère de premier niveau, cinq blocs sont choisis en fonction de leur priorité pour mettre à jour la référence du distant. Cette méthodologie permet de borner la taille de la ROI afin de garantir la taille des données à transmettre en respectant les contraintes de débit du réseau sans fil disponible.

Dans le quatrième chapitre, nous avons présenté les différents résultats expérimentaux. La première partie de ce chapitre est consacrée à la simulation de notre approche en utilisant d'une part des séquences enregistrées et des ordinateurs de développement. L'algorithme complet a été implanté et toutes les étapes ont été critiquées à l'aide de tests objectifs ou subjectifs. Pour la phase d'initialisation, la qualité de l'image et son temps de construction sont tout à fait compatibles avec notre contexte applicatif. Pour la gestion de la ROI, la difficulté associée à sa reconstruction implicite a été franchie et nous avons obtenu moins de 2% de différence. L'image courante reconstruite à l'aide de ce masque implicite ne présente pas de différences notables avec l'image d'origine. Pour la maintenance de l'image de référence, notre stratégie d'actualisation par blocs à priorité permet de combiner d'une part, une mise à jour avec la perte d'une seule image de mouvement et d'autre part, de mettre à jour les régions pertinentes vis-à-vis de l'utilisateur. Les résultats étant très prometteurs, nous avons poursuivi l'expérimentation en vue d'obtenir un prototype industriel.

La seconde partie de ce chapitre est dédiée à l'implantation de notre démarche sur un matériel industriel offrant des performances moindres. Le niveau de qualité sur les images reconstruites est similaire à celui obtenu en simulation, mais les temps de calcul sont alors des facteurs limitants. Le respect du standard JPEG2000 à l'encodeur permet d'envisager le portage vers un encodeur matériel (*hardware*). Cela n'a pas été réalisé ici car il n'y a pas actuellement sur le marché d'encodeur intégrant la gestion de la ROI. Nous avons obtenu des résultats encourageants qui invitent à poursuivre l'étude.

Plusieurs développements sont toutefois envisageables et ouvrent la voie à d'intéressantes perspectives de travail.

Un premier développement serait la proposition d'un nouveau codec par "Motion JPEG2000 adapté" prenant en compte la ROI, représenté par les schémas des figures Fig. B.1 et Fig. B.2, fournies en annexe. Dans l'avenir, l'industriel MAGYS pourrait envisager de le breveter.

Il reste également à intégrer à notre prototype la transmission et la réception via des modems GSM. Sachant qu'on a un débit de 1 img/s en réseau GSM, on aurait théoriquement un débit de 25 img/s pour le réseau UMTS, soit la cadence vidéo temps-réel en Europe.

Toutefois, la reconstruction implicite du masque au décodeur doit être améliorée pour éviter l'artefact après la construction d'une image finale. Par exemple, un opérateur morphologique de type dilatation pourrait réduire de façon significative les artefacts.

Concernant l'obtention du masque de mouvement, on pourrait envisager l'utilisation de la détection de mouvement par l'approche probabiliste des champs aléatoires de Markov en *hardware*, ce qui permet d'obtenir non seulement un masque de mouvement plus précis, mais aussi d'envisager un fonctionnement en temps-réel. Dans cette perspective, on implanterait en parallèle : l'estimation de l'image de référence, la détection de mouvement et la transmission sur un noyau temps-réel [CG05] (voir annexe C).

L'amélioration du rendu visuel de l'image reconstruite par la transformée nonlinéaire logarithmique LUX est une autre piste d'investigation que nous avons étudiée (voir annexe A). La transformée non-linéaire LUX est gourmande en calcul, mais elle peut être réalisée avec un processeur spécifique, comme un DSP ("*Digital Signal Processor*"). Par contre, nous nous éloignerons de la norme JPEG2000, dans la mesure où les codecs JPEG2000 respectant le standard ne proposent que trois espaces couleurs linéaire RGB, YUV et YCrCb.

