

Université de Sherbrooke
Faculté des Sciences appliquées
Département de génie électrique et d'informatique

**ENCODAGES ET DÉCODAGES NUMÉRIQUES
DES SIGNAUX VIDÉO COMPOSÉS NTSC ET PAL**

Mémoire de maîtrise
Spécialité: génie électrique

Abdeljalil Hammami

Sherbrooke, Québec
Canada

Août 94



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-21768-X

Dédicace

À mes parents, à mes frères, à mes soeurs et à mes amis.

SOMMAIRE

Cette étude décrit des approches d'encodage et de décodage des signaux composites PAL et NTSC. Ces approches ont pour objet de diminuer les deux artefacts majeurs dus à l'interférence entre les signaux de luminance et de chrominance connus sous les appellations, luminance croisée et diachromie ou encore respectivement en anglais " cross -color " et "cross-luminance " tout en préservant une résolution acceptable des images.

Les caractéristiques spectrales de ces signaux sont décrites afin de comprendre la procédure de design des filtres mono- et multidimensionnels, séparables et non séparables qui seront utilisés dans ces approches.

La possibilité de supprimer ces artefacts sans aucun post-traitement à l'encodeur est aussi examinée par le développement d'un décodeur adaptatif selon la détection de mouvement.

Finalement les résultats de simulations sont discutés.

REMERCIEMENT

Je profite de ces lignes pour souligner l'apport important des personnes qui m'ont appuyé dans l'accomplissement de ce travail durant toute la période de mes recherches.

Je tiens à remercier, mon directeur de recherche, Monsieur Chon Tam Ledinh, pour sa grande disponibilité et les précieux conseils prodigués tout au long de ce travail.

J'aimerais aussi remercier la compagnie Miranda Inc, qui a eu l'initiative de ce projet, d'avoir soutenu cette étude.

Je tiens aussi à remercier le Centre de recherches sur les communications du Canada pour avoir mis leurs équipements à notre disposition pour réaliser nos tests

Je remercie aussi le gouvernement tunisien qui, par le biais de son aide financière, m'a aidé à satisfaire mes ambitions.

TABLE DES MATIÈRES

DÉDICACE	ii
SOMMAIRE	iii
REMERCIEMENT	iv
TABLE DES MATIÈRES	v
LISTE DES FIGURES	viii
CHAPITRE 1	
Introduction	1
CHAPITRE 2	
Révision des techniques d'encodage et de décodage des signaux vidéos composés NTSC et PAL	4
2.1. Principaux signaux constituant un signal vidéo couleur.....	4
2.1.1. Le signal de luminance	4
2.1.2. Les signaux de chrominance	5
2.2. Principe d'encodage des différents systèmes vidéo couleurs.....	6
2.2.1 Encodage NTSC.....	6
2.2.1.1. Signal composite NTSC.....	8
2.2.1.2. Spectre de Fourier unidimensionnel	9
2.2.1.3. Spectre de Fourier multidimensionnel	10
2.2.2 Encodage PAL.....	14
2.2.2.1. Signal composite PAL.....	15
2.2.2.2. Spectre de Fourier unidimensionnel.....	16
2.2.2.3. Spectre de Fourier multidimensionnel.....	18
2.3. Décodage conventionnel des signaux NTSC et PAL.....	20
2.3.1. Décodage conventionnel NTSC.....	20
2.3.2. Décodage PAL.....	20
2.4. Diachromie et luminance croisée.....	22
2.4. 1. La diachromie.....	22
2.4. 2. La luminance croisée	23
2.5. Encodage et décodage améliorés.....	25
2.5. 1. Encodage et décodage bidimensionnels spatiaux	25
2.5.1. 1. Encodage bidimensionnel	25
2.5.1.2. Décodage amélioré	26
2.5. 2. Décodage adaptatif.....	26
2.5.2.1. Approche des connections séries.....	28
2.5.2.2 Approche adaptative selon le mouvement.....	28
2.5.2.3. Approche adaptative selon le mouvement et transition verticale.....	29
2.5. 3. Codage et décodage améliorés tridimensionnels.....	29
2.5.3.1. Codage et décodage tridimensionnels non séparables.....	29
2.5.3.2. Codage et décodage tridimensionnels séparables	30

CHAPITRE 3

Encodage numérique des signaux PAL et NTSC	31
3.1. Format D1 ou Format 4:2:2.....	32
3.2. Format D2 ou Format 4fsc.....	32
3.3. Encodage numérique NTSC	33
3.3.1. Conversion du format D1 en format D2.....	33
3.3.1.1. Conversion de fréquence des composantes de chrominance.....	33
3.3.1.2. Conversion de fréquence des composantes de luminance.....	33
3.3.1. Filtres utilisés	34
3.3.1.4. Formation du signal NTSC composite.....	35
3.3.2. Encodage numérique proposé	36
3.3.2.1. Encodage avec traitement unidimensionnel.....	38
3.3.2.2. Encodage avec traitement bidimensionnel	39
3.3.2.3. Encodage avec traitement tridimensionnel	39
3.4. Encodage PAL.....	40
3.4.1. Conversion du format D1 en format D2.....	40
3.4.1.1. Conversion de format pour les composantes de chrominance	40
Cr et Cb.....	40
3.4.1.2. Conversion de fréquence de la luminance	40
3.4.2. Conception de filtres	41
3.4.3. Formation du signal PAL composite encodage standard ou 0D.....	42
3.4.4. Configurations d'encodage proposées.....	44

CHAPITRE 4

Décodage numérique des signaux PAL et NTSC	
Approche fixe et compatible	46
4.1. Décodage NTSC.....	46
4.1.1. Décodage monodimensionnel ou 1D.....	47
4.1.2. Décodage bidimensionnel ou 2D	48
4.1.3. Décodage tridimensionnel ou 3D.....	48
4.1.4. Décodage multidimensionnel sélectif.....	48
4.1.5. Conception des filtres.....	50
4.1.5.1 Les signaux NTSC échantillonnés à 4fsc.....	51
4.2. Décodage PAL	59
4.2.1. Signal PAL composite échantillonné à 4fsc.....	60
4.2.2. Conception des filtres.....	60
4.3. Limitations de l'approche du décodage sélectif.....	65

CHAPITRE 5

Décodage numérique: Approche adaptative Système NTSC	66
5.1. Principe de l'approche.....	67
5.1.1. Module de détection de mouvement.....	69
5.1.2. Détecteur des transitions verticales	70
5.1.3. Filtre en losange	71
5.1.4. Principe de fonctionnement	73

CHAPITRE 6

Résultats des simulations et interprétations	75
6.1. Approches fixes et compatibles	76
6.1.1. Système NTSC	76
6.1.1.1. Configuration 1D-1D	76

6.1.1.2. Configuration 2D-2D	77
6.1.1.3. Configuration 3D-3D	77
6.1.2. Système PAL.....	78
6.1.2.1. Configuration 1D-1D	78
6.1.2.2. Configuration 2D-2D	79
6.1.1.3. Configuration 3D-3D	79
6.2. Approche adaptative.....	80
Conclusion.....	82
Annexe.....	84

LISTE DES FIGURES

Figure 2.1.	Schéma synoptique d'un encodeur NTSC.....	7
Figure 2.2.	Limitation des bandes des signaux I et Q.....	7
Figure 2.3.	Spectre au voisinage de 3.58 Mhz. Système NTSC	12
Figure 2.4.	Densité du spectre vidéo. Système NTSC.....	12
Figure 2.5.	Spectre à trois dimensions du signal NTSC	13
Figure 2.6.	Encodeur PAL.....	15
Figure 2.7.	Structure luminance chrominance en PAL	17
Figure 2.8.	Densité du spectre vidéo.....	17
Figure 2.9.	Spectre à trois dimensions du signal PAL.....	19
Figure 2.10.	Schéma synoptique d'un décodeur NTSC.....	21
Figure 2.11.	Schéma synoptique d'un décodeur PAL.....	21
Figure 2.12.	Encodeur amélioré. Système NTSC.....	26
Figure 3.1.	Schéma bloc de la conversion fréquentielle. Système NTSC.....	34
Figure 3.2.	Encodage standard.....	36
Figure 3.3.	Encodeur NTSC standard OD	37
Figure 3.4.	Configuration 1D	38
Figure 3.5.	Configuration 1D avec des filtres complémentaires	38
Figure 4.5.	Configuration 2D	39
Figure 4.6.	Configuration 3D	39
Figure 3.7.	Schéma bloc de la conversion fréquentielle. Système PAL.....	41
Figure 3.8.	Encodage standard.....	42
Figure 3.9.	Encodeur standard PAL OD.....	43
Figure 4.1.	Structure du décodage monodimensionnel.....	49
Figure 4.2.	Structure du décodage bidimensionnel.....	49
Figure 4.3.	Structure du décodage tridimensionnel.....	49

Figure 4.4.	Configuration globale du décodage NTSC sélectif.....	50
Figure 4.5.	Structure bidimensionnelle en treillis du signal NTSC échantillonné	53
Figure 4.6.	Structure bidimensionnelle (VT) en treillis du signal NTSC échantillonné	54
Figure 4.7.	Support de la réponse impulsionnelle	53
Figure 4.8.	Réponse impulsionnelle du filtre.....	53
Figure 4.9.	Réponse en fréquence du filtre : caractéristique de chrominance.....	57
Figure 4.10.	Contour de la réponse en fréquence.....	57
Figure 4.11.	Réponse impulsionnelle du filtre non-causal.....	58
Figure 4.12.	Décodeur sélectif PAL	59
Figure 4.13.	Structure spatiale du signal PAL échantillonné à 4fsc.....	60
Figure 4.14.	Structure bidimensionnelle du signal NTSC échantillonné à 4fsc	61
Figure 4.15.	Support de la réponse impulsionnelle	61
Figure 4.16.	Réponse impulsionnelle du filtre.....	63
Figure 4.17.	Réponse impulsionnelle du filtre : version simplifiée.....	63
Figure 4.18.	Réponse en fréquence du filtre: Caractéristique de chrominance.....	64
Figure 4.19.	Contour de la réponse en fréquence.....	64
Figure 5.1.	Configuration globale de l'approche adaptative.....	68
Figure 5.2.	Détecteur de mouvement.....	69
Figure 5.3.	Détecteur de transition. Schéma bloc	70
Figure 5.4.	Réponse en fréquence du filtre : Caractéristique de chrominance.....	70

Figure 5.5. Contour de la réponse en fréquence.....70

CHAPITRE 1

INTRODUCTION

Le PAL et le NTSC ont été conçus pour la simple tâche d'amener l'image du transmetteur au spectateur. Avec l'apparition de la télévision couleur dans les années 50, ces systèmes portaient désormais les signaux différences de couleurs comme une paire de sous-porteuses modulées dans la partie des hautes fréquences du spectre de la luminance.

Ces systèmes forçaient ainsi les composantes de luminance et chrominance à partager la même bande de fréquence. Cela permettait non seulement de transmettre l'information couleur dans les mêmes limites de la largeur de bande allouée aux signaux monochromes, mais aussi et surtout d'avoir un degré de compatibilité avec le moniteur monochrome. En fait les signaux composites transmis sont aussi bien exploitables par les moniteurs monochromes que par les moniteurs couleurs.

La tâche principale d'un décodeur consistait alors, à séparer les composantes de luminance et de chrominance et à démoduler la chrominance afin de reproduire les signaux différences de couleurs pour enfin former l'image sur l'écran.

Puisque la luminance et la chrominance partagent la même bande de fréquence, il était par conséquent très difficile de les séparer sans qu'il y ait des effets d'interférence indésirables. En effet lors de l'opération de décodage, quelques signaux de luminance sont incorrectement décodés comme étant des signaux de chrominance et des signaux de chrominance sont interprétés comme des signaux de luminance. Cela introduit des artefacts connus respectivement sous les noms de diachromie et de luminance croisée.

chapitre 1

Dans le passé, ces phénomènes n'étaient pas très visibles à cause de la résolution limitée des moniteurs et surtout à cause des caméras utilisées.

Récemment, à cause de la haute qualité exigée de l'image, ainsi que de la très bonne qualité des caméras TV, ces défauts, causés par la séparation médiocre des deux spectres de luminance et de chrominance, sont devenus très gênants et sont désormais considérés comme un problème à résoudre.

Notre étude s'inscrit dans la foulée des travaux de recherche visant à résoudre ces deux problèmes. Elle est effectuée dans le cadre d'un projet proposé par la compagnie MIRANDA TECHNOLOGIE INC.

Le problème d'encodage et de décodage des signaux PAL ou NTSC, a été, pour très longtemps, considéré comme un problème unidimensionnel. Les récentes améliorations théoriques réalisées dans le traitement multidimensionnel ont rendu possible l'étude et l'interprétation des signaux TV composite PAL et NTSC en trois dimensions. Ces interprétations 3-D ont clarifié les différentes causes qui dégradent la qualité de l'image par ces défauts de diachromie et de luminance croisée. Ainsi le chapitre 2 sera consacré à une vue globale à tous les niveaux. Ce survol permettra de comprendre les principes de formation des signaux PAL et NTSC ainsi que leurs caractéristiques spectrales à trois dimensions. Il permettra aussi de comprendre les deux artefacts ci-dessus mentionnés. Le reste du chapitre sera consacré à énumérer les différentes solutions proposées jusqu'à maintenant.

Dans le chapitre 3, consacré à l'encodage, on proposera trois techniques d'encodage multidimensionnelles pour les deux systèmes PAL et NTSC. Ces techniques ne définissent pas un nouveau standard d'encodage mais contribuent uniquement à l'amélioration de la qualité du signal composite délivré par l'encodeur.

chapitre 1

Dans le chapitre 4, on proposera aussi trois techniques de décodage multidimensionnelles basées sur la règle de compatibilité entre la structure du codeur et celle du décodeur.

Dans le chapitre 5, on proposera une technique de décodage adaptatif multidimensionnel, qui indépendamment de la manière dont le signal composite a été encodé, est capable de séparer les deux signaux de luminance et de chrominance quasi-parfaitement tout en préservant la pleine résolution de l'image.

Le chapitre six sera consacré à l'interprétation des différents résultats obtenus par ces différentes approches.

CHAPITRE 2

RÉVISION DES TECHNIQUES D'ENCODAGE ET DE DÉCODAGE DES SIGNAUX VIDÉOS COMPOSÉS NTSC ET PAL

Dans ce chapitre, on étudiera en premier les principaux signaux constituant un signal vidéo couleur. Ensuite, on expliquera les principes d'encodage et de décodage conventionnel des différents systèmes vidéo couleur et les principaux artefacts qu'engendre ce genre d'encodage-décodage. Enfin, on explorera les principales approches proposées jusqu'à présent, afin de remédier à ces défauts à savoir l'encodage et le décodage, bidimensionnels, adaptatifs et tridimensionnels.

2.1. Principaux signaux constituant un signal vidéo couleur

Les signaux constituant un signal vidéo couleur sont de deux sortes :

- le signal de luminance.
- les signaux de chrominance.

2. 1.1. Le signal de luminance

Désigné généralement par le symbole EY . L'équation fondamentale du signal de luminance contenant toute l'information nécessaire à la production de l'image en noir et blanc est:

$$EY = 0.3ER + 0.59EG + 0.11EB \quad (2.1)$$

Le signal vidéo -luminance définit la *finesse d'image* [3]. Le pouvoir de résolution de l'oeil est grand vis-à-vis du signal de luminance. Il n'en est pas de même des informations de chrominance. Le signal de luminance, compte tenu des informations de

Chapitre 2

chrominance à y ajouter, devra donc occuper *la plus large bande passante, en fréquences, possible* [4].

2.1.2. Les signaux de chrominance

Les signaux de chrominance sont, ER, EG et EB pour le rouge, vert et bleu respectivement. En réalité, on n'a retenu que deux informations de couleur : (ER - EY) et (EB - EY).

On peut définir la valeur des signaux de différence de couleurs comme suit,

$$(ER - EY) = 0.7ER - 0.59EG - 0.11EB \quad (2.2)$$

$$(EB - EY) = 0.89ER - 0.30EG - 0.59EB \quad (2.3)$$

Les équations 2-1, 2-2 et 2-3 constituent donc la conversion (ou le matriçage) des trois composantes ER, EB et EG en signaux de luminance et de différence de couleurs (ER - EY) et (EB-EY).

Les signaux de différence de couleurs ne contiennent pas, en général, d'informations à haute fréquence [1]. On les limite souvent à environ 3 MHz et à 1.2 MHz respectivement pour les signaux de qualité studio et pour ceux de transmission.

Nous allons étudier, dans les paragraphes suivants, les divers principes de codage des systèmes NTSC et PAL ainsi que leurs caractéristiques spectrales.

À noter tout de suite, que la différence entre les deux systèmes d'encodage vidéo couleur cités réside uniquement dans le mode de composition des signaux de différence de couleurs (EB- EY) et (ER - EY) et dans le mode de modulation employé de la sous-porteuse couleur par le signal composé de chrominance.

Ce mode de composition, d'une part des signaux de différences de couleurs et d'autre part de modulation de la sous-porteuse couleur par le signal composé de chrominance

Chapitre 2

constitue à l'émission l'encodage des signaux de chrominance et à la réception le décodage de ces mêmes signaux.

2.2. Principe d'encodage des différents systèmes vidéo couleurs

2.2.1 Encodage NTSC

Ce système a été dans l'ordre chronologique, le premier système compatible existant.

Le schéma synoptique complet d'un codeur NTSC est donné par la figure 2.1. On remarque tout de suite deux nouveaux signaux Q et I. Les deux lettres Q et I signifient respectivement en quadrature et en phase. Ces deux signaux sont employés en NTSC au lieu des signaux (EB - EY) et (ER - EY). Ils sont déphasés de 33° par rapport à ces derniers. Ils s'obtiennent par un simple matriçage des signaux différences de couleurs, selon les deux équations :

$$I = 0.27(EB - EY) + 0.74(ER - EY) \quad (2-4)$$

$$Q = 0.41(EB - EY) + 0.48(ER - EY) \quad (2-5)$$

Initialement, le signal I occupait une bande plus large que celle occupée par le signal Q. (figure 2.2). À noter que la sous -porteuse chrominance étant générée à une fréquence de 3.58 MHz de la porteuse vision et le son est situé à 4.5 MHz de la porteuse vision, la bande latérale supérieure des axes Q et I a été limitée à +600 Khz afin de ne pas interférer avec les signaux de la porteuse son.

Le principe est le suivant :

Les signaux de chrominance I et Q sont modulés simultanément en QAM (modulation d'amplitude en quadrature) par la sous-porteuse couleur à 3.58 MHz. Le signal de chrominance composant I est appliqué à un premier modulateur symétrique

Chapitre 2

double, il en est de même pour le signal Q qui est appliqué à un deuxième modulateur symétrique double.

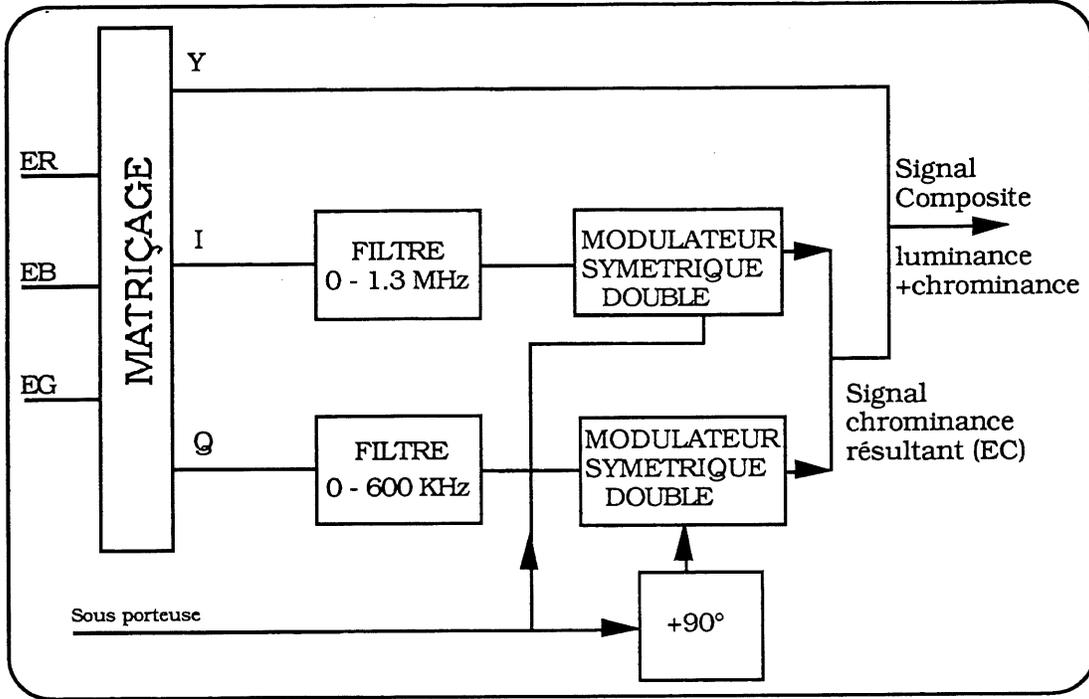


FIG 2.1. Schéma synoptique d'un encodeur NTSC

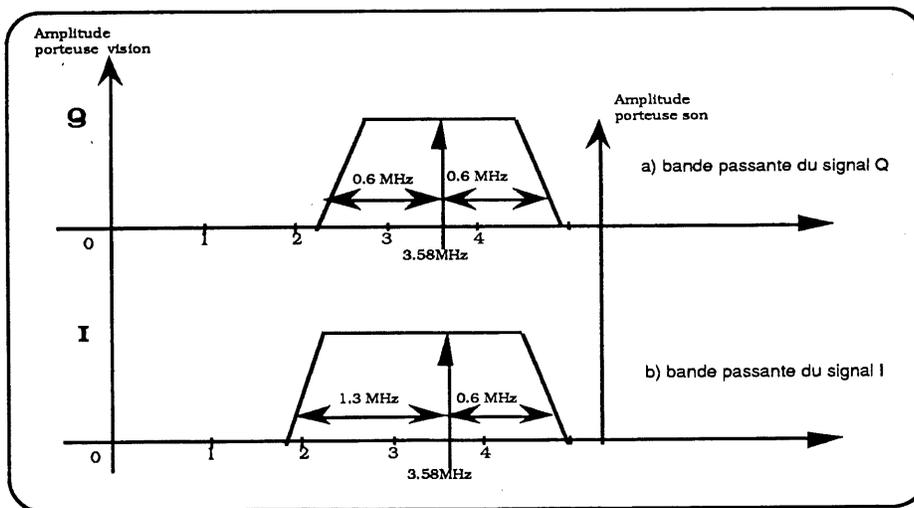


FIG 2.2. Limitation des bandes des signaux I et Q [4]

Chapitre 2

Le signal I et le signal Q modulent le signal de sous-porteuse dans chaque modulateur correspondant. Le signal de sous-porteuse est directement appliqué au modulateur symétrique double recevant le signal de chrominance composant I, par contre, il est déphasé de 90° avant d'être appliqué au modulateur symétrique double recevant le signal de chrominance composant Q. À la sortie des deux modulateurs symétriques doubles, on fait la somme des deux signaux de chrominance I et Q. On obtient alors le signal résultant de chrominance EC que l'on additionne avec le signal de luminance pour former le signal composite NTSC. Les deux filtres passe bas servent à limiter les bandes des signaux I et Q.

2.2.1.1. Signal composite NTSC

- Constitution:

Le signal composite est formé alors par la somme:

- du signal de luminance Y de 4.2 MHz de bande passante.
- du signal de chrominance modulé, porté par une sous-porteuse de chrominance que l'on désigne par fsc.

- Fréquence de ligne fh et de sous-porteuse fsc [3]

La donnée de base considérée est l'écart entre les porteuses image et son, normalisée à 4.5 MHz, celle-ci est un multiple de fh, est la solution la plus proche des normes noir et blanc à 525 lignes (soit fh = 15750 MHz) est obtenue pour :

$$fh = \frac{4500}{286} \cdot 10^3 = 15734,264Hz \quad (2-6)$$

La tolérance de stabilité est par ailleurs améliorée, ce qui permet un fonctionnement non synchronisé sur la fréquence du secteur.

Chapitre 2

La fréquence de trame devient

$$f_t = \frac{2fh}{525} = 59.94Hz \quad (2-7)$$

Pour la sous-porteuse de chrominance en décalage de ligne, la solution retenue est :

$$f_{sc} = fh \frac{455}{2} = 3.579545MHz \quad (2-8)$$

Le décalage de ligne a pour conséquence que *quatre trames*, appelées encore en anglais «*color frame*», sont nécessaires pour la coïncidence de phase de la sous-porteuse avec l'instant de synchronisation ligne.

2.2.1.2. Spectre de Fourier unidimensionnel

Pour le signal composite, qu'on peut écrire sous la forme [3]:

$$E_c = Y(t) + I(t) \cos(2\pi f_{sc}t + \Phi) + Q(t) \sin(2\pi f_{sc}t + \Phi) \quad (2-9)$$

le spectre de Fourier est la somme des spectres du signal de luminance et du signal de chrominance modulée.

Ces deux signaux ont une périodicité de base aux fréquences de ligne, f_h , et de trame, f_v , avec une structure de raies à deux niveaux $\pm m f_h$ et $\pm n f_v$. Les bandes latérales de modulation d'amplitude ont la même structure de part et d'autre de la porteuse.

La position en décalage de ligne de la sous porteuse f_{sc} conduit au décalage de ligne de toutes les raies $m f_h$ des composantes de chrominance, avec, en structure fine, l'imbrication des raies de deuxième niveau ($\pm n f_t$) des deux spectres, luminance et chrominance. (Figure 2.3).

Le spectre de luminance a ses raies primaires d'amplitudes maximales aux fréquences bases et cette amplitude, lissée, peut servir de référence.

Chapitre 2

Compte tenu des amplitudes choisies pour les signaux de chrominance I et Q, et en valeur statistique, on admet que les composantes des spectres de chrominance, autour de la porteuse f_{sc} , sont environ de -12 dB par rapport à la référence de luminance.

La décroissance des amplitudes du spectre de chrominance avec l'écart de fréquence, est, au début, sensiblement celle du spectre de luminance. Puis, l'effet du filtrage progressif se manifeste pour limiter les spectres des signaux I et Q. (figure 2.4)

2.2.1.3. Spectre de Fourier multidimensionnel [7]

En tenant compte de la relation entre la fréquence de la sous-porteuse et des taux de répétition des lignes et de trames, l'expression en trois dimensions du signal composite NTSC pour une structure entrelacée est donnée par :

$$\begin{aligned} E_c(x_1, x_2, t) = & Y(x_1, x_2, t) + I(x_1, x_2, t) \\ & \cdot \cos(2\pi f_{sc}' \cdot x_1 + \Phi) \cos \pi(f_T t - f_l' x_2) \\ & Q(x_1, x_2, t) \cdot \sin(2\pi f_{sc}' \cdot x_1 + \Phi) \cos \pi(f_T t - f_l' x_2) \end{aligned} \quad (2-10)$$

- Où
- x_1 = dimension spatiale horizontale
 - x_2 = dimension spatiale verticale
 - f'_{sc} = fréquence spatiale associée à la sous-porteuse
 - f_l = $1/T_l$, T_l la distance verticale entre deux lignes adjacentes
 - f_T = fréquence de trame (30 Hz) ($f_t = 2 \cdot f_T$)

Si S_y , S_i et S_q désignent respectivement les spectres de puissance du signal de luminance et des signaux de chrominance, le spectre de puissance du signal composite NTSC est donné par :

$$\begin{aligned} S_{Ec}(f_1, f_2, f_3) = & S_y(f_1, f_2, f_3) - 1/4 [S_c(f_1 - f_{sc}', f_2 - f_l' / 2, f_3 + f_T / 2) \\ & + S_c(f_1 + f_{sc}', f_2 - f_l' / 2, f_3 + f_T / 2) \\ & + S_c(f_1 - f_{sc}', f_2 + f_l' / 2, f_3 - f_T / 2) \\ & + S_c(f_1 + f_{sc}', f_2 + f_l' / 2, f_3 - f_T / 2)] \end{aligned} \quad (2-11)$$

Chapitre 2

Où: $S_c = S_i + S_q$
 $f_1 =$ fréquence spatiale horizontale.
 $f_2 =$ fréquence spatiale verticale.
 $f_3 =$ fréquence temporelle.