Bibliographie

- [1.000] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG2000 Part I Final Committee Draft Version 1.0. Mars, 2000.
- [Abu85] A. S. Abutaleb. Automatic thresholding of gray-level pictures using two-dimensional entropy. Computer Vision Graphics Image Processing, 29 :22–32, 1985.
- [AM05] P. M. Q. Aguiar and J. M. F. Moura. Figure-ground segmentation from occlusion. *IEEE Transactions on Image Processing*, 14(8):1109–1124, 2005.
- [Amb00] S. Ambellouis. Analyse du mouvement dans les séquences d'images par une méthode récursive de filtrage spatio-temporel sélectif. PhD thesis, Université de Lille 1 - Sciences et Technologies de Lille, France, 2000.
- [BA83] P. J. Burt and E. H. Adelson. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications*, 31 :532–540, 1983.
- [BBMP04] S. Bouchoux, E. Bourennane, J. Miteran, and M. Paindavoine. Implementation of JPEG2000 Arithmetic Decoder on a Dynamically Reconfigurable ATMEL FPGA. In *IEEE Computer Society Annual* Symposium on VLSI (ISVLSI 2004), pages 237–238, Lafayette (Louisiane), USA, February 19, 2004.
- [Bea03] B. Beaumesnil. Compression JPEG2000-Transformation couleur non-linéaire et Gestion de ROI par détection de mouvement. DEA, Université de Bordeaux 1, June, 2003.
- [Bes86] J. E. Besag. On the statistical analysis of dirty picture. J Royal Statist. Soc., 48 :259–302, 1986.
- [BPBSS97] J. Benois-Pineau, D. Barba, N. Sarris, and M. G. Strintzis. Video coding for wireless varying bit-rate communications based on area of

interest and region representation. In *International Conference on Image Processing*, volume 3, pages 555–558, Californie, USA, October 26-29, 1997.

- [BS02] A. P. Bradley and F. W. M. Stentiford. JPEG2000 and Region of Interest Coding. In DICTA2002 : Digital Image Computing Techniques and Applications, Melbourne, Australia, January 21-22, 2002.
- [BT.95] Recommandations UIT-R BT.500-7. Méthodologie d'évaluation subjective de la qualité des images de télévision. Technical report, UIT. 1995.
- [CB90] P. Chou and C. Brown. The theory and practice of Bayesian image labeling. *International Journal of Computer Vision*, 4:185–21, 1990.
- [CBC01] A. Caplier, L. Bonnaud, and J. M. Chassery. Robust fast extraction of video objects combining frame differences and adaptive reference image. In Proc. IEEE Int. Conf. Image Processing, pages 785–788, Thessaloniki, Greece, September, 2001.
- [CG05] F. Cottet and E. Grolleau. Systèmes Temps Réel de Contrôle-Commande Conception et implémentation. DUNOD, Paris, 2005.
- [Cha03] M. Chaumont. Représentation en objets vidéo pour un codage progressif et concurrentiel des séquences d'images. PhD thesis, Institut de Formation Supérieur en Informatique et Communication IFSIC, France, 2003.
- [CHH⁺04] S.-Y Chien, Y.-W Huang, B.-Y Hsieh, S.-Y Ma, and L.-G Chen. Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. *IEEE Transactions on Multimedia*, 6 (5) :732–748, 2004.
- [CK05] S. C. S. Cheung and C. Kamath. Robust background subtraction with foreground validation for urban traffic video. EURASIP Journal on Applied Signal Processing, 14(14) :2330–2340, 2005.
- [CSE00] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system : an overview. *IEEE Transactions on Consumer Electronics*, 46 :1103–1127, 2000.
- [Dau92] I. Daubechies. Ten Lectures on Wavelets. SIAM, 1992.
- [Dau98] I. Daubechies. Recent Results in Wavelet Applications. Journal of Electronic Imaging, 7(4) :2–9, 1998.

- [Dav95] F. Davoine. Compression d'images par fractales basé sur la triangulation de Delaunay. PhD thesis, Institut National Polytechnique de Grenoble - INPG, 1995.
- [Del05] C. Delgorge. Proposition et Evaluation de techniques de compression d'images ultrasonores dans le cadre d'une télé-échographie robotisée. PhD thesis, Univérsite d'Orléans, 2005.
- [DLC99] C. Dumontier, F. Luthon, and J. P. Charras. Real-time DSP implementation for MRF-based video motion detection. *IEEE Trans. on Image Processing*, 8(10) :1341–1347, October 1999.
- [Dra00] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG 2000 Part II Final Committee Draft. December, 2000.
- [DS96] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. J. Fourier Anal. Appl., 4 :247–269, 1996.
- [EHD99] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *IEEE ICCV99 Frame Rate Workshop*, Kerkyra, Greece, September, 1999.
- [Ema02] S. Emami. An error resilient JPEG2000 for wireless applications. Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th, 3 :1089 - 1090, 2002.
- [Fag00] J.-M. Fages. JPEG2000-Principe, Implémentation et évaluation Mémoire d'ingénieur CNAM. Paris, September, 2000.
- [FG03] D. Faura and P. Garda. Segmentation d'images couleurs pour la compression de séquences vidéo par l'algorithme Mask Motion JPEG2000. In CORESA'03, Janv., Lyon, France, 2003.
- [FR97] N. Friedman and S. Russell. Image Segmentation in Video Sequences : A Probabilistic Approach. In *Thirteenth Conf. on Uncertainty in Artificial Intelligence (UAI 97)*, pages 175–181, Providence, Rhode Island, USA, August 1-3, 1997.
- [FRL94] P. Fiche, V. Ricordel, and C. Labit. Etude d'algorithmes de quantification vectorielle arborescente pour la compression d'images fixes. *IRISA*, 1994.
- [GG84] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions* on Pattern Analysis and Machine Intelligence, 6:721–741, 1984.