Le spectre tridimensionnel peut être vu comme une densité variable dans l'espace $f_1 f_2 f_3$. Le spectre du signal de luminance est centré autour de $(f_1, f_2, f_3) = (0, 0, 0)$ et le spectre du signal de chrominance est centré autour de $(f_1, f_2, f_3) = (\pm f'_{sc}, f'_{l/2}, -f_T/2)$ et autour de $(\pm f'_{sc}, -f'_{l/2}, f_T/2)$.

Le spectre spatio-temporel du codage NTSC montre quelques particularités importantes dans la position des volumes de chrominance au milieu des volumes de luminance.

Les volumes de chrominance ont une situation optimale dans l'espace pour éviter le mélange des spectres de luminance et de chrominance, tout en donnant le maximum de place au volume de chrominance.

La séparation, dans l'espace, des volumes de luminance et de chrominance permet de réaliser théoriquement la séparation des signaux sans perte d'informations notable.

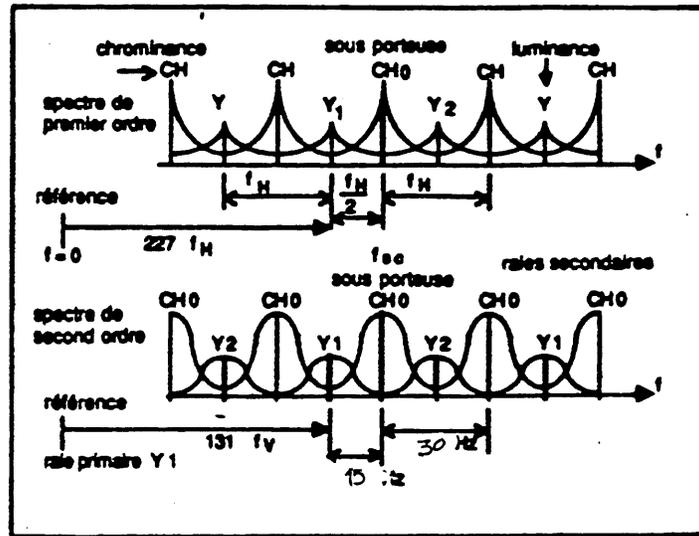


FIG 2.3. Spectre au voisinage de 3.58. Système NTSC [3]

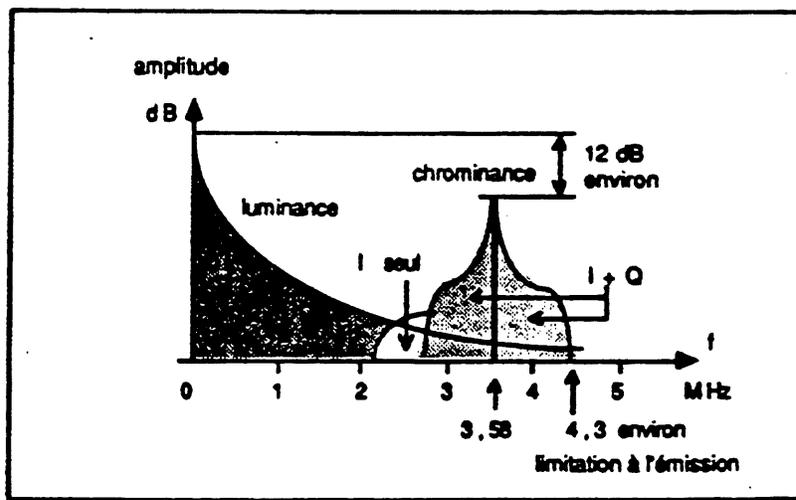
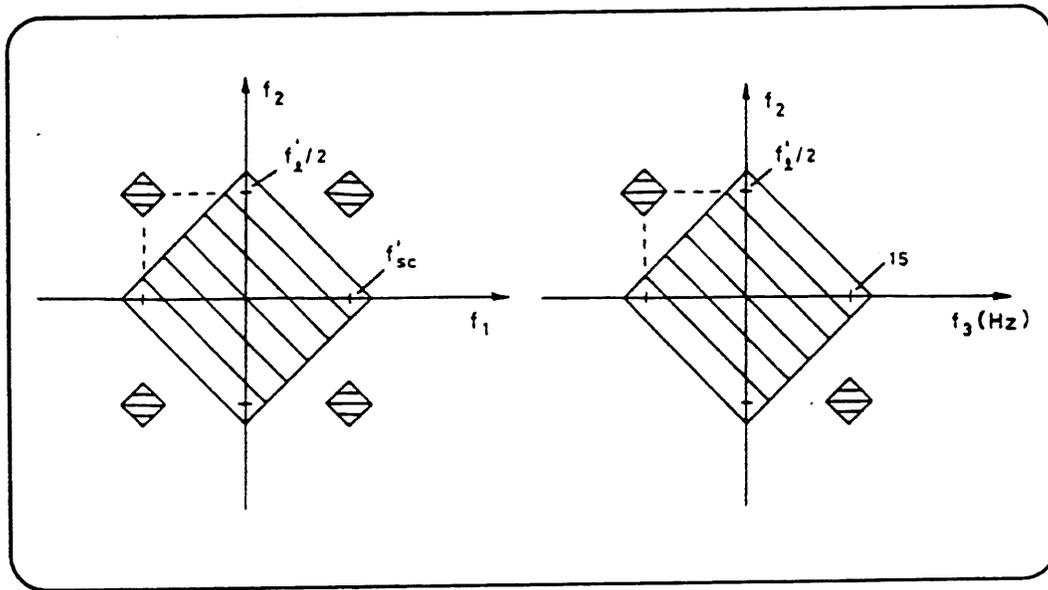
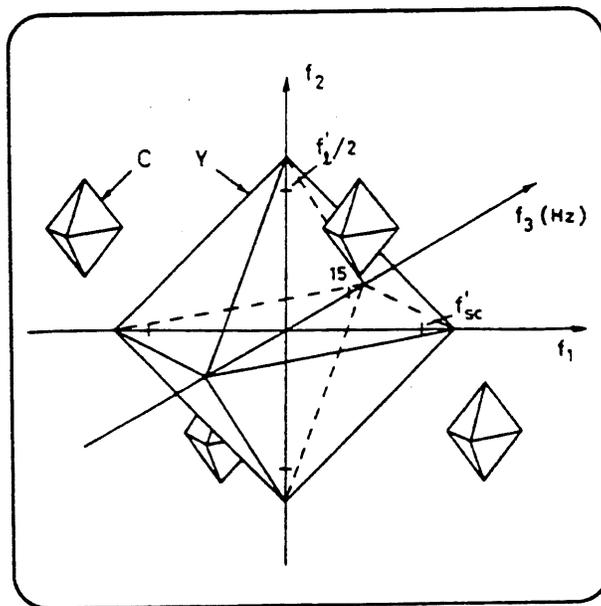


FIG 2.4. Densité du spectre vidéo. Système NTSC [3]



.a.



.b.

FIG 2.5. (a) Spectre du signal NTSC dans le domaine spatial et dans le domaine vertical-temporel (b) vue en perspective du spectre du NTSC [7]

Chapitre 2

2.2.2 Encodage PAL:

Le système NTSC présente de nombreux avantages mais il est sensible aux distorsions de phases. Toute *distorsion égale ou supérieure à $\pm 5^\circ$ d'angle de phase conduit à un défaut de teinte appréciable à l'oeil.*

Afin de remédier à ce défaut, le docteur Walter Bruch a imaginé et a mis au point, pendant les années 1962 et 1963, le système PAL [4].

Ce système est dérivé du NTSC. Les signaux I et Q ne sont pas employés, mais plutôt U et V. La différence essentielle avec le NTSC réside dans le fait, au début de chaque ligne, on inverse la phase de la sous porteuse modulée par V, d'où le nom du système PAL étant l'abréviation de « *Phase Alternate Line* ». Cette inversion alternée est continue dans le temps indépendamment des trames et des images. À la réception, on rétablit la phase correcte de la sous-porteuse en déphasant de 180° de la composante V la ligne qui avait déjà été inversée à l'émission. La figure 2.6 nous montre alors le schéma synoptique d'un codeur PAL. On remarque tout de suite deux nouveaux signaux, caractéristiques du système PAL, U et V. Ils s'obtiennent par un simple matricage des signaux de différence de couleurs selon les deux équations :

$$U = 0.493(EB - EY) \quad (2-12)$$

$$V = 0.877(ER - EY) \quad (2-13)$$

Ces deux signaux ne profitent pas de l'asymétrie de l'acuité visuelle. Ils sont traités de la même manière avec une limitation de la bande dont les spécifications sont les suivantes [26] :

Affaiblissement: $\leq 3\text{dB}$ à 1.3 MHz.
 $\approx 6\text{dB}$ à 1.6 MHz.
 $> 20\text{dB}$ à 4 MHz.

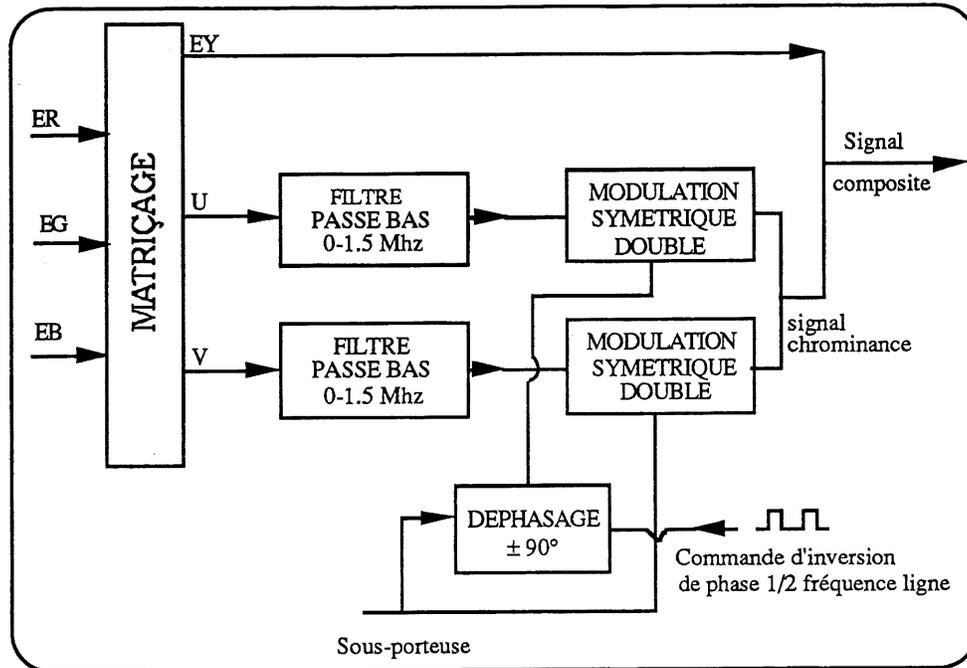


FIG 2.6.: Deux filtres passe bas limitent la bande U et V à 1.5 MHz. Ces signaux modulent la sous-porteuse couleurs à 4.43 MHz dans le modulateur symétrique double à porteuse supprimée. Le modulateur U reçoit une sous-porteuse déphasée de $+90^\circ$ par rapport à la sous-porteuse du modulateur V pour les lignes de la phase $+0$ et déphasée de -90° pour les lignes de phase -0 .

2.2.2.1. Signal composite PAL

- Constituants :

Le signal composite en codage PAL a la même structure générale que le système NTSC. Il est formé :

- . de composante de luminance
- . des composantes de chrominance
- Sous-porteuse de chrominance [3]

L'alternance de phase de V a pour effet de décaler de $f_h/2$ les raies du spectre du signal V par rapport à celle du signal U. L'analyse de meilleure compatibilité a conduit à

Chapitre 2

choisir f_{sc} de façon à produire un décalage de $f_h/4$ des raies de chrominance par rapport au raies de luminance voisines localisées à $m f_h$.

Avec la précision d'un décalage entre raies secondaires, le choix a donné, pour le système à 625 lignes :

$$f_{sc} = 284 f_h - 0.25 f_v + 0.25 Hz \quad (2-14)$$

soit avec $f_h = 15\,625\text{ Hz}$ $f_{sc} = 4\,433\,618.75\text{ Hz}$

Cette valeur est désignée couramment par son approximation 4.43 MHz.

La valeur donnée à f_{sc} peut encore s'exprimer sous des formes pouvant s'avérer plus simples. On a :

$$f_{sc} = \left(\frac{1135}{4} + \frac{1}{625} \right) \cdot f_h \quad (2-15)$$

ce qui donne inversement :

$$f_h = \frac{4 f_{sc}}{\left(1135 + \frac{4}{625} \right)} \quad (2-16)$$

2.2.2.2. Spectre de Fourier unidimensionnel

Dans le signal composite PAL qu'on peut écrire sous la forme [3]:

$$E_c(t) = Y(t) + U \sin(2\pi f_{sc} t) \pm V \cos(2\pi f_{sc} t) \quad (2-17)$$

(- ; pour les ligne paire)

les deux signaux modulateurs U et V sont des signaux structurés sur les fréquences horizontale et verticale f_h et f_v . Le signal U module f_{sc} à phase fixe et son spectre est formé de raies primaires distantes de $\pm n \cdot f_h$ de la fréquence f_{sc} . Le signal V module f_{sc} à phase alternée en ligne donc à fréquence $f_h/2$. Le spectre associé est formé de raies primaires distantes de $\pm(0.5 \cdot f_h + n f_h)$ de la fréquence f_{sc} .

Chapitre 2

Le spectre de chrominance modulée est la somme de ces deux spectres, qui comporte des raies distantes de f_{sc} de $f_H/2 \pm k f_H$. À l'intérieur du spectre du signal de luminance EY et des raies principales (Y) se place le spectre du signal de chrominance modulé avec les raies primaires associées aux deux composantes, U et V . (figure 2.7)

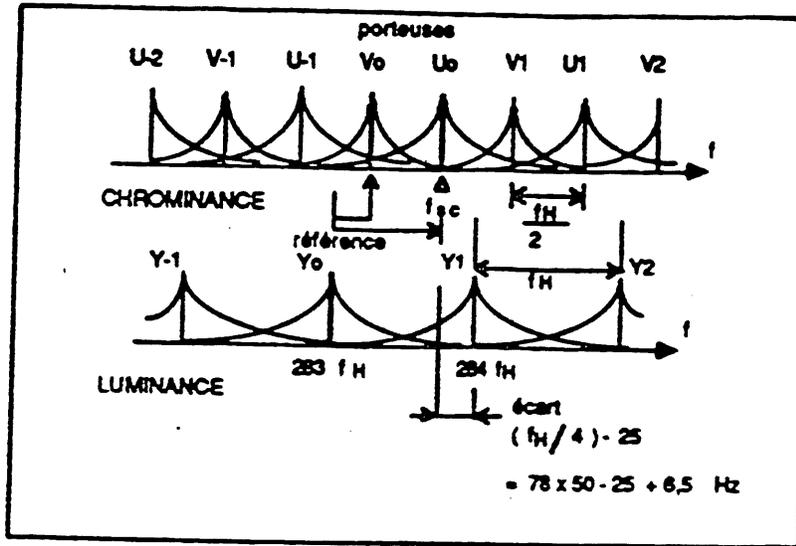


FIG 2.7. Structure luminance chrominance en PAL [3]

La bande occupée par la chrominance est identique pour les signaux U et V . La bande latérale inférieure occupe environ 1.5 MHz. (figure 2.8).

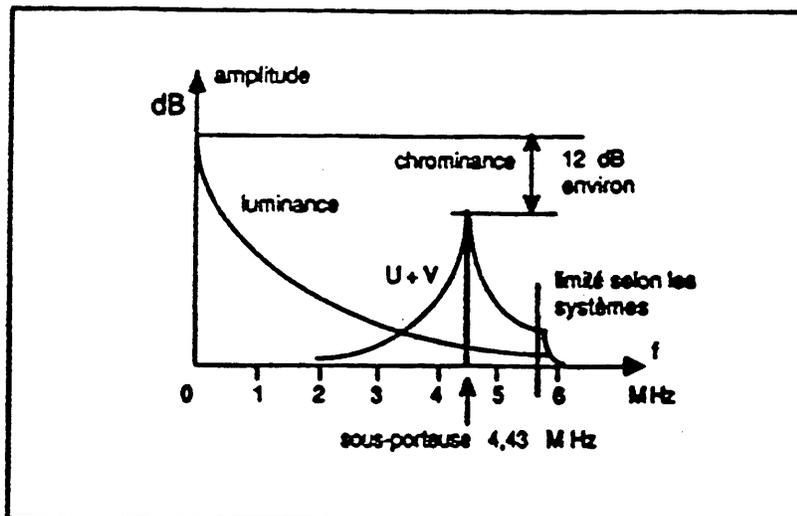


FIG 2.8. Densité du spectre vidéo [3]

Chapitre 2

2.2.2.3. Spectre de Fourier multidimensionnel [10]:

L'espace de Fourier tridimensionnel associe les axes:

Axes	Fréquence
Temporel	f en Hz
Vertical	fv en cph cycle/ picture height
Horizontal	fh en MHz

Dans le domaine spatial, les deux signaux U et V modulés représentent deux spectres différents dont les centres sont :

$$\text{signal } U : (f_h, f_v) = (+f_{sc}, +781/8).$$

$$\text{signal } V : (f_h, f_v) = (-f_{sc}, +781/8).$$

Le spectre complet comprend également les spectres symétriques centrés sur

$$\text{signal } U : (f_h, f_v) = (-f_{sc}, -781/8).$$

$$\text{signal } V : (f_h, f_v) = (+f_{sc}, -781/8).$$

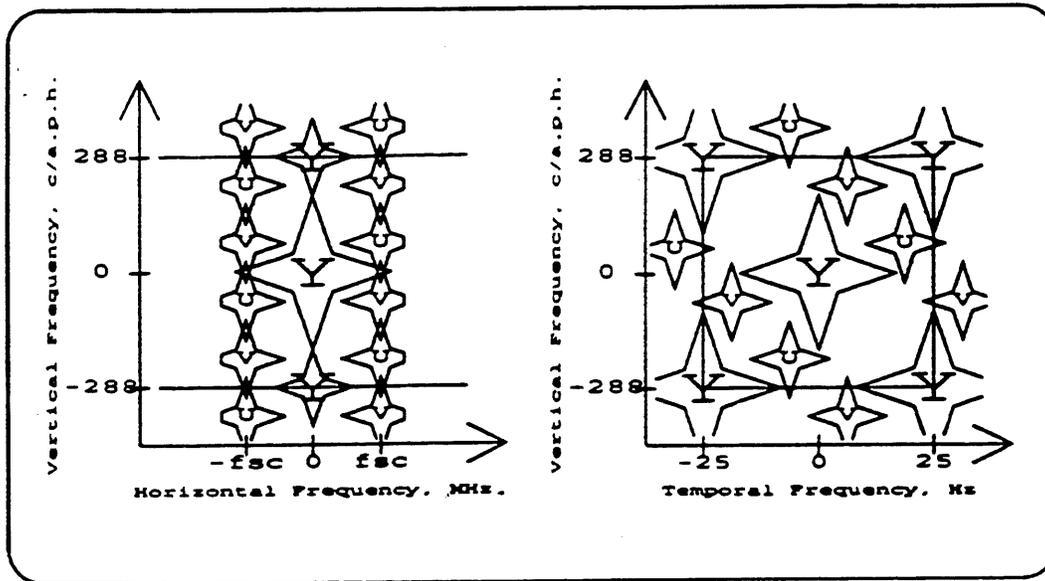
Le spectre dans ce domaine est donné par la figure 2.10.

Dans le domaine vertical temporel ces satellites de chrominance se trouvent centrés au :

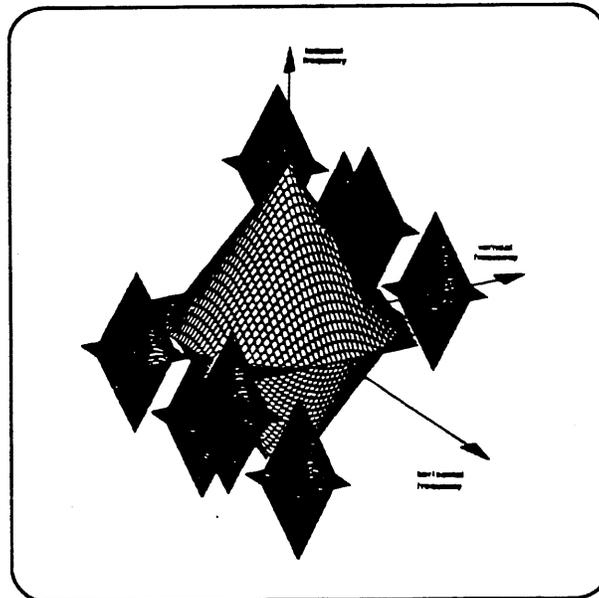
$$\text{signal } U : (f_v, f) = (781/8, 183/4) \text{ et } (-2343/8, -61/4)$$

$$\text{signal } V : (f_v, f) = (-781/8, -183/4) \text{ et } (2343/8, 61/4)$$

Cette organisation est à apprécier dans l'espace à trois dimensions. (Figure 2.9a). Cette organisation peut être comparée à celle obtenue pour le système NTSC. La modulation de phase de la composante V dédouble les volumes de chrominance et l'ensemble est moins favorable qu'en NTSC pour donner au spectre de chrominance un volume propre aussi grand que possible.



.a.



.b.

FIG 2.9. (a) Spectre du signal PAL dans le domaine spatial et dans le domaine vertical-temporel (b) vue en perspective du spectre du PAL [10].

Chapitre 2

2.3. Décodage conventionnel des signaux NTSC et PAL:

2.3.1. Décodage conventionnel NTSC :

Le rôle essentiel d'un décodeur est de séparer les signaux de luminance des signaux de chrominance. En regardant la figure 2.10 nous voyons qu'au point A nous disposons du signal couleur intégral. Deux voies se présentent alors:

A. LA VOIE DE LUMINANCE: restituant le signal EY où une trappe accordée à la fréquence de la sous porteuse chrominance élimine toute interférence avec ses derniers signaux et permet ainsi d'obtenir une exactitude de luminosité relative entre les différentes couleurs.

B. LA VOIE DE CHROMINANCE: sélectionnée par un filtre passe-bande. Au point B nous disposons du produit de modulation du signal de chrominance résultant EC.

Il est appliqué à la fois au démodulateur synchrone chargé de différencier la composante de chrominance I et au redresseur synchrone chargé de différencier la composante de chrominance Q.

Ainsi nous avons donc simultanément d'une part:

Les signaux I et Q à la sortie de chacun des deux filtres passe-bas et d'autre part le signal de luminance EY.

Ces trois composantes subissent ensuite un " matricage " afin d'obtenir les signaux ER, EG, EB.

2.3.2. Décodage PAL:

Le décodeur PAL est très voisin de celui du NTSC. En fait, la seule modification que l'on doit signaler par rapport au schéma synoptique du décodeur NTSC (Figure 2.10) est le circuit représenté par la case marquée: déphasage -90° .

A cet endroit, le schéma peut se modifier par rapport à celui du NTSC de la manière définie en figure 2.11.

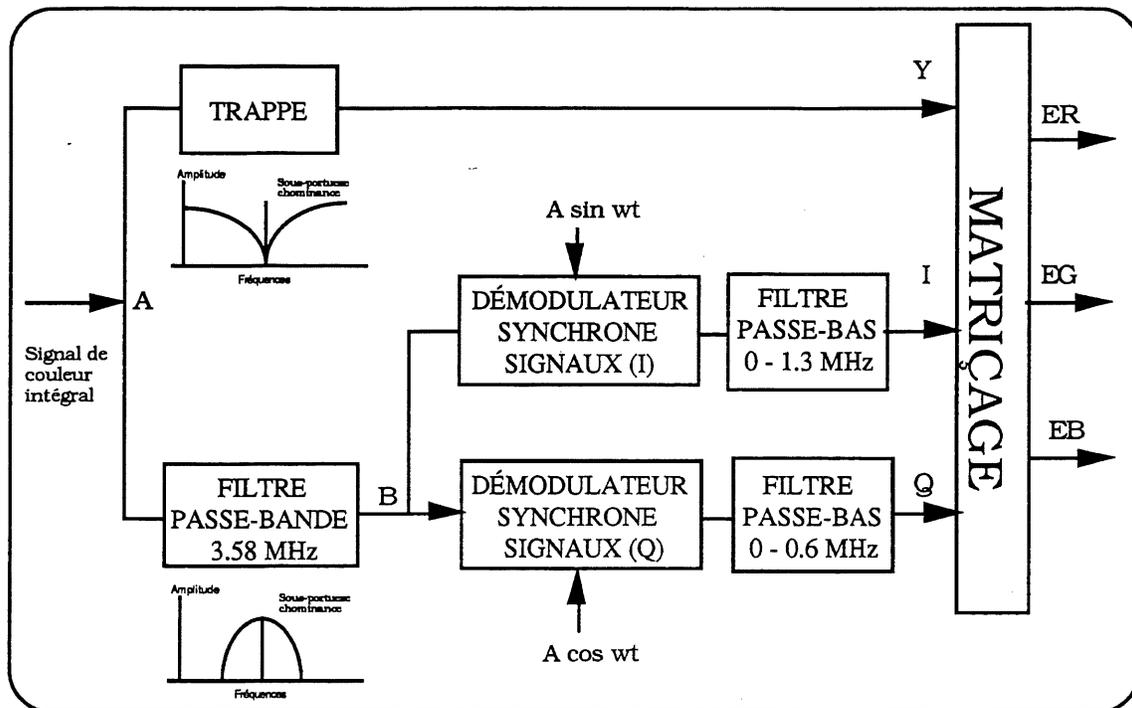


FIG 2.10. Schéma synoptique d'un décodeur NTSC

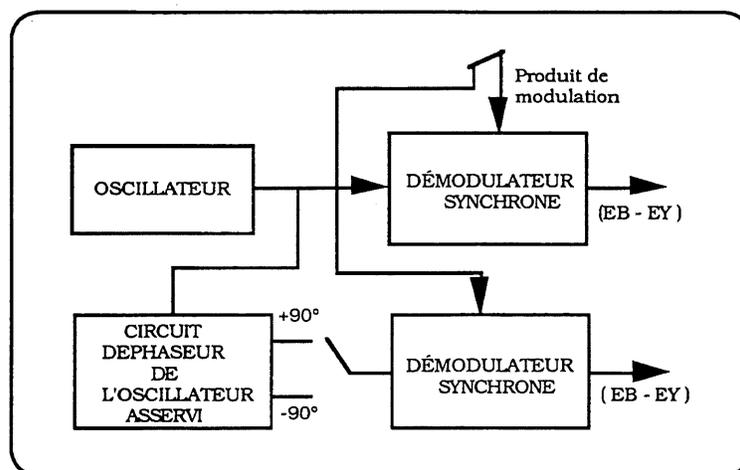


FIG 2.11.

Chapitre 2

2.4. Diachromie et luminance croisée

La condition la plus pénalisante de la compatibilité avec le vidéo noir et blanc, est la nécessité de placer le spectre du signal de chrominance dans la bande vidéo du signal de luminance. La séparation de ces deux signaux devient le point le plus important pour la qualité de l'image finale .

Le signal NTSC ou PAL résultant d'un décodeur conventionnel simple comporte en général deux défaillances standards qui sont par ordre décroissant d'importance :

- la diachromie ou ce qu'on appelle en anglais « cross-color »
- la luminance croisée ou encore en anglais « cross-luminance »

2.4. 1. La diachromie [21]

C'est la défaillance majeure produite par le chevauchement spectral des composantes de chrominance et de luminance. C'est en fait l'interprétation des composantes de luminance de haute fréquence comme étant des composantes de chrominance. La diachromie est probablement la pire dégradation des signaux PAL et NTSC. Son effet visuel sur l'image peut être décrit comme un effet d'arc-en-ciel (ce phénomène est souvent perçu dans les parties de l'image où des composantes diagonales de luminance à hautes fréquences sont présentes). La chemise rayée d'un arbitre de hockey est un bon exemple de ce phénomène ennuyeux.

L'effet subliminal de cette défaillance est aussi un facteur psychophysiologique. Pour le NTSC, par exemple, puisque ce système est une séquence de 4 champs, la phase d'une interférence couleur (diachromie) change de 180° d'un champ à l'autre. À cause de cela, les objets stationnaires dans l'image vont papilloter à la fréquence de cette séquence i.e. 15 Hz, une fréquence bel et bien inférieure à la fréquence du confort de l'oeil humain.

Chapitre 2

Comme résultat, ce papillotement sera la cause d'une fatigue si l'on regarde le poste récepteur pour une longue durée.

La diachromie dans les deux standards PAL et NTSC est devenue pire au long des années, principalement à cause des améliorations considérables qui ont été apportées aux caméras couleurs. Au début, les caméras produisaient des images couleurs douces (ne contenant pas des transitions de couleurs rapides) parcequ'elles n'avaient pas la capacité de la fonction de transfert de modulation (MTF [6]) à des luminances de haute fréquence que les caméras d'aujourd'hui possèdent. Ainsi, les effets de diachromie n'étaient pas aussi visibles qu'elles le sont aujourd'hui. Maintenant avec des caméras dont la réponse en fréquence s'étend au-delà de 7 MHz et des images produites au moyen d'ordinateurs, généralement capables de générer des transitions très rapides dans différents angles, les effets de diachromie sont devenus très visibles et par conséquent très ennuyeux.

Puisque la diachromie est considérée comme l'artefact majeur produit par les systèmes d'encodage couleur PAL et NTSC, sa réduction ou son élimination vont contribuer énormément à l'amélioration des performances globales des deux systèmes.

2.4. 2. La luminance croisée [21]

C'est le deuxième artefact des deux systèmes d'encodage couleur. Il englobe les deux phénomènes connus en anglais sous l'appellation « hanging dots patterns» [1] et le « dot crawl»[1].

Cet artefact est dû à l'interprétation des composantes de chrominance au domaine de transitions verticales comme des informations de luminance.

Même si la luminance croisée est visuellement moins ennuyeuse que la diachromie, elle est tout de même, une indication pour un bon observateur, que l'image produite n'est pas de la qualité d'une image RGB, et que chaque effort pour minimiser ou éliminer ce phénomène est recommandé.

Chapitre 2

Les deux récents développements technologiques à savoir :

a. la prolifération d'une variété de propositions de standards de télévision à haute définition et

b. l'émergence des mémoires de coût très bas pour les récepteurs TV ont conduit à une réévaluation des standards d'encodage vidéo couleur.

En effet, l'impact technique majeur des démonstrations des différents systèmes à haute définition, lors des conférences et les expositions d'équipements de télévision par-tout au monde, a été de motiver les chercheurs à réexaminer les systèmes d'encodage vidéo couleur servant au public.

Dans le parc du NTSC par exemple, où plus de 200 millions de récepteurs TV représentent presque 60% du nombre global des récepteurs du monde, l'adoption présente d'un système de haute définition est difficilement concevable, à cause de contraintes sociales (un récepteur TV à haute définition coûte plus de 5 000\$).

Cela a mené à une variété de recherche afin d'explorer les différentes façons avec lesquelles les standards NTSC et PAL conventionnels peuvent être améliorés, en particulier quand quelques unes des techniques développées pour les TV hautes définitions seront appliquées à ces derniers.

Les différentes approches proposées pour l'amélioration des systèmes d'encodage et de décodage vidéo couleurs sont généralement basées sur un filtrage numérique à deux ou à trois dimensions. Le but principal d'un filtre multidimensionnel est de limiter le spectre d'un signal à des aires spécifiques du domaine fréquentiel, afin d'éviter le chevauchement lors du multiplexage des deux composantes du signal composite.

Dans ce qui suit, on explorera les différentes approches proposées pour l'amélioration de l'encodage et du décodage vidéo couleur.

Chapitre 2

2.5. Encodage et décodage améliorés

2.5. 1. Encodage et décodage bidimensionnel spatial

2.5.1. 1. Encodage bidimensionnel

Y.C. Faroudja dans [20] affirme que la clé pour améliorer l'encodeur est d'éviter le chevauchement spectral produit par l'encodeur conventionnel en préfiltrant d'une façon appropriée, la luminance et la chrominance par un filtre en peigne avant addition. (Figure 2.12)

En fait, il propose un système d'encodage qui repose sur un filtrage bidimensionnel spatial. Dans celui-ci, la luminance et la chrominance sont gardées séparées, par un filtrage en peigne adéquat dans le domaine vertical et un filtrage passe bande dans le domaine horizontal, avant d'être ajoutées pour former ce qu'il a appelé le signal superNTSC composite [21]. De cette façon il sera possible par un filtrage similaire au décodeur, d'obtenir une image ne contenant aucun effet de diachromie ou de luminance croisée.

Cette manipulation bidimensionnelle, qui opère séparément dans les domaines vertical et horizontal, réduit d'une manière significative les interférences entre les signaux multiplexés.

Étant donné que les deux systèmes NTSC et PAL utilisent la « même technique d'encodage », le procédé décrit dans [20] et [21] est aussi applicable pour le PAL avec quelques modifications.

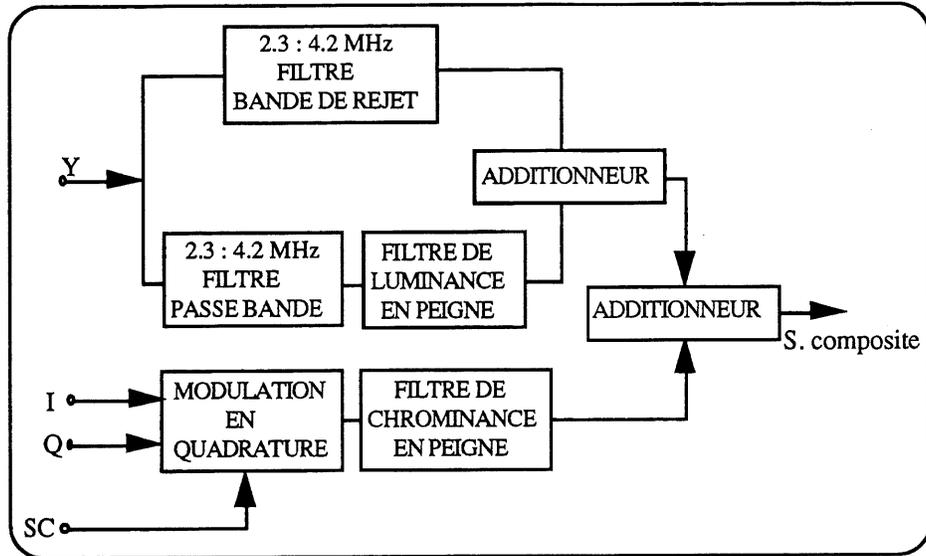


Fig 2.12. Encodeur amélioré. Système NTSC. [21]

2.5.1.2. Décodage amélioré

Faroudja affirme [20] que le degré de l'amélioration globale dans le récepteur avec l'approche ci-dessus détaillée, dépendra du type du décodeur au récepteur. L'encodeur ci-dessous mentionné si couplé avec un décodeur utilisant un filtrage similaire, permet l'obtention d'une image ne contenant « aucun effet » de diachromie ou de luminance croisée.

2.5. 2. Décodage adaptatif

Detlef Teichner [15] affirme qu'une application intensive des filtres a trois dimensions peut améliorer considérablement la qualité de l'image.

Après avoir exploré plusieurs techniques de pré et de post-filtrage à 3 dimensions pour l'élimination des artefacts d'interférence qui affectent le signal PAL, et après avoir évoqué leurs limitations, Teichner a proposé deux approches adaptatives : une selon le mouvement, l'autre selon le mouvement et les transitions verticales pour le multiplexage et la séparation des deux composantes de luminance et de chrominance.

Chapitre 2

Pour la séparation des deux composantes Teichner propose une panoplie de filtres à trois dimensions dont on cite :

- les filtres en peigne a retard trame: « field comb filters »

Lors de l'essai de ce genre de filtres, on a remarqué que ces derniers permettent une bonne reproduction des détails stationnaires orientés verticalement ou légèrement penchés. La perte de résolution et les artefacts de diachromie n'affectent que les transitions diagonales de très hautes fréquences. Aussi, les effets de la luminance croisée sont très mineurs dans les transitions verticales stationnaires de la chrominance, mais sont très perceptibles pour les détails en mouvement.

- Les filtres en peigne à retard image

Avec de pareils filtres, toutes les images stationnaires auront la pleine résolution des images au standard TV, et ne seront affectées par aucun effet d'interférence. Cependant, pour les détails en mouvement, il y aura une perte partielle dans la résolution et des effets de diachromie assez remarquables. Les effets des flous dans les détails exclusivement en mouvement sont similaires à ceux obtenus par des filtres en peigne à retard trame.

Les insuffisances et les échecs dans la séparation des composantes de luminance et de chrominance, lors de l'emploi d'un seul genre de filtre, ont conduit Teichner à considérer trois approches:

- une approche de connexions séries
- et deux approches adaptatives, une tenant compte du mouvement, l'autre tenant compte et du mouvement et des transitions verticales.

Chapitre 2

2.5.2.1. Approche des connexions séries

Dans cette approche, Teichner a pensé à combiner les avantages des filtres en peigne à retard trame pour les images stationnaires et les bénéfices des filtres en peigne à retard ligne pour les structures purement verticales. Avec le filtre passe bande en connexion série, ceci donne un filtre à trois dimensions.

Cette approche permet aux détails stationnaires verticaux et quelques diagonales dans les images stationnaires et en mouvement d'être reproduits sans aucune atténuation. Elle permet en outre une bonne réduction des effets de diachromie.

2.5.2.2 Approche adaptative selon le mouvement

Pour faire face aux défaillances de filtres en peigne à retard ligne et à retard trame en connexion série Teichner a proposé une utilisation adaptative tenant compte du mouvement.

Le principe de cette approche est le suivant :

Pour des images en mouvement, l'action des filtres en peigne à retard trame est inhibée et les filtres en peigne à retard lignes sont actionnés. Cela permet d'éviter les artefacts de diachromie et de luminance croisée causés par les filtres en peigne à retard trame pour les détails en mouvement. Par contre, lorsqu'il s'agit d'une image fixe, le filtre en peigne à retard ligne est inhibé et c'est le filtre en peigne à retard trame qui entre en action. Cela permet de préserver la pleine résolution et d'obtenir une image sans effets d'interférence pour le cas des images fixes.

Cependant, cette technique peut causer, entre autres, quelques effets dérangeants causés essentiellement par le comportement non linéaire du processus d'adaptation [15]. Aussi, il reste quelques effets résiduels de diachromie qui peuvent être plus importants que ceux obtenus par un filtre en peigne à retard trame. Également, des effets de luminance croisée restent toujours présents.

Chapitre 2

Puisque les performances de cette approche dépendront exclusivement de la qualité du détecteur de mouvement, Teichner a testé [14] plusieurs variétés de détecteurs de mouvement.

2.5.2.3. Approche adaptative selon le mouvement et les transitions verticales

Teichner affirme qu'une meilleure réduction de ces artefacts peut être réalisée par l'ajout d'un autre module d'adaptation selon les transitions verticales très rapides. Ce module permet l'inhibition de l'action du filtre vertical, lors de la détection de transitions verticales très rapides afin d'éviter le problème de luminance croisée.

On proposera dans le chapitre 5 une structure semblable à celle-ci.

2.5. 3. Codage et décodage améliorés tridimensionnels

2.5.3.1. Codage et décodage tridimensionnels non séparables

E. Dudois et al [8] ont introduit la forme en diamant du spectre du signal NTSC et, en conséquence, ont proposé un filtrage tridimensionnel en diamant non séparable qui comprend un filtre en losange non séparable et un filtre temporel additif.

Ce genre de filtre, si appliqué à la fois à l'encodeur et au décodeur, réduit d'une manière significative les artefacts d'interférence. Cependant, non seulement la complexité du filtre proposé est énorme, ce qui rend son réalisation très difficile, mais aussi les performances du filtre bande de rejet de la luminance ne sont pas idéales au long des axes horizontal et vertical.

Plus tôt, en 1976, Drewery [11], après avoir détaillé le spectre multidimensionnel des signaux vidéo PAL, a examiné en conséquence les origines des artefacts de diachromie et de luminance croisée qui affectent les signaux vidéo PAL lors du décodage.

Chapitre 2

Pour éliminer ces artefacts, Drewery affirme que la séparation totale des deux composantes de luminance et de chrominance n'est possible que si ces dernières ont été prétraitées avant d'être multiplexées. Ainsi, il a proposé une panoplie de filtres à trois dimensions non séparables formés par le produit d'un filtre horizontal et d'un filtre vertical temporel non séparable.

2.5.3.2. Codage et décodage tridimensionnels séparables

Pour réduire la complexité et faciliter la réalisation, C. T Le dinh [23] a proposé une variété de filtres à trois dimensions séparables en diamant qui se composent de simples filtres monodimensionnels. Ceux-ci opèrent individuellement dans les axes horizontal et vertical et dans les deux axes obliques.

En comparaison avec les filtres à trois dimensions déjà proposés, les filtres qu'il propose offrent des avantages considérables tels que la complexité réduite et de meilleures performances au long des trois axes horizontal, vertical et temporel du domaine fréquentiel.

CHAPITRE 3

ENCODAGE NUMÉRIQUE DES SIGNAUX PAL ET NTSC

Dans le chapitre précédent, nous avons décrit le principe d'encodage et de décodage standard NTSC et PAL. Nous avons vu, entre autres, comment à l'aide de filtrage multidimensionnel quelques-uns des artefacts causés par ce codage-décodage standard peuvent être éliminés.

À cette étape-ci, rappelons l'objectif de nos travaux de recherche. Nous avons deux signaux, un de luminance et un autre de chrominance, nous cherchons à insérer le spectre de chrominance dans celui de luminance sans qu'il y ait chevauchement entre ces deux signaux tout en préservant évidemment une résolution acceptable.

Dans ce chapitre, après avoir présenté le schéma de simulation d'un encodeur numérique standard, nous proposerons deux différentes techniques d'encodage numérique permettant un multiplexage des signaux de luminance et de chrominance avec le moindre chevauchement; une technique opérant le long de l'axe horizontal, l'autre sur l'axe horizontal et l'axe vertical à la fois, nous en déduirons une structure opérant le long des trois axes horizontal, vertical et temporel du domaine fréquentiel.

Cependant, vu que les images que nous possédons sont enregistrées selon le format D1 du standard CCIR 601 [26] appelé aussi format D1 et qu'elles doivent être traitées en format 4fsc ou D2, nous commencerons par étudier ce format d'enregistrement pour après voir la méthode de création des signaux composites numériques PAL et NTSC dans le format 4fsc.

3.1. Format D1 ou Format 4fsc

Considérant, d'une part, qu'une approche numérique compatible facilitera l'échange international en matière de programmes et, d'autre part, que le " co-siting " [6] des échantillons représentant la luminance et les signaux différences de couleurs facilitera le traitement des signaux composites numériques, le CCIR a créé le format D1 [26]. Ce format permet d'enregistrer d'une manière assez transparente l'information numérique qui résulte de l'échantillonnage du signal vidéo, 625/50 pour le PAL ou 525/60 pour le cas du NTSC, à 13.5 MHz (4*3.375 [6] [26]) pour la luminance désignée toujours par Y et 6.75 MHz (2*3.375) pour les composantes de chrominance désormais désignées par Cr et Cb. Dans ce format, les signaux Y, Cr et Cb s'obtiennent par une simple opération de matricage des signaux ER, EB et EG. L'origine de l'appellation vient du fait que nous avons :

$$\langle Y:Cr:Cb \rangle \equiv \langle 13.5:6.75:6.75 \rangle = 3.375 \bullet \langle 4:2:2 \rangle$$

Dans ce format, les images renferment 486 et 576 lignes actives respectivement pour le 525/60 et le 625/50. Chacune de ces lignes contient 720 échantillons actifs pour la composante de luminance et 360 échantillons actifs pour chacun des deux signaux de chrominance.

3.2. Format D2 ou 4fsc

Vu que le traitement numérique couleur est plus facile à réaliser si le taux d'échantillonnage est un multiple de la sous-porteuse, et encore plus facile, en particulier, pour le système PAL si ce taux est 4 fois multiple de la sous-porteuse [6], le format D2 a été créé. Ce format peut être obtenu à partir des images du format D1.

Chapitre 3

3.3. Encodage numérique NTSC

3.3.1. Conversion du format D1 en format D2

Cette conversion constitue la première étape d'encodage. En fait, c'est la translation de la fréquence d'échantillonnage du format D1 à la fréquence d'échantillonnage du format D2.

3.3.1.1. Conversion de fréquence des composantes de chrominance Cr et Cb

Le rapport 70 : 33 est approximativement égal à $14.31818 : 6.75$ le rapport entre le taux d'échantillonnage du format D2 et celui du taux d'échantillonnage du format D1 pour les composantes de chrominance avec une erreur relative de 0.13 pour million.

Donc pour transiter du format D1 au format D2, nous utiliserons 33 échantillons du format D1 pour produire 70 échantillons du format D2; ce qui veut dire qu'on utilisera 70 filtres différents ou phases différentes pour réaliser correctement cette conversion. Évidemment, vu que dans le format D1, nous avons 360 échantillons par ligne, en format D2, on aura pour le système NTSC 764 échantillons actifs (764 pixels) par ligne dans une structure de 488 lignes (le même nombre de ligne que dans le format D1).

3.3.1.2. Conversion de fréquence des composantes de luminance

Pour le cas de la luminance, le rapport 35:33 est approximativement égal à $14.31818 : 13.5$, le rapport du taux d'échantillonnage du format D2 et celui du format D1. Alors, pour convertir le format D1 en format D2, nous utiliserons 33 échantillons du format D1 pour produire 35 échantillons du format D2, et nous aurons besoin de 35 filtres différents ou phases différentes pour faire cette conversion.

Remarque :

Une simplification peut être apportée à la conversion de fréquence des composantes de chrominance. En fait, si on ramène la fréquence de ces composantes du format D1 à la

fréquence de la composante de luminance dans ce même format en faisant un sur-échantillonnage de 2 et une opération d'interpolation nous pouvons par la suite utiliser les mêmes filtres de la conversion de fréquence de la composante de luminance pour les composantes de chrominance.

La structure globale de cette conversion fréquentielle est présentée dans la figure

3.1

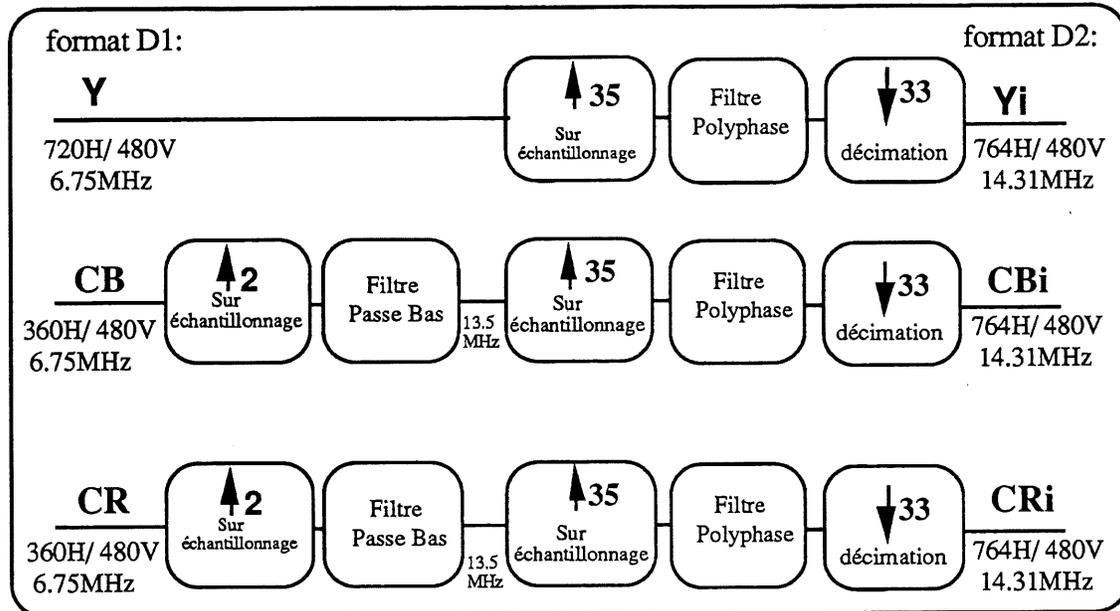


FIG 3.1. Schéma bloc de la conversion fréquentielle.
Système NTSC

3.3.1. Filtres utilisés:

A. Filtre passe bas:

C'est un filtre passe bas gaussien. Les coefficients sont donnés au tableau 1 de l'annexe.

B. Filtre polyphase:

La conception de ce filtre a été faite par la méthode de fenêtrage de Kaiser. Les spécifications sont les suivantes:

Chapitre 3

Fréquence de la bande passante $F_p = 5.515$ MHz.

Fréquence de la bande de réjection $F_s = 7.985$ MHz.

La fréquence de coupure est alors $F_c = 6.75$ MHz.

Atténuation dans la bande passante $A_p < .01$ dB

Atténuation dans la bande de réjection $A_a > 40$ dB

L'ordre du filtre optimal est $N = 561$. Comme l'ordre du filtre est très grand, pour minimiser le temps de traitement, la technique du filtre en multi- phases est utilisée [24] [25]. Cette technique permet d'éviter les multiplications des zéros inutiles. Les coefficients de ce filtre sont donnés dans l'annexe.

3.3.1.4. Formation du signal NTSC composite: encodage standard ou OD:

Après avoir converti les signaux numériques Y , Cr et Cb du format D1 au format D2, nous devons former les signaux de chrominance caractéristiques du système NTSC, à savoir les signaux I et Q . Ces deux signaux s'obtiennent par le matriçage suivant:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = 1.169 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.077 & -0.4662 \\ 0 & 0.6390 & 0.7143 \end{bmatrix} \begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} \quad (3-1)$$

Avant de passer à la modulation qui consiste à multiplier le signal I par $\cos(w_{sc}t + \emptyset)$ et le signal Q par $\sin(w_{sc}t + \emptyset)$ avec $\emptyset = 33^\circ \times (\pi/180^\circ)$, ces deux signaux sont tout les deux limités à la même largeur de bande comme spécifiée ci-dessous [26]:

$A_a > 20$ dB $f < 1.3$ MHz.

$A_p < 0.2$ dB $f > 3.6$ MHz

Pour former le signal numérique composite NTSC, il suffit d'additionner le signal de luminance aux deux signaux de chrominance modulés selon la figure 3.2.

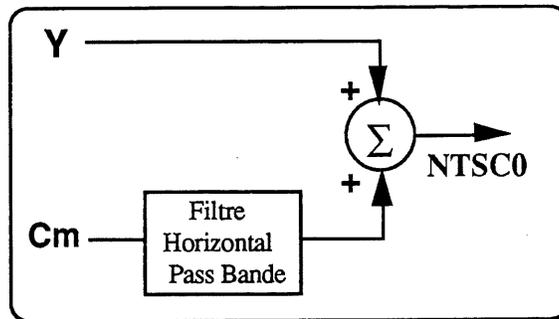


FIG 3.2.

Cm: étant la chrominance modulée

NTSC0: le signal NTSC relatif à l'encodeur OD.

En fait, ce schéma correspond à la structure d'un encodeur standard qu'on désignera dans ce qui suit par encodeur OD où aucun traitement n'a été performé sur la composante de luminance. Sur la figure 3.3, nous représentons le schéma complet d'un encodeur NTSC standard (OD)

3.3.2. Encodage numérique proposé:

L'introduction de pré-filtrage dans l'encodeur NTSC ne constitue pas un nouveau standard, il améliore uniquement la qualité de l'image. Cette amélioration dépend exclusivement de la façon dont les filtres introduits agissent.

En tout, nous proposons trois configurations.

Soit

- $H_{E,y}$ le filtre de la luminance à l'encodeur.
- $H_{E,C}$ le filtre de la chrominance à l'encodeur.