- [Gro03] R. Grosbois. Image security and Processing in the JPEG2000 compressed domain. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2003.
- [GTCS⁺01] D. Gutchess, M. Trajković, E. Cohen-Solal, D. Lyons, and A. K. Jain. A background model initialization algorithm for video surveillance. In *Eighth IEEE International Conference on Computer Vision*, pages 733–740, July 7-17, Vancouver, BC, Canada, 2001.
- [HHD00] J. Haritaoglu, D. Harwood, and L. S. Davis. W 4 : real-time surveillance of people and their activitie. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8) :809–830, August 2000.
- [IBBD03] D. Izquierdo, J. Becerra, Y. Berthoumieu, and M. Donias. Segmentation multi-descripteurs de scènes autoroutières. In CORESA'03, Lyon, January 16-17, 2003.
- [Imp04] G. Impoco. JPEG2000 A Short Tutorial. Visual Computing Lab -ISTI-CNR Pisa, 2004.
- [Kak02] Implantation JPEG2000 Kakadu. http://www.kakadusoftware. com, 2002.
- [KCHD05] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3) :172–185, 2005.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598) :671–680, 1983.
- [KZB93] Z. Kato, J. Zerubia, and M. Berthod. Bayesian image classification using Markov Random Fields. In A. M. D. G. Demoments, editor, *Maximum Entropy and Bayesian Methods*, pages 375–382. Kluwer Academic Publisher, 1993.
- [LB04] F. Luthon and B. Beaumesnil. Color and R.O.I with JPEG2000 for wireless videosurveillance. In *IEEE Conference on Image Processing*, *ICIP'04*, volume 5, pages 3205–3208, Singapore, October, 2004.
- [LF03] F. Loras and J. Fournier. H.264/MPEG-4 AVC, un nouveau standard de compression vidéo. In CORESA'03, Lyon, January 16-17, 2003.
- [Lié98] M. Liévin. Analyse entropico-logarithmique de séquences vidéo couleur : Application à la segmentation et au suivi de visages parlants. PhD thesis, Institut National Polytechnique, Grenoble-INPG, France, 1998.

- [LLF04] F. Luthon, M. Liévin, and F. Faux. On the use of entropy power for threshold selection. Signal Processing, 84 :1789–1804, 2004.
- [LNR02] F. Luthon, X. Navarro, and J. Roch. Systèmes autonomes de vision et de transmission d'images sans fil pour la surveillance routière. Rapport de pré-étude. Contrat MAGYS-LIUPPA. Bayonne, October, 2002.
- [MAG] MAGYS. Technopole IZARBEL Bidart 64210, http://www.magsys. net.
- [Mal89] S. G. Mallat. A Theory for Multiresolution Signal Decomposition : The Wavelet Representation. *IEEE Transactions of Pattern Analysis* and Machine Intelligence, 2(7):674–693, July, 1989.
- [Mar96] B. Marcotegui. Segmentation de séquences d'images en vue du codage. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, France, 1996.
- [MPL⁺05] J. Meessen, C. Parisot, C. Lebarz, D. Nicholson, and J. F. Delaigle. Smart Encoding for Wireless Video Surveillance. In *Image and Vi*deo Communications and Processing (VCIP). SPIE Proc, San Jose, January 16-20, 2005.
- [Pat94] O. Patrouix. Modélisation multi-niveau de données ultrasonores : Application à la robotique mobile. PhD thesis, Université de Montpellier II, pages 60-69, 1994.
- [PMCM01] A. Prati, I. Mikic, R. Cucchiara, and M. M. Movadi. Comparative evaluation of moving shadow detection algorithms. In *Empirical Eva*luation Methods in Computer Vision, December, 2001.
- [PR01] G. Patané and M. Russo. The enhanced LBG algorithm. Neural Networks, 14(9) :1219–1237, November, 2001.
- [PS02] P. W. Power and J. A. Schoonees. Understanding background mixture models for foreground segmentation. In *Image and Vision Computing New Zealand*, pages 267–271, Auckland, November 26-28, 2002.
- [Rio93] O. Rioul. Ondelettes régulières : Application à la compression d'images fixes. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1993.
- [RJ02] M. Rabbani and R. Joshi. An overview of the JPEG2000 still image compression standard. Signal Processing : Image Communication, 17:3–48, 2002.