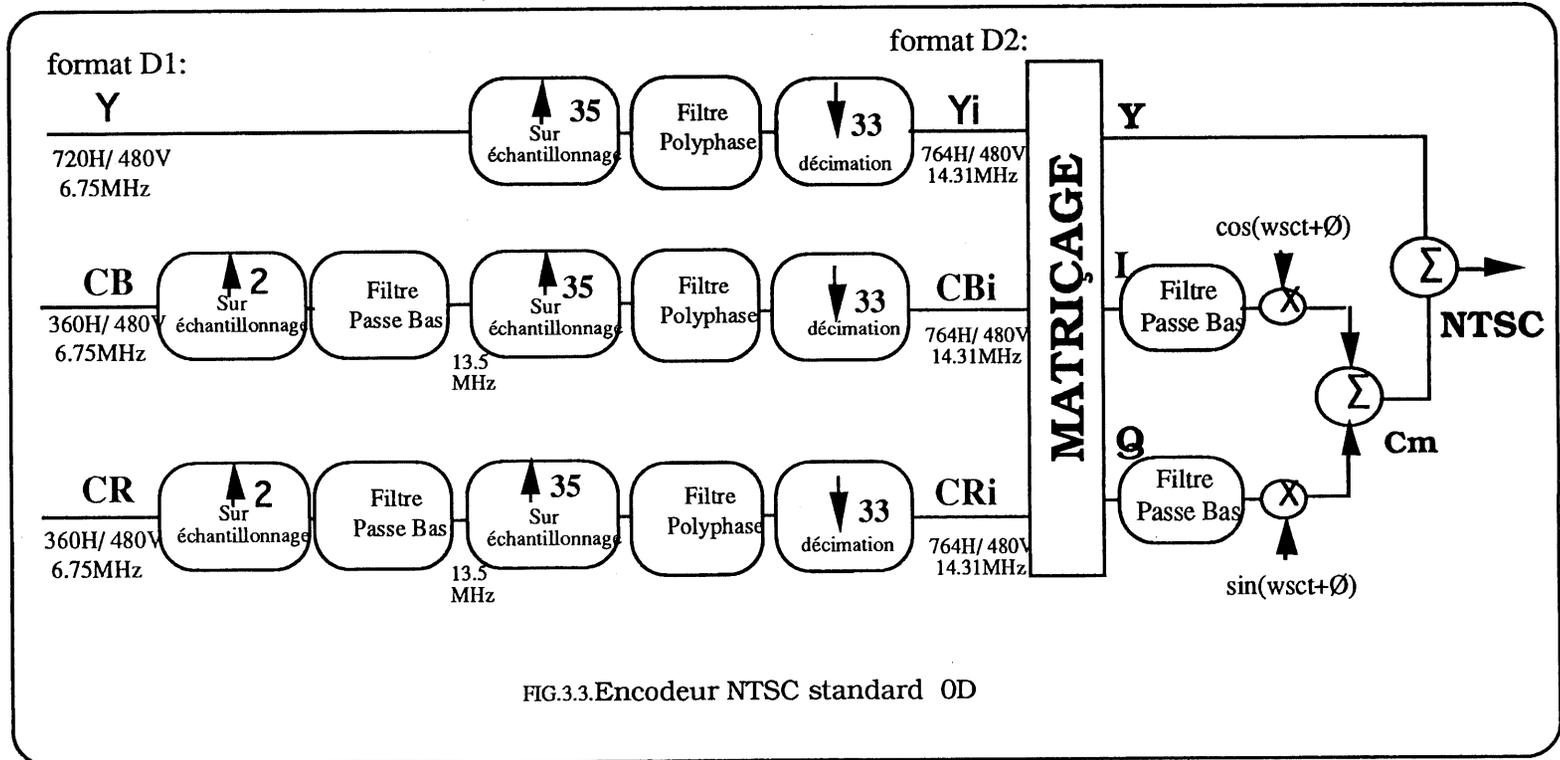


FIG.3.3.Encodeur NTSC standard 0D

3.3.2.1. Encodage avec traitement unidimensionnel: configuration 1D

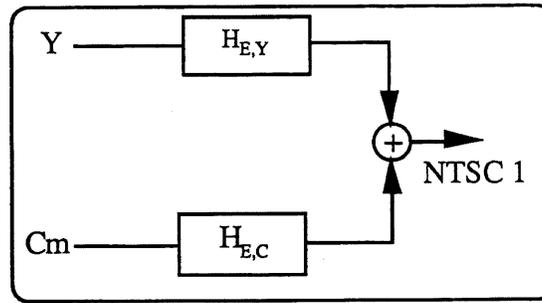


FIG.3.4. Configuration 1D

$H_{E,Y}$: est un filtre à bande de rejet.

$H_{E,C}$: est un filtre passe bande.

Si ces deux filtres sont complémentaires c.-à-d. $H_{E,Y} = 1 - H_{E,C}$, la configuration de la figure 3.4 devient comme montré à la figure 3.5. Dans tout ce qui suit, les filtres de la luminance et de la chrominance sont choisis d'une manière complémentaire (ceci diminuera la complexité du système).

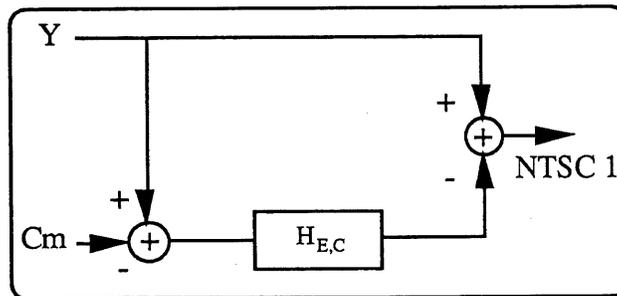


FIG 3.5. Configuration 1D avec des filtres complémentaires

Ce genre de traitement diminue énormément les interférences entre la luminance et la chrominance étant donné que ces dernières sont gardées séparées respectivement par le filtre bande de rejet et le filtre complémentaire passe bande. Cependant, avec cette configuration, il y aura une perte dans la résolution de la luminance.

3.3.2.2. Encodage avec traitement bidimensionnel

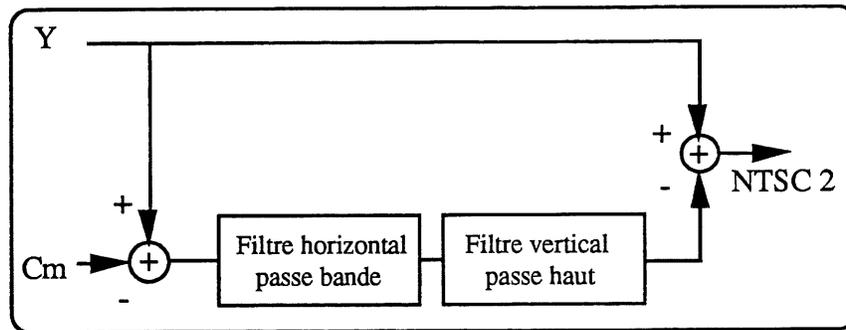


FIG 4.5. Configuration 2D

Dans cette configuration, deux filtres sont utilisés. Un filtre passe bande agissant sur l'axe horizontal, un autre passe haut agissant selon l'axe vertical. Ainsi avec ce genre de traitement, nous exploitons très bien la disposition spatiale des "satellites" de luminance et de la chrominance dans le domaine de fréquences spatiales pour bien les séparer avant leurs multiplexage. Grâce à ce traitement, la résolution de la luminance à la sortie de l'encodeur est nettement améliorée par rapport à la résolution du traitement unidimensionnel.

3.3.2.3. Encodage avec traitement tridimensionnel: configuration 3D

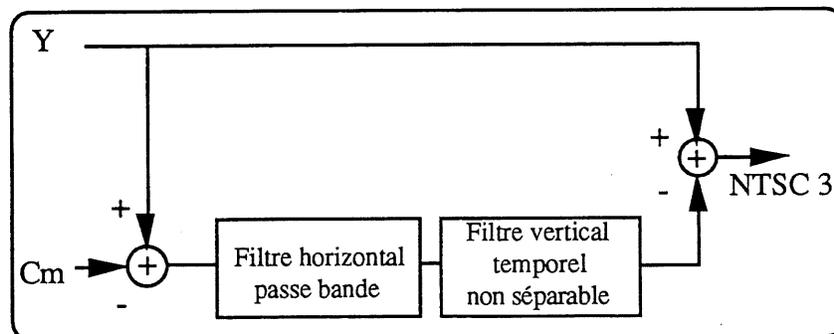


FIG 4.6. Configuration 3D

Pour diminuer davantage les interférences et surtout pour améliorer la résolution de la luminance selon l'axe temporel, un traitement tridimensionnel est suggéré. Dans cette configuration, deux filtres sont utilisés. Un filtre unidimensionnel horizontal et un

Chapitre 3

filtre non séparable (pouvant l'être) agissant à la fois dans les domaines vertical et temporel. On pourra ainsi exploiter finement la disposition des satellites de luminance et de chrominance dans l'espace de fréquence à trois dimensions afin de bien les séparer avant de les ajouter pour former le signal composite NTSC.

3.4. Encodage PAL

Comme pour le cas du NTSC, la première opération qui doit être réalisée lors de l'encodage est la conversion du format D1 en format D2.

3.4.1. Conversion du format D1 en format D2

3.4.1.1. Conversion de format pour les composantes de chrominance Cr et Cb

Pour le PAL, le rapport $118 : 45$ est approximativement égale à $17.72 : 6.75$, le rapport entre le taux d'échantillonnage du format D2 et celui du format D1 pour les composantes de chrominance. Ainsi, pour convertir le format D1 en format D2, nous utiliserons 45 échantillons du format D1 pour produire 118 échantillons du format D2, ce qui veut dire que nous aurons besoin de 118 filtres différents pour cette conversion. Comme dans le format D1 nous avons 360 échantillons actifs par lignes, dans D2 nous aurons alors 944 échantillons ou encore 944 pixels par ligne dans une structure de 480 lignes.

3.4.1.2. Conversion de fréquence de la luminance

Pour la luminance, le rapport $59:45$ est approximativement égal à $17.72:13.5$, le rapport du taux d'échantillonnage de D2 et celui du format D1 pour la composante de luminance. Pour cette translation, nous utiliserons 45 échantillons du format D1 pour avoir 59 échantillons de D2. Nous aurons alors besoin de 59 filtres différents pour la conversion. Dans le format D2, à la place de 720 échantillons par lignes nous aurons 944 échantillons actifs par ligne.

Chapitre 3

Comme pour le NTSC, une simplification peut être apportée à cette conversion en ramenant d'abord les composantes de chrominance à la fréquence de la composante de luminance, c'est à dire 13.5 MHz pour après passer à la fréquence du format 4 : 4 : 4, c'est à dire 17.72 MHz.

La structure globale de cette conversion est représentée dans la figure 3.7.

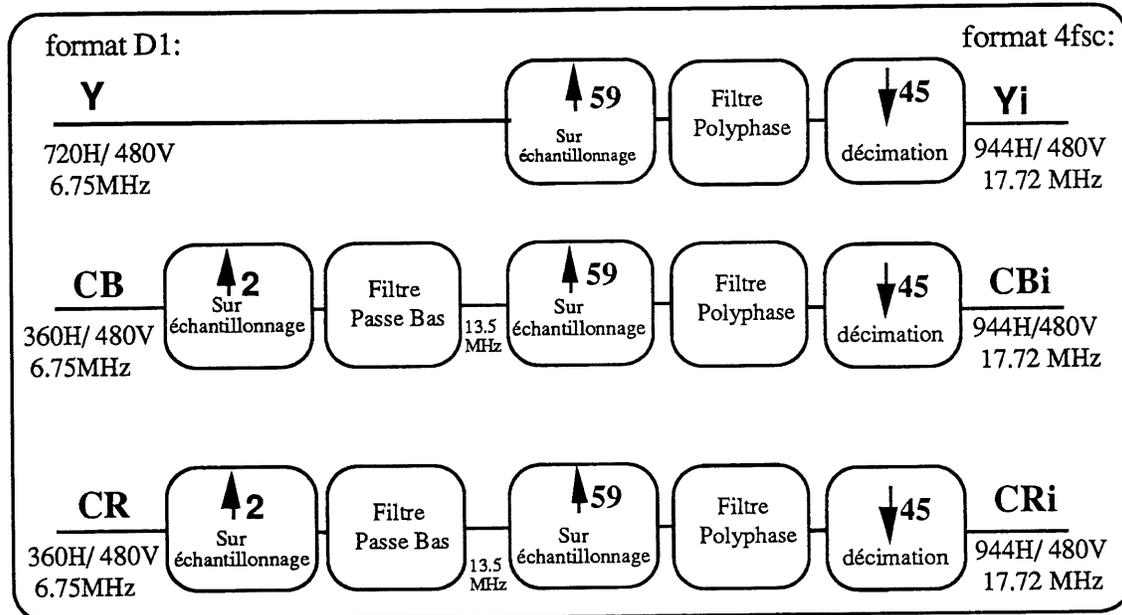


FIG 3.7. Schéma bloc de la conversion fréquentielle .
Système PAL.

3.4.2. Conception de filtres

A. Filtre passe bas:

C'est le même filtre utilisé pour le cas du NTSC.

B. Filtre Polyphase :

Pour concevoir ce filtre, nous utilisons les mêmes spécifications que celui du NTSC. Seule la fréquence d'échantillonnage change. Pour ce cas $F_e = 796.5$ MHz.

Chapitre 3

L'ordre optimal du filtre est $N=941$. Vu que l'ordre est assez élevé, pour minimiser le temps de traitement, nous utilisons la technique des filtres à multi-phases [24] [25].

3.4.3. Formation du signal PAL composite: encodage standard ou OD

Après avoir converti les signaux numériques Y , Cr et Cb du format D1 au format D2, nous formons les signaux de chrominance caractéristique du signal composite PAL à savoir les signaux U et V . La matrice permettant d'obtenir ces signaux est la suivante:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.077 & 0.493 \\ 0 & 0 & 0.877 \end{bmatrix} \cdot \begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix}$$

Nous limitons ensuite les largeurs de bandes de ces signaux. Pour former le signal composite numérique PAL nous additionnons au signal de luminance la somme des signaux filtrés U et V comme illustré à la figure 3.8.

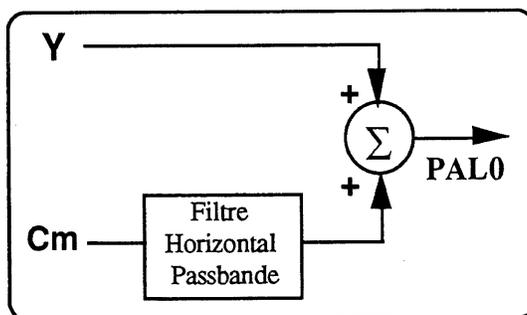


FIG 3.8. Encodage standard

Ce schéma correspond à la structure d'un encodeur standard désigné encore, comme pour le NTSC, encodeur OD où nous n'effectuons aucun traitement sur la composante de luminance.

La figure 3.9 représente le schéma complet d'un encodeur PAL standard (OD).

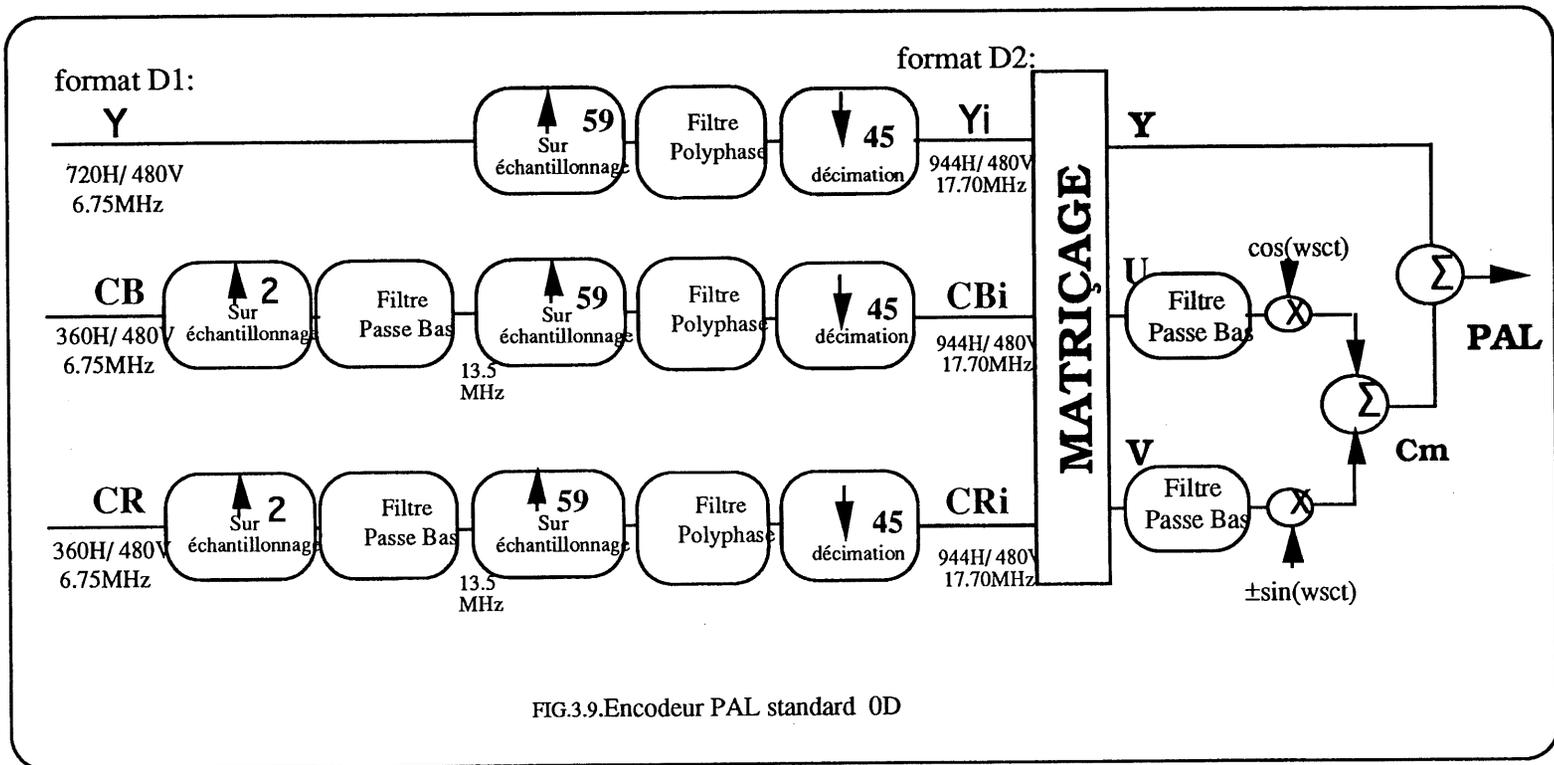


FIG.3.9.Encodeur PAL standard 0D

Chapitre 3

3.4.4. Configurations d'encodage proposées:

Pour le système PAL, nous proposerons les mêmes configurations multidimensionnelles que celles proposées pour le cas du NTSC. Les seules différences sont les suivantes:

- Pour la configuration 2D, le filtre vertical passe haut du NTSC est remplacé par un filtre vertical passe bande.

- Pour la configuration 3D, le filtre non séparable vertical-temporel est remplacé par un filtre non séparable complètement différent.

Pour des raisons de coût, de complexité, et afin de pouvoir incorporer la paire encodeur-décodeur dans un seul circuit intégré, les filtres utilisés à l'encodeur pour les deux systèmes NTSC et PAL vont être exactement le même que ceux utilisés au décodeur. Pour les mêmes raisons, la conception et les caractéristiques de ces filtres seront détaillées dans le chapitre relatif au décodage.

Les trois structures d'encodage numérique proposées dans ce chapitre exploitent bien les caractéristiques spectrales multidimensionnelles des composantes de luminance et de chrominance pour bien les multiplexer avec le moins de chevauchements possible.

Cependant, les performances de ces structures dépendent énormément de la façon dont le décodage est exécuté. En effet, même avec la structure d'encodage la plus complexe, 3D par exemple, nous pouvons obtenir une image contenant les deux artefacts de diachromie et de luminance croisée si un décodage médiocre a été exécuté lors de la séparation des composantes de luminance et de chrominance.

Chapitre 3

Pour cette raison, les performances de ces structures, à savoir les structures de décodage, seront discutées plus tard (chapitre 6) après avoir expliqué, dans le chapitre qui suit, le principe de décodage numérique proposé.

CHAPITRE 4

DÉCODAGE NUMÉRIQUE DES SIGNAUX NTSC et PAL

APPROCHE FIXE ET COMPATIBLE

Dans ce chapitre on introduira d'abord trois différentes techniques multidimensionnelles de décodage numérique. Ces trois techniques sont basées sur le principe de compatibilité entre l'encodeur et le décodeur (une compatibilité vis à vis la façon dont la luminance a été traitée à l'encodeur). Ces trois techniques ne contiennent aucun niveau d'adaptation, d'où le nom d'approche fixe et compatible.

Ensuite, on verra comment on peut regrouper ces trois techniques pour former ce qu'on appellera la structure de décodage sélectif.

Enfin, On étudiera les limitations de l'approche compatible pour ensuite adopter une seule structure à laquelle on ajoutera un circuit de compensation dont le but est d'améliorer sa performance, à savoir l'isoluminance.

4.1. Décodage NTSC

D'après les techniques de séparation des signaux de luminance et de chrominance citées dans le chapitre 2, on peut en venir à la conclusion suivante. Un filtre tridimensionnel exploite finement la disposition des spectres de luminance et de chrominance dans l'espace à trois dimensions pour bien extraire la composante de chrominance, et ce, aussi bien pour une image fixe que pour une image mobile indépendamment de la structure du codeur.

Les effets de perte de résolution dans les composantes de chrominance que ce genre de filtres peut engendrer (pour le cas d'images mobiles) peuvent être tolérés par l'oeil humain. En fait, la résolution de l'image est déterminée principalement par la

Chapitre 4

composante de luminance. De ce fait, dans les approches qu'on proposera par la suite, l'extraction des composantes de chrominance se fera par un filtre agissant dans les trois dimensions à la fois. Le filtrage tridimensionnel qu'on propose permet une bande de garde [18] entre la luminance et la chrominance, qui à son tour, réduit les effets de diachromie. La complexité de ce filtre sera discutée plus tard.

En ce qui concerne la composante de luminance et afin de préserver la résolution délivrée par l'encodeur (du point de vue performance, l'encodeur sera pris comme point de référence), on adoptera la règle de compatibilité. En effet, si à l'encodeur un traitement monodimensionnel a été effectué, un traitement similaire sera effectué au décodeur. Dans le cas d'un traitement bidimensionnel avant le multiplexage des composantes du signal composite, au décodeur un traitement bidimensionnel similaire sera effectué et la même chose s'effectuera dans le cas d'un traitement tridimensionnel.

Cela permettra, en quelque sorte, une compatibilité parfaite entre la structure du codeur et celle du décodeur, une compatibilité qui selon Y. C. Faroudja[20], et E. Dubois et al [9] permettra de réduire considérablement les artefacts d'interférence entre les composantes du signal composite, et même dans certains cas, d'obtenir une image sans interférences.

Il est à noter tout de suite que les appellations décodage 1D, 2D et 3D sont relatives à la façon dont la luminance est extraite. La chrominance, quant à elle, est toujours extraite par un filtrage à trois dimensions. Les structures qui en résultent sont les trois suivantes:

4.1.1. Décodage monodimensionnel ou 1D

Comme on peut le remarquer, cette configuration, illustrée à la figure 4.1, renferme un filtre passe bande agissant dans le domaine horizontal, dont le complément sert à extraire la luminance et un filtre vertical-temporel. En série, ces deux filtres

Chapitre 4

forment un seul filtre à trois dimensions, partiellement séparable qui extrait la chrominance.

4.1.2. Décodage bidimensionnel ou 2D

Dans cette structure, présentée dans la figure 4.2, on ajoute, dans la boucle de luminance, en plus du filtre horizontal, un filtre passe haut agissant dans le domaine vertical pour l'extraction de luminance. Si le signal NTSC, à la réception, a été engendré par un encodeur 2D, à l'aide de cette structure 2D aussi, la résolution de la luminance se trouve nettement améliorée par rapport à celle obtenue par un décodeur 1D. (Voir chapitre 6).

4.1.3. Décodage tridimensionnel ou 3D

Dans ce cas, et la luminance et la chrominance sont extraites par un filtre à trois dimensions. La résolution de la luminance est nettement meilleure que celle obtenue par les structures 1D et 2D. Si le signal NTSC a été engendré par un encodeur 3D, cette structure fournit une image non seulement de résolution tout à fait égale à celle de l'image originale (au moins sur les axes fréquentiels horizontal, vertical et temporel), mais aussi une image sans aucun artefact de luminance ou de diachromie.

Dans le cas contraire, c'est à dire si le signal a été encodé 1D ou 2D, ce mode de décodage peut causer des artefacts de luminance croisée pour le cas d'images en mouvement.

4.1.4. Décodage multidimensionnel sélectif:

En fait, les trois structures présentées ci-dessus peuvent être toutes les trois groupées dans une seule structure pour former ce qu'on appellera la structure de décodage sélectif, (voir figure 4.4).

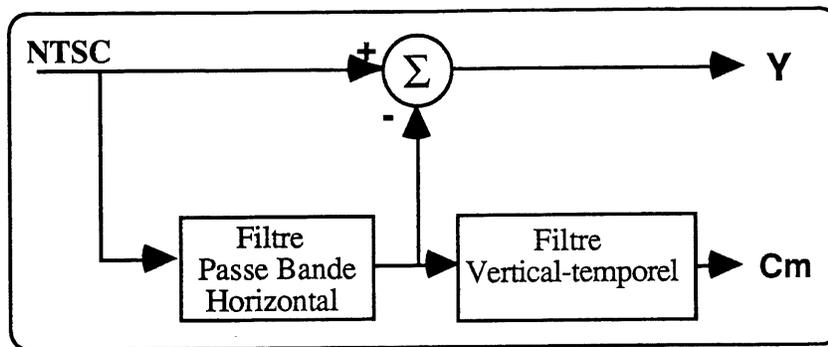


FIG 4.1. Structure du décodage monodimensionnel

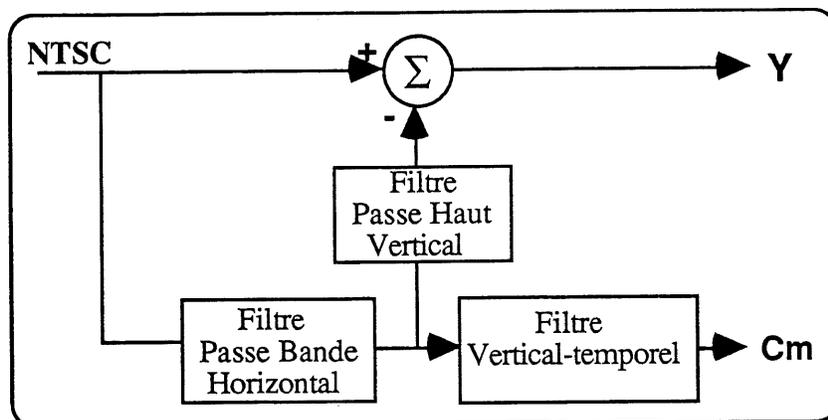


FIG 4.2. Structure du décodage bidimensionnel

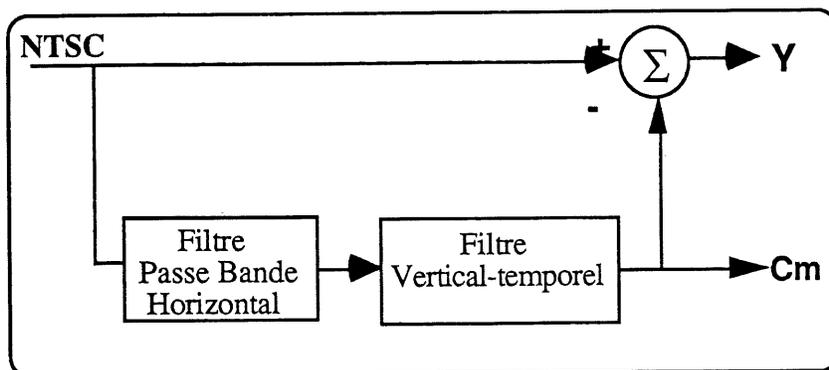


FIG 4.3. Structure du décodage tridimensionnel

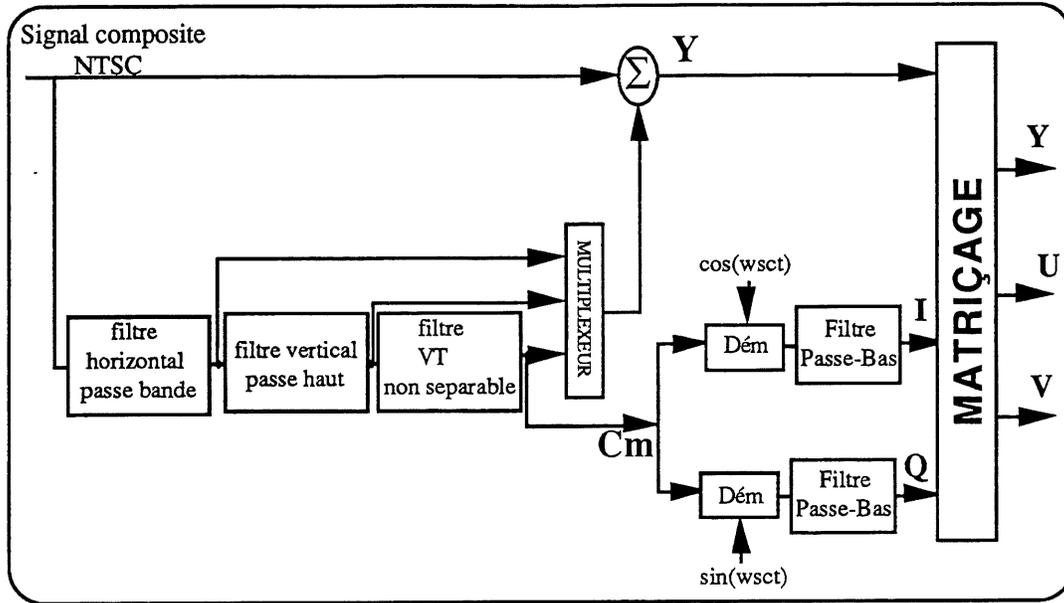


FIG 4.4. Configuration globale du décodage NTSC sélectif

Comme on peut le remarquer, cette structure renferme tous les filtres nécessaires pour un décodage 1D, 2D et 3D. En plus, cette structure renferme un multiplexeur dont le rôle est de valider le mode de décodage voulu selon la structure de l'encodeur.

Le décodage comporte aussi les deux opérations de démodulation et de matriçage pour retrouver les trois composantes standards de départ à savoir Y, U, V.

4.1.5. Conception des filtres

Tous les filtres qu'on proposera sont basés sur des techniques numériques. La conception des filtres a été faite soit par la méthode du « maximum flat », soit par la méthode du fenêtrage de Kaiser.

Quoi qu'il en soit, il n'est ni possible ni préférable d'avoir des filtres idéals (la gain est l'unité dans sa bande et zéro ailleurs). De pareils filtres sont non réalisables pour des contraintes de coût (l'ordre du filtre étant généralement très élevé). De plus les filtres à transition rapide introduisent généralement des artefacts non acceptables tels que la

Chapitre 4

répétition du détail d'une image autour d'elle même, ou ce qu'on appelle en anglais le phénomène de « *ringing* » ou encore l'effet *de Gibbs*.

Avant de passer à la conception de filtres voyons tout d'abord quelques propriétés du signal composite échantillonné.

4.1.5.1 Le signal NTSC échantillonné à 4fsc:

Le signal NTSC analogique peut être transformé en une séquence ou en un signal numérique par échantillonnage. La période d'échantillonnage est $T_s = 1/4f_{sc}$, ce qui correspond à une fréquence égale à 4fsc, une fréquence bel est bien supérieure à la limite de Nyquist [18].

D'après l'équation (2-9) le signal composite NTSC peut être aussi écrit sous la forme suivante:

$$E_c(t) = Y_a(t) + Cr_a(t) \quad (4-1)$$

Où $Y_a(t)$ et $Cr_a(t)$ sont respectivement la luminance et la chrominance analogique du signal composite.

Échantillonnée à 4fsc, l'expression de la chrominance devient alors:

$$Cr_a\left(\frac{n}{4f_{sc}}\right) = q_a\left(\frac{n}{4f_{sc}}\right)\sin\left(2\pi f_{sc}\frac{n}{4f_{sc}} + \phi\right) + i_a\left(\frac{n}{4f_{sc}}\right)\cos\left(2\pi f_{sc}\frac{n}{4f_{sc}} + \phi\right)$$

$$Cr_a\left(\frac{n}{4f_{sc}}\right) = q(n)\sin\left(\frac{\pi}{2}n + \phi\right) + i(n)\cos\left(\frac{\pi}{2}n + \phi\right)$$

Finalement, l'expression de la chrominance échantillonnée à 4fsc est donnée par:

$$Cr(n) = q(n)\sin\left(\frac{\pi}{2}n + \phi\right) + i(n)\cos\left(\frac{\pi}{2}n + \phi\right) \quad (4-2)$$

Le signal de luminance numérique quant à lui, est donné par:

$$Y(n) = Y_a\left(\frac{n}{4f_{sc}}\right) \quad (4-3)$$

Chapitre 4

Le signal NTSC numérique, qui est la somme de la luminance et de la chrominance numériques, s'écrit donc:

$$E_c(n) = Y(n) + C(n) \quad (4-5)$$

L'échantillonnage du signal composite à une fréquence de $4f_{sc}$ donne 910 pixels par ligne dans une trame de 262.5 lignes. Si on ne tient pas compte des périodes de suppression, on aura donc une partie active de 764 pixels par ligne et de 243 lignes par champ. Dans le cas du format 4 : 2 : 2 on aura tout juste 764 pixels actifs/ lignes dans une structure de 240 lignes.

Pour bien comprendre le principe de design des filtres multidimensionnels, le signal composite sera représenté par une séquence à deux dimensions $E_c(m,n)$ où:

n est un entier représentant le nombre de lignes par champ

m est entier représentant le nombre d'échantillons par ligne

Par exemple, $C(m+910n)$ est le $m^{\text{ième}}$ échantillon de la $n^{\text{ième}}$ ligne TV.

En se basant sur cette représentation, on peut dériver les expressions à deux dimensions de la luminance, de la chrominance et par conséquent, celle du signal composite.

Le signal de la chrominance est donné par:

$$\begin{aligned} C'(m,n) &= C(m+910n) = q(m+910n) \sin\left(\frac{\pi}{2}(m+910n) + \phi\right) + i(m+910n) \cos\left(\frac{\pi}{2}(m+910n) + \phi\right) \\ &= q'(m,n) \sin\left(\frac{\pi}{2}m + 455n\pi + \phi\right) + i'(m,n) \cos\left(\frac{\pi}{2}m + 455n\pi + \phi\right) \\ &= q'(m,n) \sin\left(\frac{\pi}{2}m + \pi n + \phi\right) + i'(m,n) \cos\left(\frac{\pi}{2}m + \pi n + \phi\right) \end{aligned} \quad (4-5)$$

Finalement la chrominance, dans une représentation bidimensionnelle, aura la forme de l'équation suivante:

$$C'(m,n) = q'(m,n) \sin\left(\frac{\pi}{2}m + \pi n + \phi\right) + i'(m,n) \cos\left(\frac{\pi}{2}m + \pi n + \phi\right) \quad (4-6)$$

Où

$$q'(m,n) = q(m + 910n)$$

$$i'(m,n) = i(m + 910n)$$

Le signal composite, dans une représentation bidimensionnelle, est donné par:

$$Ec(m,n) = C'(m,n) + Y'(m,n) \tag{4-7}$$

Où

$$Y'(m,n) = Y(m + 910n)$$

Puisque le standard prévoit une phase d'échantillonnage de \emptyset , la structure bidimensionnelle en treillis du signal composite NTSC échantillonné dans le domaine spatial est donnée par la figure suivante.

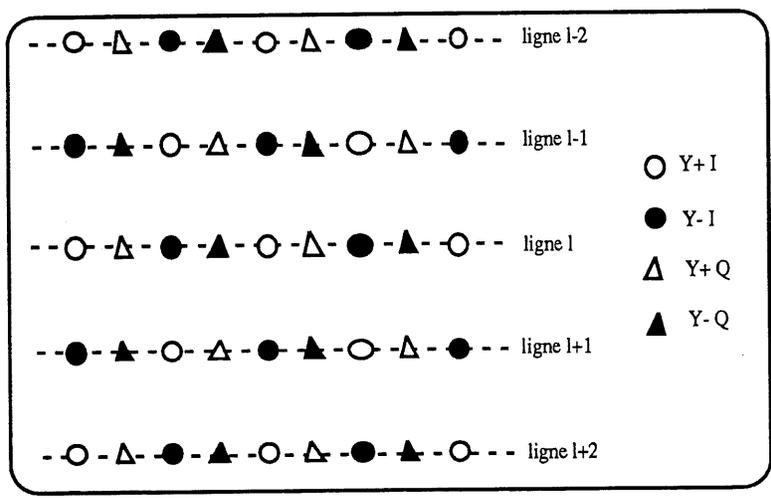


FIG 4.5. Structure bidimensionnelle en treillis du signal NTSC échantillonné

Dans le domaine vertical-temporel, d'après la règle de phase de la sous-porteuse, cette structure devient tel que l'indique la figure 2.6.

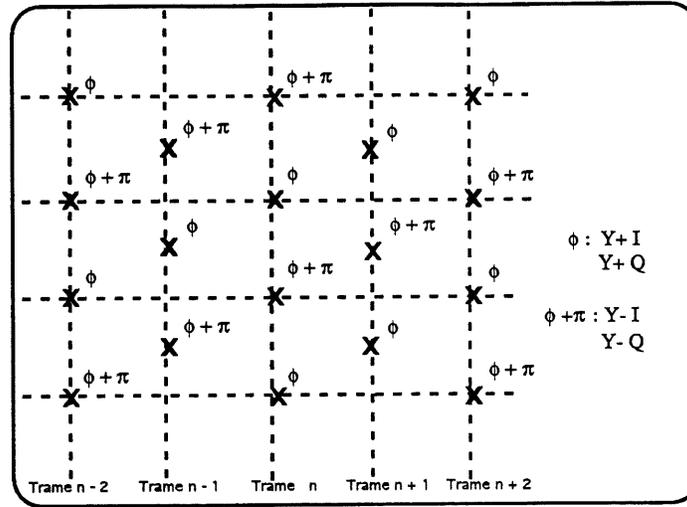


FIG 4.6. Structure bidimensionnelle (VT) en treillis du signal NTSC échantillonné

A. Filtre horizontal passe bande

Si on désigne par w_1 la fréquence normalisée au long de l'axe horizontal, l'expression du filtre est celle donnée par l'équation suivante :

$$H(w_1) = c_0 + 2 \sum_{n=1}^{(N-1)/2} c_n \cos(2.n.w_1) \quad (4-7)$$

où N est le nombre de coefficients du filtre

Selon les spécifications données, les coefficients quantifiés sont donnés dans le tableau 1 de l'annexe.

B. Filtre passe haut vertical:

Si on désigne par w_2 la pulsation le long de l'axe vertical, l'expression du filtre est celle donnée par l'équation suivante:

$$H(w_2) = c_0 + 2 \sum_{n=1}^{(N-1)/2} c_n \cos(n.w_2) \quad (4-8)$$

Les coefficients du filtre sont donnés au tableau 1 de l'annexe.

C. Filtre vertical temporel non séparable

Chapitre 4

w_3 , n_2 et n_3 désignent respectivement la fréquence normalisée le long de l'axe temporel, la cote verticale et la cote temporelle.

Le support de la réponse impulsionnelle est donné pour l'instant par la figure 4.7. On peut remarquer tout de suite que la structure renferme 2 champs de retard et 2 champs d'avance, 3 lignes de retard et 3 lignes d'avance. En tout, cela nécessite 4 trames et 4 lignes en mémoire.

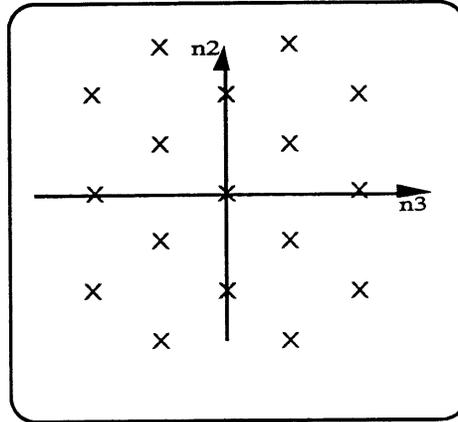


FIG 4.7. Support de la réponse impulsionnelle

La réponse impulsionnelle est donnée par l'expression:

$$\begin{aligned}
 h^*(n_2, n_3) = & h(n_2, n_3) + h(n_2 + 2, n_3) + h(n_2 - 2, n_3) + h(n_2 - 3, n_3 - 1) + h(n_2 - 1, n_3 - 1) \\
 & + h(n_2 + 1, n_3 - 1) + h(n_2 - 3, n_3 - 1) + h(n_2 - 2, n_3 - 2) + h(n_2, n_3 - 2) \\
 & + h(n_2 + 2, n_3 - 2) + h(n_2 - 3, n_3 + 1) + h(n_2 - 1, n_3 + 1) + h(n_2 + 1, n_3 + 1) \\
 & + h(n_2 + 3, n_3 + 1) + h(n_2 - 2, n_3 + 2) + h(n_2, n_3 + 2) + h(n_2 + 2, n_3 + 2)
 \end{aligned}
 \tag{4-9}$$

Par transformée en Z cette expression devient:

$$\begin{aligned}
 H^*(z_2, z_3) = & h(3, 2) + h(1, 2)z_2^2 + h(5, 2)z_2^{-2} + h(4, 1)z_2^{-1}z_3^{-1} + h(2, 1)z_2^1z_3^{-1} + h(6, 1)z_2^{-3}z_3^{-1} \\
 & + h(0, 1)z_2^3z_3^{-1} + h(5, 0)z_2^{-2}z_3^{-2} + h(3, 0)z_3^{-2} + h(1, 0)z_2^2z_3^{-2} + h(6, 3)z_2^{-3}z_3^1 \\
 & + h(4, 3)z_2^{-1}z_3^1 + h(2, 3)z_2^1z_3^1 + h(0, 3)z_2^3z_3^1 + h(5, 4)z_2^{-2}z_3^2 + h(3, 4)z_3^2 \\
 & + h(1, 4)z_2^2z_3^2
 \end{aligned}
 \tag{4-10}$$

Étant donné que la propriété de symétrie est vérifiée, la réponse impulsionnelle vérifie les relations suivantes:

Chapitre 4

$$\begin{aligned}
 h(4,1) &= h(2,3) \\
 h(2,1) &= h(4,3) \\
 h(1,2) &= h(5,2) = h(3,0) = h(3,4) \\
 h(5,0) &= h(1,0) = h(5,4) = h(1,4)
 \end{aligned}
 \tag{4-11}$$

L'équation du filtre devient alors:

$$\begin{aligned}
 H^*(z_2, z_3) &= h(3,2) + h(1,2).(z_2^2 + z_2^{-2}) + h(3,0).(z_3^2 + z_3^{-2}) + h(4,1).(z_2^{-1}z_3^{-1} + z_2^1z_3^1) \\
 &\quad + h(2,1).(z_2^1z_3^{-1} + z_2^{-1}z_3^1) + h(5,0).(z_2^{-2}z_3^{-2} + z_2^2z_3^{-2} + z_2^{-2}z_3^2 + z_2^2z_3^2) \\
 &\quad + h(6,1).(z_2^{-3}z_3^{-1} + z_2^3z_3^1) + h(0,1).(z_2^3z_3^{-1} + z_2^{-3}z_3^1)
 \end{aligned}
 \tag{4-12}$$

Finalement, dans le plan de Fourier, cette équation devient:

$$\begin{aligned}
 H^*(w_2, w_3) &= h(3,2) + 2.h(1,2).\cos(2w_2) + 2.h(3,0).\cos(2w_3) + 2.h(4,1).\cos(w_2 + w_3) \\
 &\quad + 2.h(2,1).\cos(w_2 - w_3) + 2.h(5,0).\cos(2.(w_2 + w_3)) + \cos(2.(w_2 - w_3))) \\
 &\quad + 2h(6,1).\cos(3w_2 + w_3) + 2h(0,1).\cos(3w_2 - w_3)
 \end{aligned}
 \tag{4-13}$$

Par la méthode du « maximum flat » on trouve la réponse impulsionnelle ci-dessous:

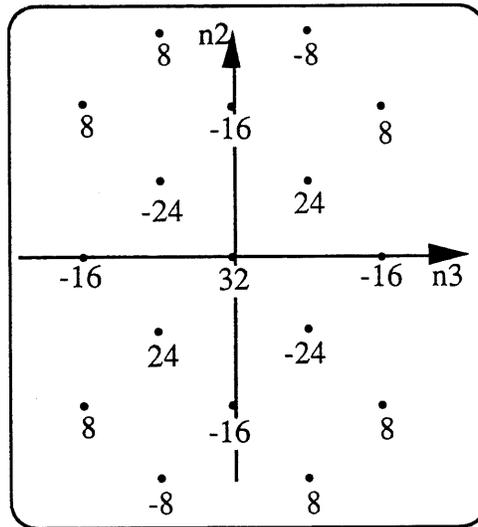


FIG 4.8. Réponse impulsionnelle du filtre

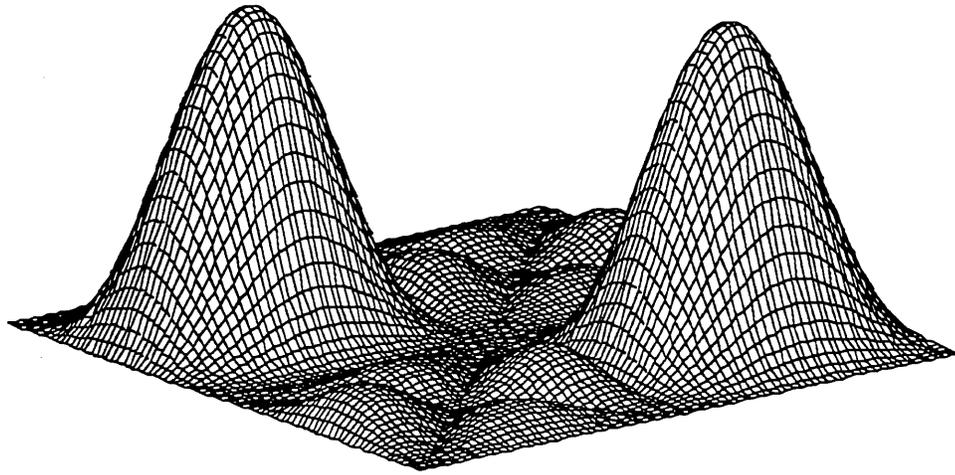


FIG 4.9. Réponse en fréquence du filtre
Caractéristique de chrominance

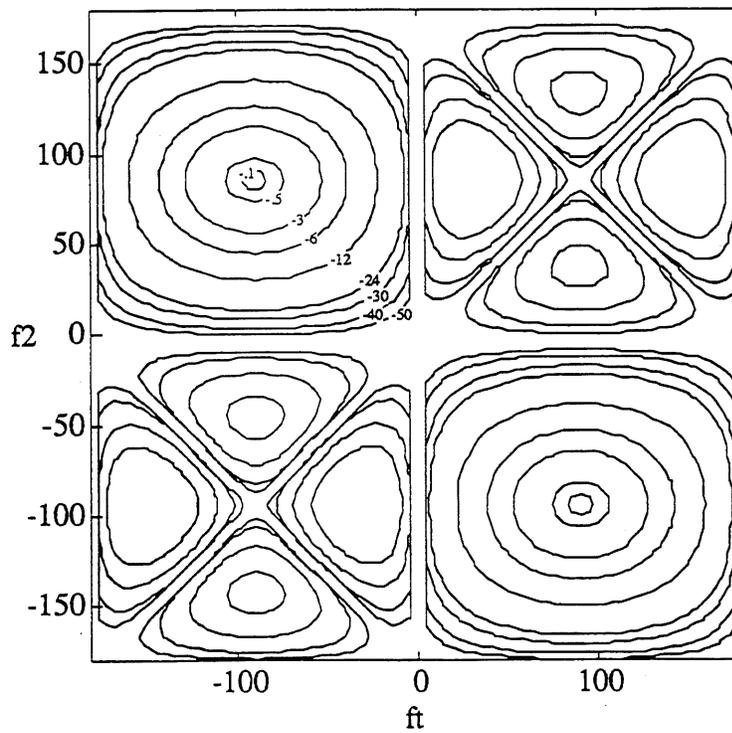


FIG.4.10. Contour de la réponse en fréquence

Chapitre 4

N.B. Tous les coefficients doivent être normalisés par 256.

Les figures 4.9 et 4.10 représentent respectivement la réponse fréquentielle du filtre et la représentation en contour de cette même réponse.

Bien que la réalisation de cette structure en temps réel ne constitue guère un problème, on essaiera par la suite de la simplifier pour éviter le problème du non-synchronisme entre l'image et le son que le retard de plus de 2 champs peut causer.

La version simplifiée du filtre est représentée sur la figure 4.11

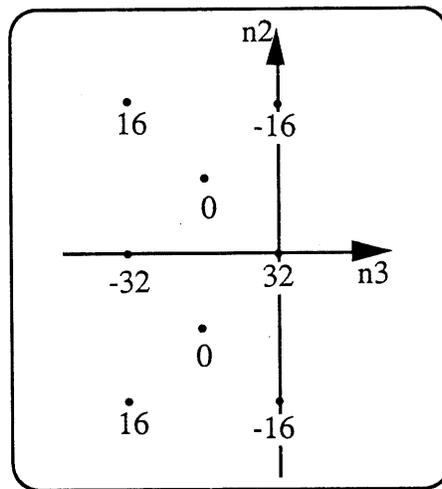


FIG 4.11. Réponse impulsionnelle du filtre non-causal

N.B. Les coefficients devront être normalisés par 128.

Comme on peut le remarquer, cette structure, qui n'est que la structure de la figure 4.8 repliée sur elle-même, contient uniquement 2 retards trames, un retard et une avance ligne, ce qui la rend beaucoup plus simple que cette dernière.

Bien que la version 4 retards de champ soit meilleure de point de vue performances théoriques, (atténuation dans la bande passante, dans la bande stop et la linéarité de la phase temporelle) au cours de tests subjectifs, on n'a remarqué aucune différence notable du point de vue taux de diachromie dans les images filtrées par ces deux filtres. Pour cela la version à 2 retards trames sera donc adoptée.

4.2. Décodage PAL

Pour le décodage PAL, on gardera les mêmes structures que celles développées pour le cas du décodage NTSC. Étant donné que les caractéristiques spectrales, dans les domaines horizontal-vertical, et vertical-temporel du système PAL sont différentes de celle du NTSC, les seules modifications effectuées n'affectent que le type de filtres utilisés. En effet, le filtre vertical passe haut du système NTSC va être remplacé par un vertical passe bande, et le filtre vertical-temporel, par un autre de structure différente.

Comme le spectre unidimensionnel des systèmes PAL et NTSC est essentiellement le même (la seule différence est que la sous-porteuse est située à 4.43 MHz au lieu de 3.58 MHz pour le système NTSC) et que la largeur de bande d'un filtre numérique est proportionnelle à la fréquence d'échantillonnage, on gardera le même filtre horizontal pour les deux systèmes.

La structure globale du décodeur sélectif PAL est donnée par la figure qui suit:

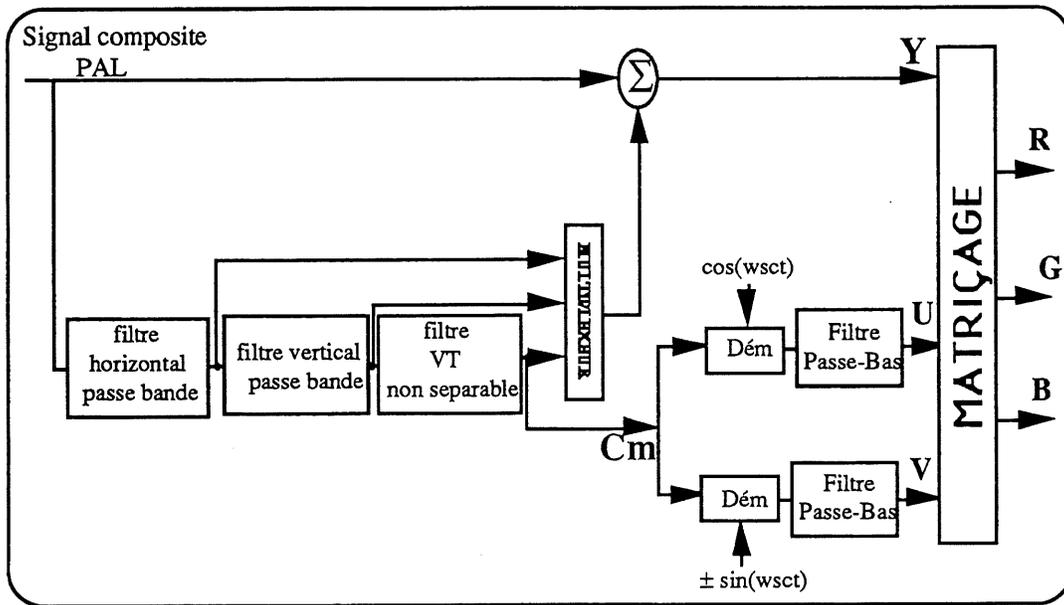


FIG 4.12. Décodeur sélectif PAL

Avant de passer à la conception de filtre voyons quelques propriétés caractéristiques du signal PAL composite échantillonné.

Chapitre 4

4.2.1. Signal PAL composite échantillonné à 4fsc

Le signal composite PAL analogique donné par l'expression (2-17) s'il est échantillonné à 4fsc résulte en 1135 pixels par ligne pour la partie active dans une structure de 312.5 lignes.

Après avoir fait un développement similaire à celui du NTSC, le signal numérique PAL, dans une représentation bidimensionnelle, est donnée par l'expression:

$$Ec(m,n) = C'(m,n) + Y'(m,n) \tag{4-14}$$

Où

$$Y'(m,n) = Y(m + 1135)$$

$$C'(m,n) = U(m,n) \sin\left(\frac{\pi}{2}m + \pi n\right) \pm V(m,n) \cos\left(\frac{\pi}{2}m + \pi n\right)$$

Les structures bidimensionnelles du signal PAL, dans les domaines spatial et vertical-temporel, sont représentées respectivement dans les figures 4.13 et 4.14

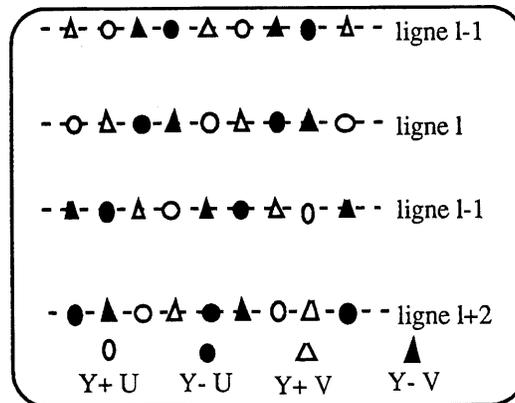


FIG 4.13. Structure spatiale du signal PAL échantillonné à 4fsc

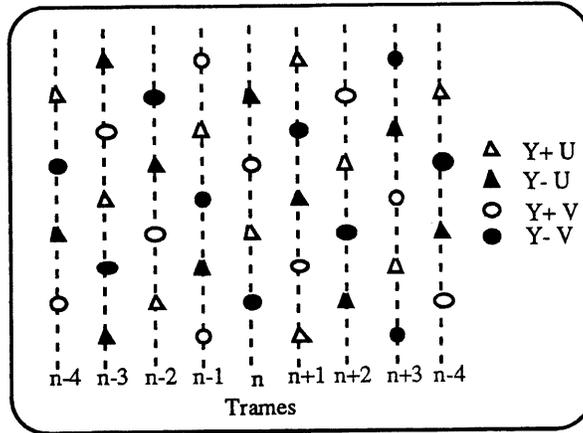


FIG 4.14. Structure bidimensionnelle (VT) du signal NTSC échantillonné à $4f_{sc}$

4.2.2. Conception des filtres

A. Le filtre passe bande horizontal: c'est le même que le filtre passe bande du système NTSC.

B. Le filtre passe bande vertical: l'expression de ce filtre est donnée par l'expression (4-2). Les coefficients sont dans le tableau 4.2 de l'annexe

B. Filtre vertical- temporel:

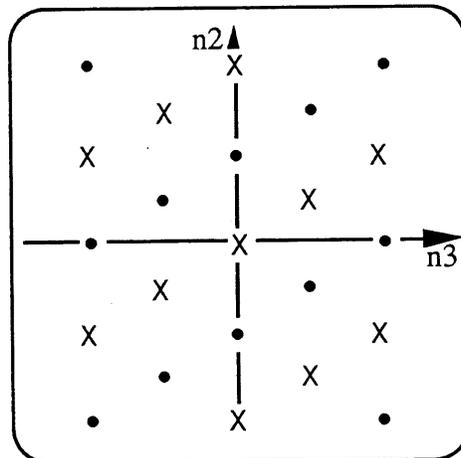


FIG 4.15. Support de la réponse impulsionnelle.