[KKJB00]	J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic back- ground model for tracking. In <i>Proceedings of the 6th European Confe-</i> <i>rence on Computer Vision-Part II</i> , pages 336–350. Springer-Verlag, London, June 26-July 01, 2000.
[SG99]	C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. <i>CVPR</i> , 2 :246–252, Fort Collins, June 23-25, 1999.
[Sha48]	Claude E. Shannon. A Mathematical Theory of Communication. <i>The Bell System Technical Journal</i> , 27:379–423,623–656, 1948.
[Sun05]	P. Sunna. AVC/H.264 Un système de codage vidéo évolué pour la HD et SD. UER -Revue Technique, Geneva, 2005.
[Swe95]	W. Sweldens. The Lifting Scheme : A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, <i>Wavelet Applications in Signal and Image Processing III</i> , pages 68–79. Proc. SPIE 2569, 1995.
[Swe98]	W. Sweldens. The Lifting Scheme : A construction of second generation wavelets. <i>SIAM Journal on Mathematical Analysis</i> , 29:511–546, 1998.
[SWS03]	R. Schafer, T. Wiegand, and H. Schwarz. The emerging H.264/AVC standard. <i>EBU, Technical Review</i> , Geneva, 2003.
[Tau00]	D. Taubman. High performance scalable image compression with EBCOT. <i>IEEE Transactions on Image Processing</i> , 9(7) :1151–1170, 2000.
[TM02]	D. S. Taubman and M. W. Marcellin. <i>JPEG2000 Image Compression fundamentals standard and practice</i> . Kluwer academic publishers, Netherlands, 2002.
[TOWS02]	D. Taubman, E. Ordentlich, M. Weinberger, and G. Seroussi. Embedded block coding in JPEG2000. <i>Signal Processing : Image Communication</i> , 17:49–72, 2002.
[VMP98]	P. Vannoorenberghe, C. Motamed, and JG. Postaire. Réactualisa- tion d'une image de référence pour la détection du mouvement dans les scènes urbaines. <i>Traitement du Signal</i> , 15 (2) :139–148, 1998.
[WA03]	S. Wu and A. Amin. Automatic thresholding of gray-level using multi-stage approach. <i>International Conference on Document Analysis and Recognition (ICDAR'03)</i> , 38:493–497, Edinburgh, August 3-6, 2003.

.

.

Гт

- [WB01] G. Welch and G. Bishop. An introduction to the Kalman filter. In *Siggraph*, California, August 12-17, 2001.
- [WS06] H. Wang and D. Suter. A novel robust statistical method for background initialization and visual surveillance. In Asian Conference on Computer Vision (ACCV), volume I, pages 328–337, India, January 13-16, 2006.
- [Yos04] T. Yoshida. Background differencing technique for image segmentation based on the status of reference pixels. In *International Confe*rence on Image Processing ICIP'04, pages 3487–3490, Singapore, October 24-27, 2004.
- [Zad65] L. A. Zadeh. Fuzzy sets. Information and Control, 8:338–353, 1965.

Publications de l'auteur

Congrès Internationaux

- T. Totozafiny, F. Luthon, and O. Patrouix. Pros and Cons of the Nonlinear LUX Color Transform for Wireless Transmission with Motion JPEG2000. In *Image and Vision Computing New Zealand (IVCNZ'06)*, pages 49-54, Great Barrier Island, November 27-29, 2006.
- T. Totozafiny, O. Patrouix, F. Luthon, and J-M. Coutellier. Dynamic Background Segmentation for Remote Reference Image Updating within Motion Detection JPEG2000, In *IEEE International Symposium on Industrial Electronics (ISIE2006)*, volume 1, pages 505-510, Montreal, July 9-13, 2006.
- T. Totozafiny, O. Patrouix, F. Luthon, and J-M. Coutellier. Motion Reference Image JPEG2000 : road surveillance application with wireless device. In Visual Communications and Image Processing (VCIP'05), Proceedings of the SPIE, volume 5960, pages 1839-1848, Beijing, July 12-15, 2005.