•: Échantillon non utilisé

Chapitre 4

La réponse impulsionnelle est donnée par l'expression:

$$\begin{aligned}h^*(n_2, n_3) = & h(n_2, n_3) + h(n_2 - 4, n_3) + h(n_2 + 4, n_3) + h(n_2 - 3, n_3 - 1) + h(n_2 + 1, n_3 - 1) \\ & + h(n_2 - 2, n_3 - 2) + h(n_2 + 2, n_3 - 2) + h(n_2 - 1, n_3 + 1) + h(n_2 + 3, n_3 + 1) \\ & + h(n_2 - 2, n_3 + 2) + h(n_2 + 2, n_3 + 2).\end{aligned}$$

(4-15)

Par transformée en Z cette expression devient:

$$\begin{aligned}H^*(z_1, z_2) = & h(4, 2) + h(8, 2)z_2^{-4} + h(0, 2)z_2^4 + h(7, 1)z_2^{-3}z_3^{-1} + h(3, 1)z_2^1z_3^{-1} + h(6, 0)z_2^{-2}z_3^{-2} \\ & + h(2, 0)z_2^2z_3^{-2} + h(5, 3)z_2^{-1}z_3^1 + h(1, 3)z_2^3z_3^1 + h(6, 4)z_2^{-2}z_3^2 + h(2, 4)z_2^2z_3^2\end{aligned}$$

(4-16)

Comme la propriété de symétrie est vérifiée, la réponse impulsionnelle vérifie les conditions suivantes:

$$\begin{aligned}h(8, 2) &= h(0, 2) \\ h(7, 1) &= h(1, 3) \\ h(2, 0) &= h(6, 4) \\ h(3, 1) &= h(5, 3) = h(6, 0) = h(2, 4)\end{aligned}\tag{4-17}$$

L'équation du filtre devient alors:

$$\begin{aligned}H^*(w_2, w_3) = & h(4, 2) + 2.h(8, 2).\cos(4w_2) + 2.h(7, 1).\cos(2w_2 + w_3) \\ & + 2.h(3, 1).\cos(w_2 - w_3) + 2.h(6, 0).\cos(2w_2 + 2w_3) \\ & + h(2, 0).\cos(2w_2 - w_3)\end{aligned}\tag{4-18}$$

Par la méthode du « maximum » flat on trouve la réponse impulsionnelle ci-dessous.

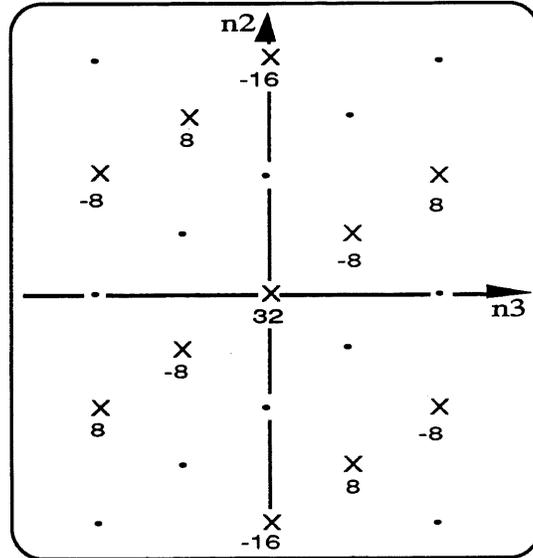


FIG 4.16. Réponse impulsionnelle du filtre

Les figures 4.18 et 4.19 représentent respectivement la réponse fréquentielle du filtre et la représentation en contour de cette même réponse.

Bien que la réalisation de cette structure en temps réel ne constitue guère un problème, on a voulu que les deux systèmes aient le même degré de complexité c'est pourquoi on donne aussi une version simplifiée du filtre qui ne comporte que 2 retards trames. Cette version est donnée par la figure qui suit.

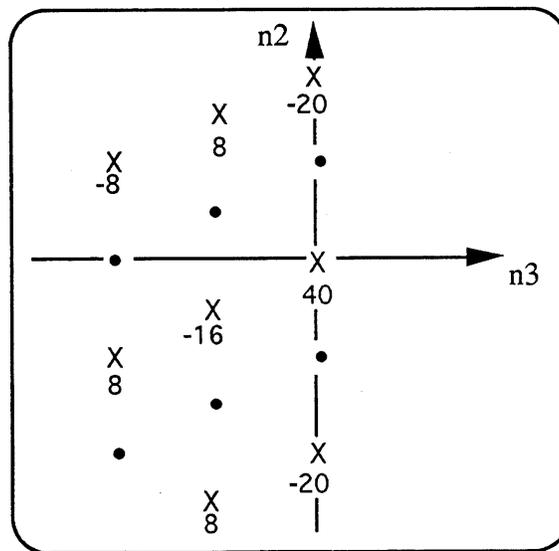


FIG 4.17. Réponse impulsionnelle : version simplifiée

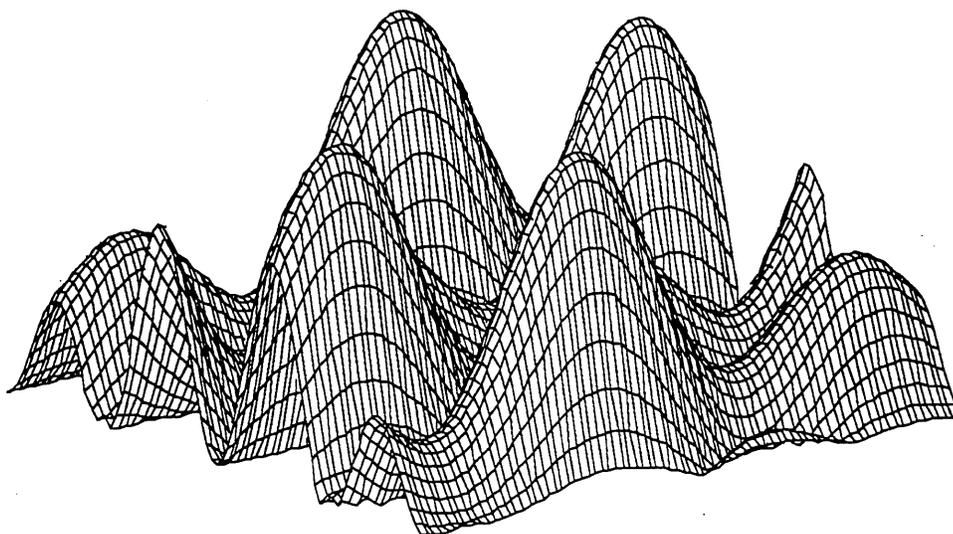


FIG 4.18. Réponse en fréquence du filtre caractéristique de chrominance

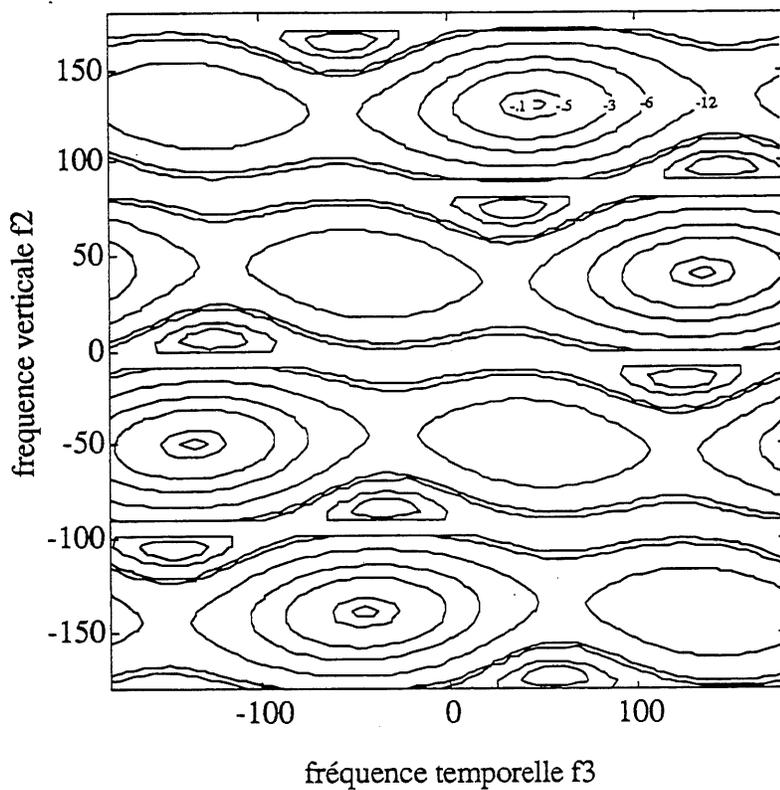


FIG 4.19. Contour de la réponse en fréquence

Chapitre 4

4.3. Limitations de l'approche du décodage sélectif:

L'incompatibilité entre la structure du codeur et du décodeur veut dire tout simplement que les deux structures ne sont pas identiques, c'est à dire, si le signal PAL ou NTSC à l'entrée du décodeur est délivré par un encodeur 1D alors qu'on valide le mode de décodage 2D ou s'il est délivré par un encodeur 2D et qu'on fait un décodage 3D. Lors des simulations on a remarqué que cette incompatibilité introduit des effets assez remarquables de luminance.

Même le multiplexeur qui permet à l'utilisateur de transiter d'un mode de décodage à un autre peut causer des effets ennuyeux dus à ce changement.

Par contre, quel que soit la structure de l'encodeur 1D, 2D ou 3D, si le signal est décodé 1D aucun effet de luminance croisée n'apparaîtra. Pour cette raison, on adoptera par défaut ce type de décodage.

La perte de résolution dans la composante de luminance que ce type de décodage engendre peut être corrigée par la mise en série d'un algorithme de compensation de résolution appelé aussi isoluminance [22].

Développé au sein de notre groupe de recherche, cet algorithme permet le rehaussement de la résolution de l'image. Il consiste à amplifier les composantes de haute fréquence que contient l'image. Cette technique s'avère efficace du fait qu'elle est simple et facilement réalisable sur des circuits électroniques.

Les performances de ces approches sont discutées dans le chapitre 6.

CHAPITRE 5

DÉCODAGE NUMÉRIQUE : APPROCHE ADAPTATIVE

SYSTÈME NTSC

Les approches d'encodage et de décodage présentées dans le chapitre 4, qui touchaient et au codeur et au décodeur, permettent d'améliorer les performances des systèmes NTSC et PAL, et ce en particulier pour le cas d'encodage et de décodage 2D et 3D.

Malheureusement, les encodeurs actuellement en service sont, pour la plupart, des cas du type OD. En plus, l'existence d'un énormes archives NTSC et PAL, encodées OD, nécessite le développement d'un décodeur qui devra séparer les composantes du signal composite sans engendrer d'artefacts d'interférence.

Dans ce chapitre, on présentera une technique de décodage adaptative, qui, indépendamment de la structure de l'encodeur, devra être capable de séparer la luminance et la chrominance quasiparfaitement. La structure qu'on présente est relative au système NTSC cependant l'approche est facilement généralisable pour le cas du système PAL.

Cette technique est similaire à celle proposée par Teichner [14][15], cependant on propose deux différences notables :

- une détection horizontale/verticale élaborée

- un filtrage en série de la chrominance.

chapitre 5

La dernière technique a pour but d'introduire une bande de garde entre la luminance et la chrominance, ce qui a pour effet de réduire d'une façon significative les effets de diachromie.

5.1. Principe de l'approche

Cette approche contient deux niveaux d'adaptations :

A. Niveau d'adaptation selon la détection de mouvement

Étant donné que, pour le cas des images fixes, un filtre en peigne à retard trame est à lui seul capable de séparer la luminance et la chrominance sans aucun artefact d'interférence tout en préservant la pleine résolution de l'image[1], dans la structure qu'on propose, on utilisera un détecteur de mouvement dont le rôle essentiel est de valider l'action du filtre en peigne pour le cas des parties fixes de l'image.

B. Un niveau d'adaptation selon les transitions verticales

Pour éviter le problème de luminance croisée, en particulier le « dot crawl », qui généralement affecte les transitions de hautes fréquences verticales de chrominance si elles sont filtrées verticalement dans la structure de l'approche adaptative, on utilisera un circuit dont le rôle est de détecter la présence de ces transitions dans la composante de chrominance et d'inhiber l'action du filtre vertical en ces endroits pour favoriser l'action du filtre horizontal.

De plus, pour bien éviter les problèmes de diachromie, qui généralement affectent les transitions diagonales de la composante de luminance, on ajoute, en série, dans la voie de séparation de chrominance en plus du filtre passe bande horizontal et du filtre vertical un filtre en losange. Ce filtre permet d'éliminer les artefacts résiduels de diachromie surtout si le signal composite n'est pas pré-filtré. La figure 5.1 représente la structure détaillée de cette approche.

chapitre 5

On remarque tout de suite que cette structure comporte les éléments suivants :

Pour la luminance on trouve en parallèle:

- un filtre horizontal passe bande
- un filtre vertical passe haut

(Ces deux filtres sont les mêmes que ceux développés pour le cas du décodage du système NTSC).

- un filtre temporel en peigne (les coefficients de ce dernier sont donnés au tableau 3 de l'annexe).

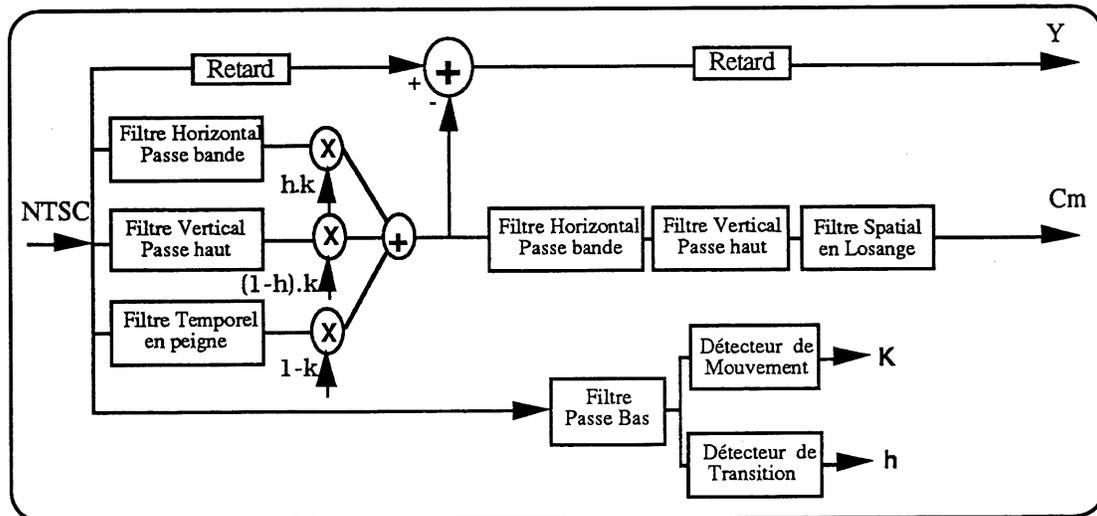


FIG 5.1. Configuration globale de l'approche adaptative

- le détecteur de mouvement
- le détecteur des transitions verticales

Pour la séparation de la chrominance, en plus des trois filtres ci-dessus mentionnés on trouve en connexion série :

- le même filtre horizontal passe bande

chapitre 5

- le même filtre vertical passe haut
- un filtre en losange qu'on détaillera plus tard

Ensemble, ces filtres forment un filtre en diamant totalement séparable (évoqué dans [10][23]). Ce dernier permet une bonne extraction de la composante de chrominance.

Dans ce qui suit, on détaillera les modules mentionnés en rubrique.

5.1.1. Module de détection de mouvement

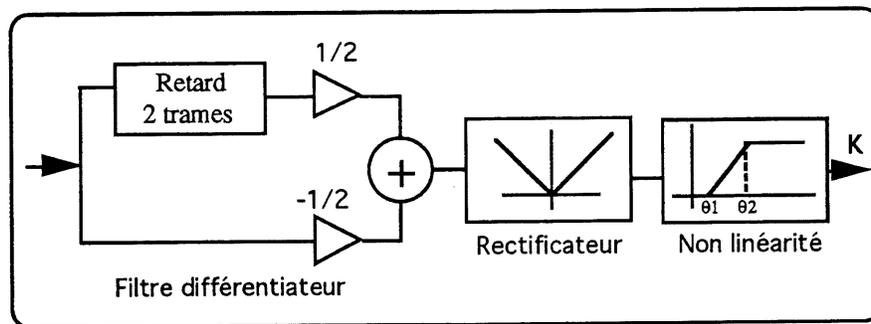


FIG 5.2. Détecteur de mouvement

La structure détaillée de ce détecteur est représentée dans la figure 5.2. Ce module comporte :

A. Un filtre différentiateur

La plus simple façon de détecter le changement dans une image et de comparer les signaux espacés d'une trame. En effet les composantes d'une scène sont supprimées, tandis que les portions de fréquence appartenant aux parties de l'image où il y avait changement, dominent le signal différence.

B. Un rectificateur:

Ce rectificateur sert à éliminer le signe négatif des signaux de différentiation.

C. Une non-linéarité

Le signal à la sortie du rectificateur, sera transformé par la non-linéarité en un paramètre K. Cette constante contrôlera le processus de commutation. Les seuils Θ_1 et Θ_2 sont déterminés expérimentalement. Pour notre cas, $\Theta_1 = 3$ et $\Theta_2 = 10$.

5.1.2. Détecteur des transitions verticales

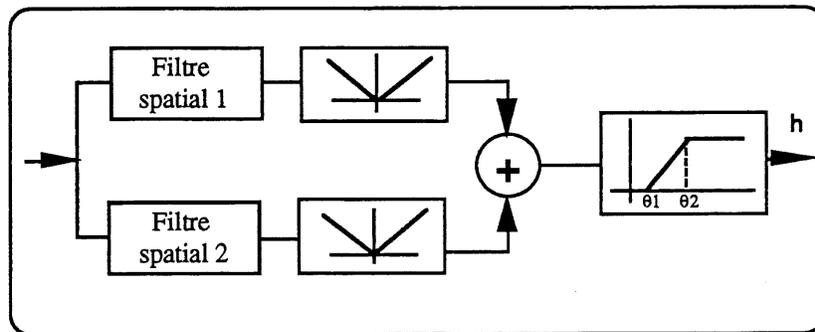


FIG 5.3. Détecteur de transition. Schéma bloc

Ce détecteur contient :

A. Deux filtres spatiaux dont le rôle est de détecter les transitions verticales dans l'image . Les supports de leurs réponses impulsionnelles sont:

$$\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{Réponse impulsionnelle du filtre 1}$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \text{Réponse impulsionnelle du filtre 2}$$

Le premier filtre sert à détecter les transitions ou encore les lignes horizontales. Le deuxième sert à détecter les transitions verticales.

N.B. Tous les coefficients sont normalisés par 8.

chapitre 5

B. Deux rectificateurs

C. Une non-linéarité : le signal de sortie du rectificateur est transformé par la non-linéarité en un paramètre h qui commandera le processus de commutation. Les seuils Θ_1 et Θ_2 sont déterminés expérimentalement. Pour notre cas, ils sont respectivement égaux à 7 et 15.

5.1.3. Filtre en losange

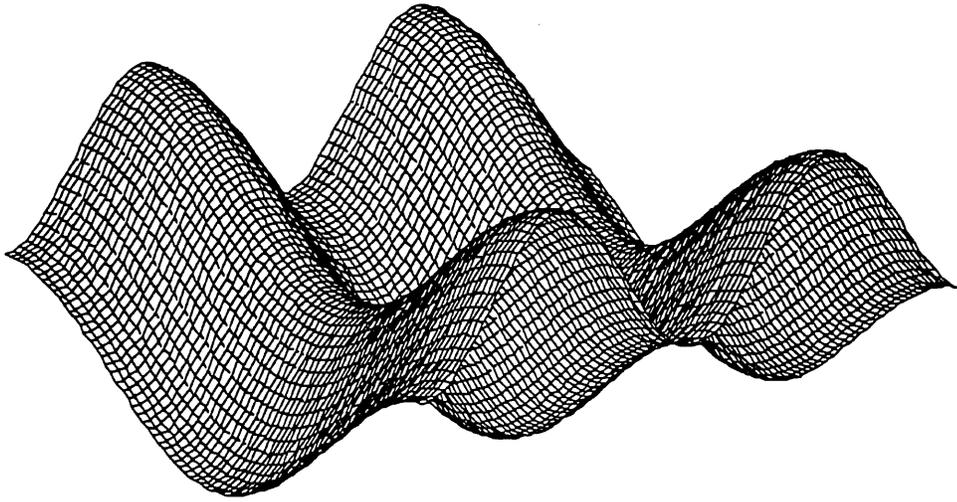
Ce filtre a été évoqué pour la première fois par Dubois dans [9]. Ce filtre tient compte de la forme spectrale des composantes de chrominance dans le domaine des fréquences spatiales qui ont la forme d'un losange pour bien les extraire.

La réponse impulsionnelle de ce filtre est donnée par

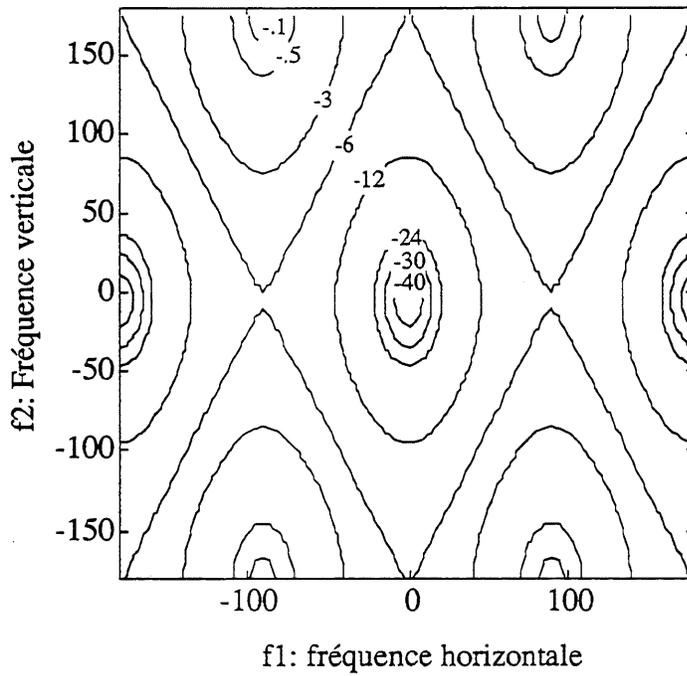
$$\begin{array}{ccccc} 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \end{array}$$

On normalise le gain par 8.

La réponse en fréquence de ce filtre est représentée dans la figure 5.4



5.4. Réponse en fréquence du filtre
Caractéristique de chrominance



5.5. Contour de la réponse en fréquence

chapitre 5

5.1.4. Principe de fonctionnement

Le principe du fonctionnement du décodeur est simple à expliquer. On considère d'abord les deux cas limites suivants :

- Pour le cas d'images fixes, le détecteur de mouvement délivrera un paramètre $k = 0$, une constante bien représentative de l'absence du mouvement, cette même constante permettra d'inhiber, à la fois, l'action du filtre horizontal et du filtre vertical, et par contre permettra au filtre temporel en peigne d'agir seul.

- Dans le cas d'images en mouvement, le détecteur délivrera un paramètre $k = 1$. Par conséquent, l'action du filtre temporel en peigne est inhibée. Il y aura un compromis seulement entre le filtre horizontal et le filtre vertical.

Si dans ce même cas, il y a présence de transitions verticales, le détecteur délivrera un paramètre $h=1$, ce qui annulera l'action du filtre vertical, qui si utilisé dans ce cas, engendrerait l'artefact de luminance croisée qui se manifeste par du «dot crawl». Dans le cas contraire, c'est à dire $h=0$, c'est le filtre horizontal qui sera inhibé et le filtre vertical entrera en action.

En réalité, le processus n'est pas du type tout ou rien. Dans le but d'éviter une commutation brusque entre les différents modes de décodage (temporel versus spatial ou horizontal versus vertical qui peut causer des artefacts résiduels), nous utilisons une adaptation douce qui varie graduellement en fonction des détections temporelle, horizontale et verticale.

Cette approche permet, non seulement une bonne séparation de la luminance et la chrominance mais aussi de préserver la résolution de l'image (voir chapitre 6.)

Page 74 manquante dans notre exemplaire.

CHAPITRE 6

RÉSULTATS DES SIMULATIONS ET INTERPRÉTATIONS

Les simulations d'encodage et de décodage relatives aux deux systèmes NTSC et PAL ont été réalisées sur les trois séquences d'images suivantes :

- Flower.ccir : La photo 1 en annexe, est la composante de luminance. La photo 2 comprend les deux composantes de chrominance. Cette séquence est jugée très critique pour l'opération encodage-décodage du fait qu'elle renferme beaucoup de transitions diagonales qui sont la cause principale de l'artefact de diachromie.
- Mobile.ccir : La photo 3 en annexe, est la composante de luminance, tandis que la photo 4 comprend les deux composantes de chrominance. Cette séquence est très critique étant donné qu'elle renferme des transitions de couleurs très rapides dans toutes les directions qui est la cause principale de l'artefact de luminance croisée le « dot crawl » et « les hanging dots ».
- Zone.ccir : La photo 5 est aussi connue sous le nom de mire de Fresnel ou, en anglais « zone plate ». Cette image est généralement utilisée pour avoir une idée des performances du point de vue résolution. Elle contient toutes les fréquences spatiales possibles qu'une image peut contenir. Elle peut être, comme on va le voir, une représentation du spectre bidimensionnel spatial d'un système donné.

Ces trois séquences constituent donc notre point de référence.

On discutera maintenant des performances relatives à l'approche fixe et compatible et à l'approche adaptative qu'on a présentées respectivement dans le chapitre 4 et le chapitre 5.

Chapitre 6

6.1. Approches fixes et compatibles

6.1.1. Système NTSC

6.1.1.1. Configuration 1D-1D

A. Composante de luminance

On remarque que l'artefact de luminance croisée est totalement disparu. La photo 11 nous montre le résultat (multiplié par 15) d'un décodage 1D d'un signal composite encodé 1D et ne contenant que la composante de chrominance. Le problème de luminance croisée, malgré cette multiplication, est à peine visible.

Cependant, la résolution de la luminance est très réduite. La photo 10 est une bonne indication de cette perte. On peut remarquer aussi cette perte sur les photos 6 et 8. L'amélioration de la résolution est rendue possible grâce à l'algorithme de l'isoluminance comme on l'a mentionné au chapitre 4. Les photos 6 et 8, traitées par cet algorithme se retrouvent avec une résolution nettement améliorée (voir les photos 13 et 14 en annexe).

B. Composante de chrominance

L'artefact de diachromie est totalement disparu. La photo 12 nous montre le résultat d'un décodage 1D d'un signal composite encodé 1D et qui ne contient que la composante de luminance.

La résolution de la composante de chrominance seule (résolution verticale) a été jugée moyenne par rapport à celle de l'image originale. Si on regarde l'image formée par la composante de luminance et celle de la chrominance on constate que la perte de résolution est non appréciable par l'oeil.

Chapitre 6

6.1.1.2. Configuration 2D-2D

A. Composante de luminance

Au point de vue de l'artefact de luminance croisée (photo24), les résultats obtenus au moyen de la configuration 2D-2D, sont presque identiques à ceux obtenus par la configuration 1D-1D. Même s'il reste, dans quelques cas, des artefacts résiduels de luminance croisée (généralement ses résidus ne sont pas ennuyeux), on peut au moyen d'un filtre vertical qui renferme quelques lignes de retard en plus, éliminer ces artefacts résiduels.

Cependant, la résolution obtenue est nettement meilleure que celle obtenue par la configuration 1D-1D. La photo 23 de la mire de Fresnel est une indication de cette amélioration. D'après cette photo, on remarque qu'on couvre presque tout l'espace des fréquences spatiales. Les seules régions de fréquence qu'on ne couvre pas sont celles occupées par les quatre satellites de chrominance. Sur la photo 23 ces zones apparaissent comme quatre trous ce qui représente bien le spectre bidimensionnel du signal composite NTSC.

B. Composante de chrominance:

Au point de vue de l'artefact de diachromie on obtient des résultats presque similaires à ceux obtenus par la configuration 1D-1D. Il y a tout de même un peu plus de diachromie (Photo 25), mais ceci reste toujours inappréciable par l'oeil.

6.1.1.3. Configuration 3D-3D

La résolution obtenue par cette configuration et celle de l'image originale sont identiques (voir photo33). À l'aide de cette configuration, on couvre carrément tout l'espace des fréquences spatiales (sur cette photo on ne voit aucun trou). Aussi les photos 31 et 32 sont de résolutions identiques à celle des images originales 1 et 3.

Chapitre 6

Au point de vue de la luminance croisée, la configuration 3D-3D permet d'obtenir des images sans cet artefact.

B. Composante de chrominance

Le problème de diachromie est complètement supprimé par cette configuration. La photo 34 qui est le résultat (multipliée par 15) du décodage d'un signal NTSC composite ne contenant que le signal de luminance, ne présente aucun artefact de diaphotie.

6.1.2. Système PAL

6.1.2.1. Configuration 1D-1D

A. Composante de luminance

On remarque que l'artefact de luminance croisée est totalement disparu. La photo 19 nous montre le résultat (multiplié par 15) d'un décodage 1D d'un signal composite encodé 1D et ne contenant que la composante de chrominance. Le problème de luminance croisée, malgré cette multiplication, n'est pas visible.

Cependant, la résolution de la luminance est très réduite. La photo 18 est une bonne indication de cette perte. Comparée à la résolution obtenue dans le système NTSC, la résolution obtenue dans le système PAL est nettement meilleure (voir les photos 10 et 18). Ceci est évident, car dans le système PAL, la sous porteuse est placée à 4,43 MHz et non à 3,58 MHz. Pour le cas du PAL on n'a pas besoin de compenser cette perte de résolution. La photo 15 qui résulte d'un décodage 1D si elle est traitée avec l'algorithme de l'isoluminance, se retrouve tout de même avec une résolution améliorée (voir photo 17).

Chapitre 6

B. Composante de chrominance:

L'artefact de diachromie est totalement disparu. La photo 20 nous montre le résultat d'un décodage 1D d'un signal composite encodé 1D et qui ne contient que la composante de luminance.

La résolution de la composante de chrominance seule (résolution verticale) a été jugée moyenne par rapport à l'original. Si on regarde l'image formée par la composante de luminance et celle de la chrominance, on constate que cette perte de résolution est acceptable par l'oeil.

6.1.2.2. Configuration 2D-2D

A. Composante de luminance

Au point de vue de l'artefact de luminance croisée (photo 29), les résultats obtenus au moyen de la configuration 2D-2D sont presque identiques à ceux obtenus par la configuration 1D-1D.

Cependant, la résolution est nettement meilleure que celle obtenue par la configuration 1D-1D. La photo 28 de la mire de Fresnel est une bonne indication de cette amélioration. D'après cette photo, on remarque qu'on couvre presque tout l'espace des fréquences spatiales. Les seules régions de fréquence qu'on ne couvre pas sont celles occupées par les huit satellites de chrominance. Sur la photo 28, ces zones apparaissent comme huit trous ce qui représente bien le spectre bidimensionnel du signal composite PAL.

Chapitre 6

B. Composante de chrominance:

Au point de vue de l'artefact de diachromie, on obtient des résultats presque similaires à ceux obtenus par la configuration 1D-1D. Il y a tout de même un peu plus de diachromie (photo 30), mais ceci reste toujours inappréciable par l'oeil.

6.1.2.3. Configuration 3D-3D:

La résolution obtenue par cette configuration et celle de l'image originale sont identiques. Comme le prouvent les photos 36, 37 et 38. D'après la photo 38, on remarque, à l'aide de cette configuration, on couvre carrément tout l'espace des fréquences spatiales (sur cette photo on ne voit aucun trou).

Du point de vue de la luminance croisée, la configuration 3D-3D permet d'obtenir des images sans cet artefact.

B. Composante de chrominance

Le problème de diachromie est complètement supprimé par cette configuration. La photo 36, qui est le résultat (multiplié par 15) de décodage d'un signal NTSC composite ne contenant que le signal de luminance, ne présente aucun artefact de diachromie.

Les résultats obtenus par les trois configurations ci-dessus sont nettement meilleurs, à tous les niveaux, que ceux obtenus par un encodeur et un décodeur standards.

6.2. Approche adaptative

A. Composante de luminance

La résolution obtenue au moyen de cette approche est très bonne. La photo 45 prouve qu'avec cette approche on couvre tout l'espace des fréquences spatiales. En effet cette photo ne contient aucun trou.

Chapitre 6

Au point de vue de la luminance croisée, cette approche permet d'obtenir des images sans aucun effet de luminance croisée. La photo 46 nous montre le résultat (multiplié par 15) du décodage d'un signal composite constitué uniquement par le signal de chrominance et qui est engendré par un encodeur standard OD. Malgré la multiplication par 15, les effets de luminance croisée restent quand même à peine visible.

B. Composante de chrominance:

Le problème de diachromie est nettement diminué par cette approche (photo 47) La seule chose qu'on peut reprocher à cette approche c'est la perte de résolution dans les composantes de chrominance. Dans pas mal de cas d'images mobiles, cette perte de résolution dans les composantes de chrominance est appréciable.

CONCLUSION

La présente recherche a montré, hors de tout doute, que l'utilisation des techniques numériques multidimensionnelles de traitement de signaux permet de réduire, d'une manière significative, les artefacts d'interférence entre les deux signaux de luminance et de chrominance.

À partir de la discussion des propriétés des signaux composites NTSC et PAL ainsi que de leurs caractéristiques spectrales, il est devenu clair que la séparation des deux composantes de luminance et de chrominance est possible étant donné deux conditions :

1. Premièrement, l'espace de fréquence à trois dimensions doit être divisé selon la disposition des composantes de luminance et de chrominance ce qui implique la nécessité d'un pré-traitement à l'encodeur.

2. Deuxièmement, l'utilisation de filtres appropriés.

La première condition est à l'origine de l'approche fixe et compatible proposée dans le chapitre 4. Cette approche est non seulement simple vu qu'elle fait intervenir des filtres facilement réalisables sur des circuits électroniques, mais aussi efficace du fait qu'elle nous permet d'avoir des images de qualité comparable aux qualités des images dans les studios (surtout pour le cas d'un traitement à trois dimensions à l'encodeur et au décodeur).

Cependant, le filtrage à l'encodeur bien qu'il soit non coûteux, reste quand même non pratique aux standards composites. ce fait nous a conduit à proposer, dans le chapitre 5, une autre approche adaptative.

Cette approche adaptative, qui permet de contrôler la quantité d'interférence entre les signaux de luminance et de chrominance en tenant compte du mouvement et aussi des transitions verticales, permet de réduire d'une manière significative ces interférences tout en préservant la pleine résolution.

ANNEXE

Annexe

1. Coefficients des filtres de la figure 3.4

a. Filtre Passe Bas:

$$[-5 \ 0 \ 37 \ 64 \ 37 \ 0 \ -5] / 64$$

b. Filtre Polyphase:

0,02857143							
0,02853235	-0,00076675	0,00035279	-0,00020029	0,00011823	-0,00006790	0,00003608	-0,00001652
0,02841532	-0,00148327	0,00069039	-0,00039295	0,00023195	-0,00013302	0,07045222	-0,00003206
0,02822091	-0,00214766	0,00101069	-0,00057666	0,00034037	-0,00019488	0,00010290	-0,00004655
0,02795012	-0,00275838	0,00131179	-0,00075020	0,00044273	-0,00025309	0,00013321	-0,05988939
0,02760431	-0,00331421	0,00159197	-0,00091246	0,00053839	-0,00030727	0,00016120	-0,00007202
0,02718520	-0,00381427	0,00184976	-0,00106247	0,00062676	-0,00035711	0,00018673	-0,00008289
0,02669491	-0,00425799	0,00208390	-0,00119937	0,00070733	-0,00040232	0,00020967	-0,00009246
0,02613589	-0,00464518	0,00229332	-0,00132245	0,00077966	-0,44269107	0,00022992	-0,00010071
0,02551092	-0,00497594	0,00247721	-0,00143111	0,00084341	-0,00047803	0,00024742	-0,10762157
0,02482312	-0,00525071	0,00263497	-0,00152490	0,00089832	-0,00050822	0,00026212	-0,00011321
0,02407590	-0,00547023	0,00276620	-0,00160349	0,00094418	-0,00053317	0,00027400	-0,00011749
0,02327294	-0,00563554	0,00287074	-0,00166670	0,00098091	-0,00055285	0,00028307	-0,00012048
0,02241819	-0,00574799	0,00294864	-0,00171447	0,00100847	-0,00056727	0,00028938	-0,00012223
0,02151584	-0,00580919	0,00300012	-0,00174685	0,00102692	-0,00057650	0,00029297	-0,00012279
0,02057027	-0,00582100	0,00302565	-0,00176405	0,00103637	-0,00058062	0,00029393	-0,00012221
0,01958605	-0,00578553	0,00302584	-0,00176636	0,00103703	-0,00057978	0,00029235	-0,00012056
0,01856791	-0,00570511	0,00300150	-0,00175421	0,00102916	-0,00057416	0,00028836	-0,00011792
0,01752071	-0,00558228	0,00295360	-0,00172811	0,00101308	-0,00056397	0,00028209	-0,00011436
0,01644938	-0,00541976	0,00288328	-0,00168869	0,00098918	-0,00054945	0,00027368	-0,00010998
0,01535895	-0,00522043	0,00279178	-0,00163667	0,00095790	-0,00053088	0,00026332	-0,00010485
0,01425448	-0,00498730	0,00268051	-0,00157283	0,00091973	-0,00050856	0,00025116	-0,00009908
0,01314103	-0,00472350	0,00255097	-0,00149805	0,00087519	-0,00048281	0,00023740	-0,00009275
0,01202365	-0,00443227	0,00240477	-0,00141326	0,00082486	-0,00045396	0,00022221	-0,00008596
0,01090735	-0,00411691	0,00224358	-0,00131944	0,00076933	-0,00042237	0,00020581	-0,00007881
0,00979705	-0,00378075	0,00206915	-0,00121763	0,00070923	-0,00038841	0,00018839	-0,00007138
0,00869756	-0,00342716	0,00188330	-0,00110889	0,00064519	-0,00035246	0,00017014	-0,00006377
0,00761358	-0,00305952	0,00168786	-0,00099432	0,00057788	-0,00031488	0,00015126	-0,00005607
0,00654963	-0,00268117	0,00148469	-0,00087503	0,00050796	-0,00027606	0,00013196	-0,00004836
0,00551006	-0,00229541	0,00127566	-0,00075213	0,00043609	-0,00023637	0,00011243	-0,00004071
0,00449903	-0,00190549	0,00106263	-0,00062673	0,00036294	-0,00019619	0,00009284	-0,00003320
0,00352044	-0,00151457	0,00084742	-0,00049994	0,00028914	-0,00015587	0,00007337	-0,00002591
0,00257798	-0,00112570	0,00063184	-0,00037284	0,00021535	-0,00011576	0,00005420	-0,00001889
0,00167504	-0,00074181	0,00041763	-0,00024648	0,00014217	-0,07620755	0,00003549	-0,00001220
0,00081477	-0,00036569	0,00020648	-0,00012188	0,00007020	-0,00003752	0,00001738	-0,00000589
0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000

Annexe

2. Coefficients des filtres de la figure 3.7:

a. Filtre polyphase:

0,01694915							
0,01694099	-0,00027335	0,00012516	-0,00007098	0,00004190	-0,00002408	0,00001281	-0,00005878
0,01691653	-0,00053640	0,00024735	-0,00014049	0,00008293	-0,00047617	0,00002528	0,00015629
0,01687580	-0,00078887	0,00036627	-0,00020835	0,00012299	-0,00007055	0,00003739	-0,00017041
0,01681888	-0,00103054	0,00048167	-0,00027441	0,00016197	-0,00009283	0,00004911	-0,00022302
0,01674588	-0,00126119	0,00059330	-0,00033849	0,00019979	-0,00011440	0,00006041	-0,00027333
0,01665691	-0,00148065	0,00070094	-0,00040045	0,00023635	-0,00013520	0,00007126	-0,00032126
0,01655214	-0,00168875	0,00080436	-0,00046016	0,00027156	-0,00015520	0,00008165	-0,00036673
0,01643176	-0,00188535	0,00090336	-0,00051748	0,00030536	-0,00017435	0,00009154	-0,00040964
0,01629597	-0,00207034	0,00099776	-0,00057230	0,00033766	-0,00019260	0,00010093	-0,00044994
0,01614502	-0,00224364	0,00108739	-0,00062448	0,00036840	-0,00020993	0,00010980	-0,00048758
0,01597917	-0,00240518	0,00117209	-0,00067395	0,00039751	-0,00022629	0,00011812	-0,00052251
0,01579872	-0,00255493	0,00125173	-0,00072059	0,00042495	-0,00024166	0,00012589	-0,00055470
0,01560398	-0,00269287	0,00132618	-0,00076433	0,00045066	-0,00025602	0,00013310	-0,00058412
0,01539530	-0,00281902	0,00139535	-0,00080509	0,00047460	-0,00026933	0,00013974	-0,00061076
0,01517303	-0,00293340	0,00145914	-0,00084281	0,00049672	-0,00028158	0,00014579	-0,00063462
0,01493758	-0,00303607	0,00151749	-0,00087743	0,00051700	-0,00029277	0,00015127	-0,00065571
0,01468935	-0,00312713	0,00157033	-0,00090890	0,00053541	-0,00030286	0,00015615	-0,00067403
0,01442878	-0,00320666	0,00161762	-0,00093720	0,00055194	-0,00031186	0,00016045	-0,00068963
0,01415631	-0,00327479	0,00165935	-0,00096230	0,00056655	-0,00031976	0,00016416	-0,00070252
0,01387242	-0,00333168	0,00169550	-0,00098417	0,00057926	-0,00032656	0,00016729	-0,00071275
0,01357761	-0,00337748	0,00172608	-0,00100283	0,00059005	-0,00033226	0,00016984	-0,00072037
0,01327237	-0,00341240	0,00175112	-0,00101825	0,00059893	-0,00033687	0,00017181	-0,00072544
0,01295724	-0,00343663	0,00177064	-0,00103047	0,00060591	-0,00034039	0,00017322	-0,00072802
0,01263275	-0,00345040	0,00178471	-0,00103950	0,00061100	-0,00034284	0,00017407	-0,00072818
0,01229947	-0,00345396	0,00179338	-0,00104537	0,00061422	-0,00034423	0,00017437	-0,00072601
0,01195794	-0,00344757	0,00179673	-0,00104812	0,00061559	-0,00034458	0,00017415	-0,00072157
0,01160877	-0,00343150	0,00179486	-0,00104779	0,00061515	-0,00034391	0,00017340	-0,00071497
0,01125253	-0,00340606	0,00178787	-0,00104444	0,00061293	-0,00034223	0,00017215	-0,00070630
0,01088982	-0,00337156	0,00177588	-0,00103814	0,00060896	-0,00033959	0,00017041	-0,00069565
0,01052125	-0,00332830	0,00175902	-0,00102895	0,00060330	-0,00033600	0,00016820	-0,00068313
0,01014743	-0,00327665	0,00173742	-0,00101695	0,00059599	-0,00033150	0,00016554	-0,00066883
0,00976898	-0,00321693	0,00171123	-0,00100223	0,00058708	-0,00032611	0,00016245	-0,00065288
0,00938653	-0,00314952	0,00168062	-0,00098487	0,00057663	-0,00031988	0,00015895	-0,00063538
0,00900069	-0,00307478	0,00164576	-0,00096497	0,00056469	-0,00031284	0,00015506	-0,00061645
0,00861211	-0,00299310	0,00160681	-0,00094263	0,00055133	-0,00030502	0,00015080	-0,00059620
0,00822140	-0,00290485	0,00156398	-0,00091796	0,00053661	-0,00029647	0,00014619	-0,00057474
0,00782919	-0,00281044	0,00151744	-0,00089108	0,00052061	-0,00028723	0,00014126	-0,00055219
0,00743611	-0,00271027	0,00146741	-0,00086209	0,00050339	-0,00027734	0,00013604	-0,00052868
0,00704278	-0,00260474	0,00141409	-0,00083113	0,00048503	-0,00026685	0,00013054	-0,00050431
0,00664982	-0,00249427	0,00135769	-0,00079831	0,00046560	-0,00025579	0,00012479	-0,00047920
0,00625783	-0,00237926	0,00129842	-0,00076376	0,00044518	-0,00024422	0,00011882	-0,00045348
0,00586742	-0,00226014	0,00123651	-0,00072761	0,00042385	-0,00023217	0,00011265	-0,00042724
0,00547918	-0,00213733	0,00117218	-0,00069000	0,00040168	-0,00021970	0,00010630	-0,00040061
0,00509371	-0,00201123	0,00110566	-0,00065105	0,00037876	-0,00020686	0,00009981	-0,00037369
0,00471157	-0,00188227	0,00103718	-0,00061091	0,00035518	-0,00019368	0,00009318	-0,00034659
0,00433334	-0,00175086	0,00096697	-0,00056972	0,00033100	-0,00018022	0,00008646	-0,00031942
0,00395955	-0,00161742	0,00089525	-0,00052760	0,00030632	-0,00016653	0,00007965	-0,00029227
0,00359075	-0,00148235	0,00082227	-0,00048471	0,00028122	-0,00015264	0,00007279	-0,00026525
0,00322746	-0,00134606	0,00074824	-0,00044117	0,00025578	-0,00013861	0,00006590	-0,00023844
0,00287018	-0,00120895	0,00067341	-0,00039713	0,00023007	-0,00012448	0,00005901	-0,00021195
0,00251941	-0,00107140	0,00059800	-0,00035272	0,00020419	-0,00011030	0,00005213	-0,00018584
0,00217562	-0,00093381	0,00052222	-0,00030808	0,00017821	-0,00009611	0,00004528	-0,00016022
0,00183925	-0,00079656	0,00044632	-0,00026334	0,00015221	-0,00008192	0,00003849	-0,00013514
0,00151076	-0,00066001	0,00037049	-0,00021863	0,00012627	-0,00006787	0,00003177	-0,00011069
0,00119054	-0,00052452	0,00029497	-0,00017408	0,00010046	-0,00005391	0,00002516	-0,00008694
0,00087899	-0,00039044	0,00021996	-0,00012982	0,00007486	-0,00004010	0,00001865	-0,00006393
0,00057649	-0,00025811	0,00014566	-0,00008598	0,00004954	-0,00002649	0,00001228	-0,00004174
0,00028338	-0,00012786	0,00007228	-0,00004266	0,00002456	-0,00001311	0,00000606	-0,00002042
0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000	0,00000000

b. filtre passe bas:

$$[-5 \ 0 \ 37 \ 64 \ 37 \ 0 \ -5] / 64$$

Table.1. Coefficients des filtres de la figure 4.4
Configuration globale du décodage NTSC

Filtre	Horizontal Passe Bande	Vertical Passe Haut	horizontal Passe Bas	Vertical- temporel
C0	168/512	1/2	64/64	Voir texte
C1	0	-1/4	37/64	
C2	-128/512		0	
C3	0		-5/64	
C4	48/512			
C5	0			
C6	0			
C7	0			
C8	-4/512			

Table.2. Coefficients des filtres de la figure 4.14
Configuration globale du décodage PAL

Filtre	Horizontal Passe Bande	Vertical Passe Haut	horizontal Passe Bas	Vertical- temporel
C0	168/512	1/2	64/64	Voir texte
C1	0	0	37/64	
C2	-128/512	-1/4	0	
C3	0		-5/64	
C4	48/512			
C5	0			
C6	0			
C7	0			
C8	-4/512			

Table.3. Coefficients des filtres de la figure 5.1

Filtre	Horizontal Passe Bande	Vertical Passe Haut	horizontal Passe Bas	Temporel En peigne	Spatial En Losange
C0	168/512	20/64	64/64	-1/2	Voir texte
C1	0	-15/64	37/64	1/2	
C2	-128/512	6/64	0		
C3	0	-1/64	-5/64		
C4	48/512				
C5	0				
C6	0				
C7	0				
C8	-4/512				

PROGRAMME DE L'ENCODEUR-DÉCODEUR NTSC
APPROCHE FIXE

```
#include <stdio.h>
extern "C" char tolower(char);
extern "C" double atof(char *);
#include <Ops/Ops.h>
#include <math.h>

image *conv135143(int frame,char *input,char *comp);
image *encoder(int frame,char *input,int opt);
image *decoder(image *input,int OD);
image *filtmp2d(image *past,image *present);
image *demodulation(int n,image *input,char C);
image *matricage(image *cI,image *cQ,char C);
image *matricageY(image *inputY);
void convfD1(image *cY,char C,image* &fD1c);
void ShowUse(char *str);
main(int argc,char **argv)
{

// Initialisation des variables

char input[60];
char output[60];
input[0]=0;output[0]=0;
int nbframe=1,startframe=1,CurrArg=0;
int opt=0,OD=0;
if(argc<6) ShowUse(argv[0]);
while(++CurrArg<argc)
{
if((argv[CurrArg] [0] !='-')) ShowUse(argv[0]);
switch(tolower(argv[CurrArg] [1]))
{
case'i'://Input sequence
strcpy(input,argv[++CurrArg]);
{
header h(input);
nbframe=h.length;
if(strcmp(h.type(),"YUV") !=0)
error("Ce n,est pas une sequence YUV");
}
break;
case'o'://Output sequence
strcpy(output,argv[++CurrArg]);
break;
case's':
startframe=atoi(argv[++CurrArg]);
break;
case'n':
nbframe=atoi(argv[++CurrArg]);
if(nbframe<2){
printf("\nN.B.\n\nle nombre de frame doit etre (-n nbframe>=2) Recommencez\n\n");
exit(0);}
}
}
}
```

Annexe

```
break;
case 'e': //option d'encodage
    opt=atoi(argv[++CurrArg]);
if(opt!=0 & opt!=1 & opt!=2){
    printf("\noption (ope) mal choisie tapez 0 1 ou 2\n\n");exit(-1);

    break;
case 'd': // option de decodage
    OD=atoi(argv[++CurrArg]);
if(OD!=1 & OD!=2){
    printf("\noption (opd) mal choisie tapez 0 1 ou 2\n\n");exit(-1);
    break;
default:
cerr<<"option invalide "<<argv[CurrArg]<<'\n';
}
}
// début du traitement

image *NTSC,*chroma,*chromat,*LUMA,*Yest,*Uest,*Vest,*I,*Q,*presente;
image *NTSCpa,*NTSCpr,*cfD1y,*cfD1u,*cfD1v;
char comp[2]='I','Q';
char cname[3]='Y','U','V';
    printf("\nlongueur de la sequence=%d\n",nbframe);
    printf("traitement du frame ");
    for(int frame_no=startframe;frame_no<=nbframe;frame_no++)
    {
        printf("%.%d",frame_no); fflush(stdout);
        NTSC = encoder(frame_no,input,opt);
        chroma= decoder(NTSC,OD);
        if(frame_no==1){
            delete chroma;
            NTSCpa=NTSC;}
        if(frame_no>=2){
            presente=chroma;
            NTSCpr=NTSC;
            chromat=filtmp2d(NTSCpa,NTSCpr);delete NTSCpa;
            LUMA= *NTSCpr-*presente;delete presente;
            Yest= matricageY(LUMA); delete LUMA;
            //Yest->display("Y composante de luminance");
            I=demodulation(frame_no,chromat,comp[0]);
            Q=demodulation(frame_no,chromat,comp[1]);delete chromat;
            Uest=matricage(I,Q,cname[1]);
            //Uest->display("U composante de chrominance");
            Vest=matricage(I,Q,cname[2]);delete I;delete Q;
            //Vest->display("V composante de chrominance");
            convfD1(Yest,cname[0],cfD1y);delete Yest;
            cfD1y->display("Yest");
            convfD1(Uest,cname[1],cfD1u);delete Uest;
            cfD1u->display("Uest");
            convfD1(Vest,cname[2],cfD1v);delete Vest;
            cfD1v->display("Vest");
            if (frame_no==2){cfD1y->header(output,YUV,nbframe-(startframe));
                cfD1y->write();}
            else {cfD1y->append(output);}
            cfD1u->append(output);
            cfD1v->append(output);
            delete cfD1y;delete cfD1u;delete cfD1v;
            NTSCpa=NTSCpr;
```