A. Amélioration du rendu visuel par LUX

Une autre transformée couleur permet d'obtenir, en comparaison avec les deux modèles utilisés dans la norme JPEG2000 ($RGB \rightarrow YUV$ et $RGB \rightarrow YCrCb$), une bonne restitution des couleurs de l'image compressée ; c'est la transformée logarithmique non-linéaire LUX (Logarithmic hUe eXtension) [Lié98]. On note que cette transformée n'est pas incluse dans le standard. Le modèle logarithmique a pour origine d'introduire une non-linéarité, et ce non seulement pour être en adéquation avec le système visuel humain, mais aussi pour permettre une description efficace des teintes tout en s'affranchissant des problèmes de bruit et des variations d'éclairage.

Dans une application routière, [LB04] a montré que la transformée couleur LUX permet d'améliorer le rendu visuel d'image encodée par JPEG2000. Obtenir un meilleur rendu visuel dans la région d'intérêt est un gain complémentaire en plus de l'encodage par la gestion de ROI.

Il existe plusieurs versions de l'espace couleur LUX. Son utilisation dépend de l'application visée. L'équation qui permet de passer de l'espace couleur linéaire RGB à l'espace LUX est :

$$L = (R+1)^{t_{11}}(G+1)^{t_{12}}(B+1)^{t_{13}} - 1$$
 (A.1)

$$X = (R+1)^{t_{21}}(G+1)^{t_{22}}(B+1)^{t_{23}} - 1$$
(A.2)

$$U = (R+1)^{t_{31}}(G+1)^{t_{32}}(B+1)^{t_{33}} - 1$$
 (A.3)

et son inverse est donnée par :

$$R = (L+1)^{a_{11}}(X+1)^{a_{12}}(U+1)^{a_{13}} - 1$$
(A.4)

$$G = (L+1)^{a_{21}}(X+1)^{a_{22}}(U+1)^{a_{23}} - 1$$
 (A.5)

$$B = (L+1)^{a_{31}} (X+1)^{a_{32}} (U+1)^{a_{33}} - 1$$
 (A.6)

Тав. А.1.	: Temps	de	$\operatorname{compression}$:	transformée	LUX	et	standard	avec	la	sé-
	quence	LA	PS.								

Transformée couleur	Temps de compression (ms) PC AMD 1.6 GHz
Standard $(RGB \rightarrow YCrCb)$	180
$LUX(RGB \rightarrow LUX)$	600

où (t_{ij}) et (a_{ij}) sont respectivement les éléments de la matrice T et A (cf. Eq. 1.42).

Ce sont des transformées que nous allons étudier. Pour cela à la place de la conversion standard $RGB \rightarrow YCrCb$, on utilise $RGB \rightarrow LUX$. La Fig. A.1 montre l'exemple d'un résultat obtenu par l'espace couleur LUX. On constate qu'on a une meilleure restitution couleur (feu arrière gauche). On a ainsi un gain de l'ordre de 1 à 4 dB (Fig. A.1-e). Le gain est obtenu au détriment du temps de compression. En effet, le temps de la compression avec la conversion $RGB \rightarrow LUX$ est presque quatre fois celui de la transformée standard $RGB \rightarrow YCrCb$ (voir Tab. A.1). C'est la transformée LUX qui est gourmande en calculs. Ceci résulte de l'opération avec l'exponentielle.

De ce fait, on pourra estimer discutable le gain qualitatif obtenu car, d'un côté il y a amélioration visuelle du rendu couleur, mais de l'autre côté le temps de calcul reste très élevé, et cela d'autant plus que LUX n'est pas implanté dans le standard, ce qui compromettra le travail de tout utilisateur d'un codec JPEG2000 en *hardware*. Dans ce contexte industriel, contraint où nous sommes placé, nous avons donc décider de ne pas investiguer plus en avant l'usage de cette transformée couleur.



(a) Image finale reconstruite par transformée lastandard irréversible

(b) Image finale reconstruite par la transformée LUX





(d) Zoom sur b)

(c) Zoom sur a)



FIG. A.1.: Amélioration par la transformée non-linéaire logarithmique LUX; le feu arrière est d'un rouge plus vif.

B. Schémas détaillés

B.1. Encodeur



FIG. B.1.: Schéma détaillé de l'encodeur.

B.2. Décodeur



FIG. B.2.: Schéma détaillé du décodeur.

C. Architecture de l'implantation de l'algorithme

L'implantation du codeur nécessite trois processus en parallèle : estimation d'une référence, encodage et transmission. Le premier concerne l'acquisition vidéo et l'estimation d'une image de référence. La deuxième permet l'extraction de la région d'intérêt (la segmentation) ainsi que l'encodage JPEG2000. Comme nous sommes dans le cadre d'un système embarqué, pour l'implantion logicielle, on peut utiliser les méthodes de conception pour les systèmes temps-réel : SA-RT ("Structured Analysis for Real Time Systems") et DARTS ("Design Approach for Real-Time Systems") qui sont développées dans [CG05]. La Fig. C.1 montre les processus à réaliser.



FIG. C.1.: Diagramme des processus pour implantation de notre algorithme en temps-réel.

D. JPEG2000 : transmissions progressives

Le standard JPEG2000 permet la transmission progressive. Pour cela de nombreuses options sont disponibles. La transmission progressive consiste à afficher la partie d'image qu'on souhaite voir en premier. La réalisation de celle-ci se fait au moment du codage. Comme les données dans le codestream sont regroupées dans différents paquets puis insérées dans différentes couches de qualité, l'ordre d'insertion des paquets dans le codestream est défini ainsi que l'ordre des coefficients à décoder par le décodeur. Un marqueur segment concernant celui-ci dans le codestream indique l'option choisie pendant l'encodage. Le contenu de ce marqueur est la composition de 4 variables : résolution R, composante C (couleur), niveau L et position P. Dans la première partie de la norme, 5 options sont disponibles. Elles sont réalisées en fonction de l'application visée.

- Progression LRCP : couche-résolution-composante-position, utilisée pour la recherche d'une image dans une base de données;
- Progression RLCP : résolution-couche-composante-position pour une application client-serveur ; lorsque le client demande une résolution différente de celle du serveur (Fig. D.1-a) ;
- Progression RPCL : résolution-position-composante-couche, permettant la réalisation d'une scalabilité dans la sous-bande;
- Progression PCRL : position-composante-résolution-couche, permettant une progression par qualité (Fig. D.1-b);
- Progression CPRL : composante-position-résolution-couche, permettant d'obtenir une haute qualité pour une région spécifiée dans l'image selon une composante choisie.



Réception complète

(a) Résolutions



Décodage à t=300

Décodage à t=150

Décodage à t=60

(b) Qualité

FIG. D.1.: Transmissions progressives : résolutions et qualité.

E. Implantations JPEG2000

E.1. Logicielle

On peut citer les versions suivantes :

- Jasper, développé en C mais qui ne code pas la région d'intérêt. C'est un open source. On peut le télécharger gratuitement sur internet (http://www.ece.uvic.ca/~mdadams/jasper). Le Tab. E.1 montre les surcoûts (mémoires, calculs, etc) de JPEG2000 par rapport à JPEG, mesures réalisés avec le Jasper.
- JJ2000, développé en Java mais très lent. C'est un open source. On peut aussi le télécharger gratuitement sur internet (http://jj2000.epfl.ch).
- LuraWave, développé en C avec un SDK par la société LuraTech et facilement trouvable dans le commerce.
- Kakadu, développé en C++ et assembleur par D. Taubman et M. Marcellin [TM02] et facilement trouvable dans le commerce. C'est une version qui permet l'encodage ROI. C'est la plus rapide par rapport aux versions mentionnées ci-dessus.

Dans cette thèse, nous avons utilisé la version Kakadu.

E.2. Matérielle

Actuellement dans le marché, de nombreux codecs JPEG2000 en *hardware* sont disponibles. Citons par exemple :

Taille mémoire	x 40
Accès en mémoire	x 1.9
Temps d'exécution du codage	x 34
Temps d'exécution du décodage	x 8

TAB. E.1.: Surcoûts de JPEG2000 par rapport à JPEG [LNR02].

- BA111JPEG2000E, développé avec un DSP spécifique, est un encodeur en hardware pour le FPGA Xilinx XC2V3000-6.
- CTR-1471, disponible en format PC104, permet l'encodage en temps-réel (40 Mbits/s). On peut encoder en une seconde une image dont la taille est de 40 Mbits.