Annexe

```
    }  
  }  
  printf("\n... fini! \n");  
}
```

// Routine de l'encodeur

```
image *encoder(int frame,char *input,int opt){  
int m=frame;  
char *cname[] = { "Y","U","V" };  
    image *fD2,*cY,*cU,*cV;  
    for (int c=0; c<3; c++){  
        fD2 = conv135143(frame,input,cname[c]);  
        if(cname[c]=="Y") cY=fD2;  
        if(cname[c]=="U") cU=fD2;  
        if(cname[c]=="V") cV=fD2;  
    }  
int su1=16,su2=128;  
image *y=new image(cY->rows,cY->cols);  
image *I= new image(cY->rows,cY->cols);  
image *Q=new image(cY->rows,cY->cols);  
    for( int j=0;j<cY->rows;j++)  
    {  
        for( int i=0;i<cY->cols;i++)  
        {  
y->pixref(i,j)=.5913 *(cY->pixread(i,j)-su1) ;  
I->pixref(i,j)=.5961 *(cU->pixread(i,j)-su2) - 0.2750 *(cV->pixread(i,j)-su2);  
Q->pixref(i,j)=0.3874 *(cU->pixread(i,j)-su2) + .42260 *(cV->pixread(i,j)-su2);  
}}  
delete cU;delete cV;delete cY;delete fD2;  
image *I0=I->deinterlace(0);  
image *I1=I->deinterlace(1);delete I;  
image *Q0=Q->deinterlace(0);  
image *Q1=Q->deinterlace(1);delete Q;  
  
filter fm("modul.flr");  
image *If0=fm.horizontal(*I0);delete I0;  
image *If1=fm.horizontal(*I1);delete I1;  
image *Qf0=fm.horizontal(*Q0);delete Q0;  
image *Qf1=fm.horizontal(*Q1);delete Q1;  
  
image *Cm0=new image(*Qf0,image::nocopy);  
image *Cm1=new image(*Qf0,image::nocopy);  
//modulation  
for( j=0;j<Qf0->rows;j++)  
    {  
        for( int i=0;i<Qf0->cols;i++)  
        {  
Cm0->pixref(i,j)=Qf0->pixread(i,j)*sin(PI/2*i)*cos(PI*(-m+j))+If0->  
>pixread(i,j)*cos(PI/2*i)*cos(PI*(-m+j));  
Cm1->pixref(i,j)=Qf1->  
>pixread(i,j)*sin(PI/2*i)*cos(PI*(m+j))+If1->pixread(i,j)*cos(PI/2*i)*cos(PI*(-m+j));  
}}  
delete Qf0;delete Qf1;delete If0;delete If1;  
filter fcr("notch5.flr");  
filter fver("verticalentsc.flr");  
filter fy("y_filter.flr");
```

Annexe

```
image *NTSC=new image(y->rows,y->cols);
if(opt==2){
    image *Cm=Cm0->interlace(*Cm1);delete Cm0;delete Cm1;
    image *Y_C=new image(*Cm,image::docopy);delete Cm;////*Cm;delete Cm;
    image *Y_CO=Y_C->deinterlace(0);
    image *Y_C1=Y_C->deinterlace(1); delete Y_C;
    image *Y_C0f=fcr.horizontal(*Y_CO); delete Y_CO;
    image *Y_C1f=fcr.horizontal(*Y_C1);delete Y_C1;
    image *Y_C0fv=fver.vertical(*Y_C0f); delete Y_C0f;
    image *Y_C1fv=fver.vertical(*Y_C1f); delete Y_C1f;
    image *Y_Cf=Y_C0fv->interlace(*Y_C1fv);delete Y_C0fv;delete Y_C1fv;
    NTSC=Y_Cf;/////delete Y_Cf;
}
if(opt==0){
    image *Cm00=fcr.horizontal(*Cm0);delete Cm0;
    image *Cm11=fcr.horizontal(*Cm1);delete Cm1;
    image *Cm=Cm00->interlace(*Cm11);delete Cm00;delete Cm11;
    image *y1=fy.horizontal(*y);
    NTSC=*y1+*Cm;delete Cm;delete y1;
}
if(opt==1){
    image *Cm=Cm0->interlace(*Cm1); delete Cm0;delete Cm1;
    image *Y_C=new image(*Cm,image::docopy);delete Cm;////*Cm;delete Cm;
    image *Y_CO=Y_C->deinterlace(0);
    image *Y_C1=Y_C->deinterlace(1);delete Y_C;
    image *Y_C0f=fcr.horizontal(*Y_CO);delete Y_CO;
    image *Y_C1f=fcr.horizontal(*Y_C1); delete Y_C1;
    image *Y_Cf=Y_C0f->interlace(*Y_C1f);delete Y_C1f;delete Y_COf;
    NTSC=Y_Cf;/////delete Y_Cf;
}
delete y;
return NTSC

} // Fin de la routine.
```

// Routine du décodeur

```
image *decoder(image *input,int OD){
    filter fh("notch5.flt");
    filter fver("verticalentsclong.flt");
    image *resultat;
    image *field0=input->deinterlace(0);
    image *field1=input->deinterlace(1);
    image *out0=fh.horizontal(*field0);delete field0;
    image *out1=fh.horizontal(*field1);delete field1;

    if(OD==2){
        image *B0=fver.vertical(*out0);delete out0;
        image *B1=fver.vertical(*out1);delete out1;
        resultat=B0->interlace(*B1); delete B0;delete B1;
    }
    else{
        resultat=out0->interlace(*out1); delete out0;delete out1;
    }
    return resultat;
} //fin de la routine
```

// Routine de démodulation

```
image *demodulation(int n,image *input,char C){
    char comp1[2]={I,'Q'};
    filter f1("demodul.flt");
    image *input0=input->deinterlace(0);
```

Annexe

```
image *input1=input->deinterlace(1);
image *evenfield=new image(*input0,image::nocopy);
image *oddfield=new image(*input0,image::nocopy);
    for(int j=0;j<input0->rows;j++){
        for(int i=0;i<input0->cols;i++){
            if(C==comp1[1]){
                evenfield->pixref(i,j)=2*input0->pixread(i,j)*(sin((i*PI/2)))*cos(PI*(-n+j));
                oddfield->pixref(i,j)=2*input1->pixread(i,j)*(sin((i*PI/2)))*cos(PI*(-n+j));
            }
            else{
                evenfield->pixref(i,j)=2*input0->pixread(i,j)*(cos((i*PI/2)))*cos(PI*(-n+j));
                oddfield->pixref(i,j)=2*input1->pixread(i,j)*(cos((i*PI/2)))*cos(PI*(-n+j));
            }
        }
    }
delete input0;delete input1;
image *evenfield0=f1.horizontal(*evenfield);delete evenfield;
image *oddfield1=f1.horizontal(*oddfield);delete oddfield;
image *resu=evenfield0->interlace(*oddfield1);delete oddfield1;delete evenfield0;
return resu;
}
```

//Fin de la routine

```
image *matricage(image *cI,image *cQ,char C){
char cname[2]='U','V';
int su1=128;
image *res=new image(cI->rows,cI->cols);
    for(int j=0;j<cI->rows;j++){
        for(int i=0;i<cI->cols;i++){
            if(C==cname[0]){
                res->pixref(i,j)=1.1790 *cI->pixread(i,j)+ 0.7672 *cQ->pixread(i,j)+su1;
            }
            else{
                res->pixref(i,j)=-1.0808 *cI->pixread(i,j)+ 1.6630 *cQ->pixread(i,j)+su1;
            }
        }
    }
return res;
}
image *matricageY(image *inputY){
int su2=16;
image *Y=new image(inputY->rows,inputY->cols);
    for(int j=0;j<inputY->rows;j++){
        for(int i=0;i<inputY->cols;i++){
            Y->pixref(i,j)=1.6912*inputY->pixread(i,j)+su2;
        }
    }
return Y;
}
```

// Routine de la conversion du format D1 au format D2

```
image *conv135143(int frame,char *input,char *comp){
    image *in=new image(frame,input,comp);

int col,pp,xx,U1=35,D1=33;
float pu,xu,gm;
filter f("updown.flr");
image *C135;
    if(comp=="U" || comp=="V") C135=f.ihorizontal(*in,2);
    else C135=new image(*in,image::docopy);delete in;
FILE *fptr;
long double coef[601];
    if((fptr=fopen("coefntsc","r"))==NULL)
        printf("LE FICHER EST VIDE");
    for (int n1=0;n1<=600;n1++){
```

Annexe

```
fscanf(fp, "%lf\n", &coef[n1]);

//Passage de 13.5MHz a 14.31MHz
col=(U1*C135->cols-(U1*C135->cols)%D1)/D1;
col+=1;
image *fD2=new image(C135->rows,col);
for( int j=0;j<fD2->rows;j++)
{
    for( int i=0;i<fD2->cols;i++)
    {
        pu=0;
        for(int n=0;n<=16 ;n++)
        {
            xx=U1*n+(D1*i)%U1;
            gm=35*coef[xx];
            pp=(i*D1-(i*D1)%U1)/U1-n;
            xu=C135->pixread(pp,j);
            pu=pu+gm*xu;
        }
        fD2->pixref(i,j)=pu;
    }
}

delete C135;
fclose(fp);
image *fD2c=new image(fD2->rows,col);
for( j=0;j<fD2c->rows;j++)
{
    for( int i=0;i<fD2c->cols;i++)
    {
        fD2c->pixref(i,j)=fD2->pixread(i+8,j);}
}

delete fD2;
return fD2c;
} // Fin de la routine
```

// Implémentation du filtre vertical-temporel

```
image *filtmp2d(image *past,image *present){
    filter fh("notch5.fl");
    image *past1=fh.horizontal(*past);
    image *present1=fh.horizontal(*present);
    int A=16,Ce=32;
    image *tmp=new image(present1->rows,present1->cols);
    double p0;
    for(int j=0;j<present1->rows;j++){
        if((j%2)!=0){
            for(int i=0;i<present1->cols;i++){
                p0=Ce*(present1->pixread(i,j)-past1->pixread(i,j))+A*(-present1->pixread(i,j-2)-present1->pixread(i,j+2)+past1->pixread(i,j-2)+past1->pixread(i,j+2));
                tmp->pixref(i,j)=1./128*p0;}}
            else{
                for(int i=0;i<present1->cols;i++){
                    p0=Ce*(present1->pixread(i,j)-past1->pixread(i,j))+A*(-present1->pixread(i,j-2)-present1->pixread(i,j+2)+past1->pixread(i,j-2)+past1->pixread(i,j+2));
                    tmp->pixref(i,j)=1./128*p0;}}
            }
        delete past1;
        delete present1;
        return tmp;
    }
```

Annexe

```
}  
// Fin de la routine  
  
// Routine de la conversion du format D2 en format D1  
  
void convfD1(image *input,char C,image* &fD1c){  
int col,p,x,U1=33,D1=35;  
float py,gm,xy;  
char cname[3]={'Y','U','V'};  
filter dirac(1);  
FILE *fptr;  
long double coef[628];  
    if((fptr=fopen("coefntscinv","r"))==NULL)  
        printf("LE FICHER EST VIDE");  
    for (int n1=0;n1<=627;n1++){  
        fscanf(fptr,"%lf\n",&coef[n1]);  
col=int(U1*input->cols-(U1*input->cols)%D1)/D1;  
image *fD1=new image(input->rows,col);  
    for( int j=0;j<fD1->rows;j++)  
    {  
        for( int i=0;i<fD1->cols;i++)  
        {  
            py=0;  
            for(int n=0;n<=17 ;n++)  
            {  
                x=U1*n+(D1*i)%U1;  
                gm=33*coef[x];  
                p=(i*D1-(i*D1)%U1)/U1-n;  
                xy=input->pixread(p,j);  
                py=py+gm*xy;  
            }  
            fD1->pixref(i,j)=py;  
        }  
    }  
fclose(fptr);  
image *fD1c1=new image(fD1->rows,col);  
    for( j=0;j<fD1c1->rows;j++)  
    {  
        for( int i=0;i<fD1c1->cols;i++)  
        {  
            fD1c1->pixref(i,j)=fD1->pixread(i+9,j);}  
    }  
delete fD1;  
if(C==cname[1] || C==cname[2]) fD1c=dirac.horizontal(fD1c1,2);  
else fD1c=dirac.horizontal(fD1c1,1);  
delete fD1c1;  
  
}// Fin de la routine  
  
void ShowUse(char *str)  
{  
    printf("\n usage: %s -i input -o output -s start frame -n number of frame -e encodage -d  
    decodage\n",str);  
    exit(1);  
}
```

FIN DU PROGRAMME

PROGRAMME DE L'ENCODEUR DECODEUR PAL
APPROCHE FIXE

```
#include <stdio.h>
extern "C" char tolower(char);
extern "C" double atof(char *);
#include <Ops/Ops.h>
#include <math.h>

image *conv135143(int frame,char *input,char *comp);
image *encoder(int frame,char *input,int opt);
image *decoder(image *input,int OD);
image *filtmp2d(image *past,image *present);
image *demodulation(int n,image *input,char C);
image *matricage(image *cU,image *cV,char C);
image *matricageY(image *inputY);
void convfD1(image *cY,char C,image* &fD1c);
void ShowUse(char *str);
main(int argc,char **argv)
{
//Initialisation

char input[60];
char output[60];
input[0]=0;output[0]=0;
int nbframe=1,startframe=1,CurrArg=0;
int opt=0,OD=0;
if(argc<6) ShowUse(argv[0]);
while(++CurrArg<argc)
{
if((argv[CurrArg] [0] !='-')) ShowUse(argv[0]);
switch(tolower(argv[CurrArg] [1]))
{
case'i'://Input sequence
strcpy(input,argv[++CurrArg]);
{
header h(input);
nbframe=h.length;
if(strcmp(h.type0,"YUV") !=0)
error("Ce n,est pas une sequence YUV");
}
break;
case'o'://Output sequence
strcpy(output,argv[++CurrArg]);
break;
case's':
startframe=atoi(argv[++CurrArg]);
break;
case'n':
nbframe=atoi(argv[++CurrArg]);
if(nbframe<2){
printf("\nN.B.\n\nle nombre de frame doit etre (-n nbframe>=) Recommencez\n\n");
exit(0);}
break;
case'e': //option d'encodage
```

Annexe

```
    opt=atoi(argv[++CurrArg]);
if(opt!=0 & opt!=1 & opt!=2){
    printf("\noption (ope) mal choisie tapez 0 1 ou 2\n\n");exit(-1);

    break;
    case 'd': // option de decodage
        OD=atoi(argv[++CurrArg]);
if(OD!=1 & OD!=2){
    printf("\noption (opd) mal choisie tapez 0 1 ou 2\n\n");exit(-1);
    break;
default:
cerr<<"option invalide "<<argv[CurrArg]<<'\n';
}
}

image *PAL,*chroma,*chromat,*LUMA,*Yest,*cRest,*cBest,*cU,*cV,*PALpa,*presente;
image *PALpr,*cfD1y,*cfD1u,*cfD1v;
char cname[3]=('Y','U','V');

// Début du traitement
printf("\nlongueur de la sequence=%d\n",nbframe);
printf("traitement du frame ");
for(int frame_no=startframe;frame_no<=nbframe;frame_no++)
{
    printf(".%d",frame_no); fflush(stdout);
    PAL = encoder(frame_no,input,opt);
    chroma= decoder(PAL,OD);
    if(frame_no==1){
        delete chroma;
        PALpa=PAL;}
    if(frame_no>=2){
        presente=chroma;
        PALpr=PAL;
        chromat=filtmp2d(PALpa,PALpr);delete PALpa;
        LUMA= *PALpr-*presente;delete presente;
        Yest= matricageY(LUMA); delete LUMA;
        // Yest->display("Yest Luminance");
        cU=demodulation(frame_no,chromat,cname[1]);
        cV=demodulation(frame_no,chromat,cname[2]);delete chromat;
        cRest=matricage(cU,cV,cname[1]); //cRest->display("cRest 38");
        cBest=matricage(cU,cV,cname[2]);delete cU;delete cV;
        //cBest->display("cBest 38");
        convfD1(Yest,cname[0],cfD1y);delete Yest;
        cfD1y->display("Y2fnotch5");
        convfD1(cRest,cname[1],cfD1u);delete cRest;
        cfD1u->display("U2fnotch5");
        convfD1(cBest,cname[2],cfD1v);delete cBest;
        cfD1v->display("V2fnotch5");
        if (frame_no==2){cfD1y->header(output,YUV,nbframe-(startframe));
            cfD1y->write();}
        else {cfD1y->append(output);}
        cfD1u->append(output);
        cfD1v->append(output);
        delete cfD1y;delete cfD1u;delete cfD1v;
        PALpa=PALpr;
    }
}
printf("\n... fini! \n");
}
```

// Routine de l'encodeur

```

image *encoder(int frame,char *input,int opt){
int m=frame;
char *cname[] = { "Y","U","V" };
    image *fD2,*cY,*cR,*cB;
    for (int c=0; c<3; c++){
fD2 = conv135143(frame,input,cname[c]);
if(cname[c]=="Y") cY=fD2;
if(cname[c]=="U") cR=fD2;
if(cname[c]=="V") cB=fD2;
    }
image *y=new image(cY->rows,cY->cols);
image *U= new image(cY->rows,cY->cols);
image *V=new image(cY->rows,cY->cols);
    for( int j=0;j<cY->rows;j++)
    {
        for( int i=0;i<cY->cols;i++)
        {
y->pixref(i,j)= (cY->pixread(i,j)) ;
U->pixref(i,j)=.0000 *(cR->pixread(i,j)) - .8741 *(cB->pixread(i,j));
V->pixref(i,j)=1.2300 *(cR->pixread(i,j)) + .0000 *(cB->pixread(i,j));
        }
    }
delete cR;delete cB;delete cY;delete fD2;
image *U0=U->deinterlace(0);
image *U1=U->deinterlace(1);delete U;
image *V0=V->deinterlace(0);
image *V1=V->deinterlace(1);delete V;
filter fm("modulpal15.flr");
image *Uf0=fm.horizontal(*U0);delete U0;
image *Uf1=fm.horizontal(*U1);delete U1;
image *Vf0=fm.horizontal(*V0);delete V0;
image *Vf1=fm.horizontal(*V1);delete V1;
image *Cm0=new image(*Vf0,image::nocopy);
image *Cm1=new image(*Vf0,image::nocopy);
//modulation
    if(m%4==1){
for( j=0;j<Vf0->rows;j++)
    {
        for( int i=0;i<Vf0->cols;i++)
        {
Cm0->pixref(i,j)=-(Vf0->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))+Uf0-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));
Cm1->pixref(i,j)=Vf1->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))+Uf1-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j));}}
        if(m%4==2){
for( j=0;j<Vf0->rows;j++)
    {
        for( int i=0;i<Vf0->cols;i++)
        {
Cm0->pixref(i,j)=(Uf0->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))-Vf0-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));
Cm1->pixref(i,j)=-(Uf1->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))-Vf1-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));}}
        if(m%4==3){
for( j=0;j<Vf0->rows;j++)
    {
        for( int i=0;i<Vf0->cols;i++)
        {

```

Annexe

```
Cm1->pixref(i,j)=-(Vf1->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))+Uf1-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));
Cm0->pixref(i,j)=Vf0->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))+Uf0-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j));}}
if(m%4==0){
for( j=0;j<Vf0->rows;j++)
{
for( int i=0;i<Vf0->cols;i++)
{
Cm1->pixref(i,j)=(Uf1->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))-Vf1-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));
Cm0->pixref(i,j)=-(Uf0->pixread(i,j)*sin(PI/2*(i+j))*cos(PI*(m-j))-Vf0-
>pixread(i,j)*cos(PI/2*(i+j))*cos(PI*(m-j)));}}
delete Vf0;delete Vf1;delete Uf0;delete Uf1;
filter fh("notch5.flr");
filter fv("verticalepal.flr");
image *PAL=new image(y->rows,y->cols);
if(opt==2){

image *Cm=Cm0->interlace(*Cm1);delete Cm0;delete Cm1;
image *Y_C=new image(*Cm,image::docopy);delete Cm;///*Cm; delete
image *Y_C0=Y_C->deinterlace(0);
image *Y_C1=Y_C->deinterlace(1); delete Y_C;
image *Y_C0f=fv.vertical(*Y_C0); delete Y_C0;
image *Y_C1f=fv.vertical(*Y_C1);delete Y_C1;
image *Y_C0fv=fh.horizontal(*Y_C0f);delete Y_C0f;
image *Y_C1fv=fh.horizontal(*Y_C1f);delete Y_C1f;
image *Y_Cf=Y_C0fv->interlace(*Y_C1fv);delete Y_C0fv;delete Y_C1fv;
PAL=Y_Cf;/////delete Y_Cf;
}
if(opt==0){

image *Cm00=fh.horizontal(*Cm0);delete Cm0;
image *Cm11=fh.horizontal(*Cm1);delete Cm1;
image *Cm=Cm00->interlace(*Cm11);delete Cm00;delete Cm11;
PAL=*y+*Cm;delete Cm;}
if(opt==1){

image *Cm=Cm0->interlace(*Cm1); delete Cm0;delete Cm1;
image *Y_C=new image(*Cm,image::docopy);delete Cm;///*Cm; delete
image *Y_C0=Y_C->deinterlace(0);
image *Y_C1=Y_C->deinterlace(1);delete Y_C;
image *Y_C0f=fh.horizontal(*Y_C0);delete Y_C0;
image *Y_C1f=fh.horizontal(*Y_C1); delete Y_C1;
image *Y_Cf=Y_C0f->interlace(*Y_C1f);delete Y_C1f;delete Y_C0f;
PAL=Y_Cf;/////delete Y_Cf;
}
delete y;
return PAL;

}/////Fin de la routine

// Routine du d codeur

image *decoder(image *input,int OD){
filter fh("notch5.flr");
filter fv("verticalepallong.flr");
image *resultat;
image *field0=input->deinterlace(0);
image *field1=input->deinterlace(1);
```

Annexe

```
image *out0=fh.horizontal(*field0);delete field0;
image *out1=fh.horizontal(*field1);delete field1;
if(OD==2){
image *B0,*B1;
B0=fv.vertical(*out0);
B1=fv.vertical(*out1);
delete out0,delete out1;
resultat=B0->interlace(*B1); delete B0;delete B1;}
else{
resultat=out0->interlace(*out1); delete out0;delete out1;}
return resultat;
}// Fin de la routine
```

// Routine de la Démodulation

```
image *demodulation(int n,image *input,char C){
int z=n;
char comp1[2]=('U','V');
filter f1("demodul.flr");
image *input0=input->deinterlace(0);
image *input1=input->deinterlace(1);
image *evenfield=new image(*input0,image::nocopy);
image *oddfield=new image(*input0,image::nocopy);
if(z%4==1){
if(C==comp1[0]){
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= 2 * input0->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= -2 * input1->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));}}
else{
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= -2 * input0->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= 2 * input1->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));}}
}
}
if(z%4==2){
if(C==comp1[0]){
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= -2 * input0->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= 2 * input1->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));}}
else{
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= -2 * input0->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= 2 * input1->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));}}
}
}
if(z%4==3){
if(C==comp1[0]){
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= -2 * input0->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= 2 * input1->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));}}
else{
for(int j=0;j<input0->rows;j++){
for(int i=0;i<input0->cols;i++){
evenfield->pixref(i,j)= 2 * input0->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));
oddfield->pixref(i,j)= -2 * input1->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));}}
}
}
}
```

Annexe

```
}
  if(z%4==0){
    if(C==comp1[0]){
      for(int j=0;j<input0->rows;j++){
        for(int i=0;i<input0->cols;i++){
          evenfield->pixref(i,j)= 2 * input0->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));
          oddfield->pixref(i,j)= -2 * input1->pixread(i,j) * sin(PI/2*(i+j))*cos(PI*(z-j));}}
        else{
          for(int j=0;j<input0->rows;j++){
            for(int i=0;i<input0->cols;i++){
              evenfield->pixref(i,j)= 2 * input0->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));
              oddfield->pixref(i,j)= -2 * input1->pixread(i,j) * cos(PI/2*(i+j))*cos(PI*(z-j));}}
            }
          delete input0;delete input1;
          image *evenfield0=f1.horizontal(*evenfield);delete evenfield;
          image *oddfield1=f1.horizontal(*oddfield);delete oddfield;
          image *resu=evenfield0->interlace(*oddfield1);delete oddfield1;delete evenfield0;
          return resu;
        }
      }
```

//Fin de la routine

```
image *matricage(image *cU,image *cV,char C){
  char cname[2]='U', 'V';
  image *res=new image(cU->rows,cV->cols);
  for(int j=0;j<cU->rows;j++){
    for(int i=0;i<cU->cols;i++){
      if(C==cname[0]){
        res->pixref(i,j)=0.0000 *cU->pixread(i,j)+ 0.8130 *cV->pixread(i,j);
      }
      else{
        res->pixref(i,j)=1.1440 *cU->pixread(i,j)+ 0.0000 *cV->pixread(i,j);
      }
    }
  }
  return res;
}

image *matricageY(image *inputY){
  image *Y=new image(inputY->rows,inputY->cols);
  for(int j=0;j<inputY->rows;j++){
    for(int i=0;i<inputY->cols;i++){
      Y->pixref(i,j)=inputY->pixread(i,j);
    }
  }
  return Y;
}
```

// Routine de la conversion de la format D1 en format D2

```
image *conv135143(int frame,char *input,char *comp){ en format d
  image *in=new image(frame,input,comp);
  int col,pp,xx,U1=59,D1=45;
  float pu,xu,gm;
  filter f("updown.fl");
  image *C135;
  if(comp=="U" || comp=="V") C135=f.i.horizontal(*in,2);
  else C135=new image(*in,image::docopy);
  delete in;
  FILE *fptr;
  long double coef[1003];
  if((fptr=fopen("coefPAL","r"))==NULL)
    printf("LE FICHER EST VIDE");
  for (int n1=0;n1<=1002;n1++){
```

Annexe

```
fscanf(fp, "%f\n", &coef[n1]);
//Passage de 13.5MHz a 14.31MHz
col=(U1*C135->cols-(U1*C135->cols)%D1)/D1;
image *fD2=new image(C135->rows,col);
for( int j=0;j<fD2->rows;j++)
{
    for( int i=0;i<fD2->cols;i++)
    {
        pu=0;
        for(int n=0;n<=16 ;n++)
        {
            xx=U1*n+(D1*i)%U1;
            gm=U1*coef[xx];
            pp=(i*D1-(i*D1)%U1)/U1-n;
            xu=C135->pixread(pp,j);
            pu=pu+gm*xu;
        }
        fD2->pixref(i,j)=pu;
    }
}
delete C135;
fclose(fp);
image *fD2c=new image(fD2->rows,col);
for( j=0;j<fD2c->rows;j++)
{
    for( int i=0;i<fD2c->cols;i++)
    {
        fD2c->pixref(i,j)=fD2->pixread(i+10,j);}
}
delete fD2;
return fD2c;
} //Fin de la routine
```

// Implémentation du filtre vertical- temporel

```
image *filtmp2d(image *past,image *present){
int A=-19,B=8,C=10,D=-16,Ce=38;
filter fh("notch5.flt");
image *past1=fh.horizontal(*past);
image *present1=fh.horizontal(*present);
image *tmp=new image(present1->rows,present1->cols);
double p0;
for(int j=0;j<present1->rows;j++){
    if((j%2)!=0){
        for(int i=0;i<present1->cols;i++){
            p0=Ce*present1->pixread(i,j)+A*(present1->pixread(i,j-4)+present1->pixread(i,j+4))+B*(past1->pixread(i,j-3)+past1->pixread(i,j+5))+C*(-past1->pixread(i,j-2)+past1->pixread(i,j+2))+D*past1->pixread(i,j+1);
            tmp->pixref(i,j)=1./128*p0;}}
        else{
            for(int i=0;i<present1->cols;i++){
                p0=Ce*present1->pixread(i,j)+A*(present1->pixread(i,j-4)+present1->pixread(i,j+4))+B*(present1->pixread(i,j-3)+present1->pixread(i,j+5))+C*(-past1->pixread(i,j-2)+past1->pixread(i,j+2))+D*present1->pixread(i,j+1);
                tmp->pixref(i,j)=1./128*p0;}}
            }
    delete past1;delete present1;
    return tmp;
}
```

Annexe

// Fin de la routine

// Routine de la conversion du Format D2 en Format D1

```
void convfD1(image *input,char C,image* &fD1c){
int col,p,x,U1=45,D1=59;
float py,gm,xy;
char cname[3]='Y','U','V';
filter dirac(1);
FILE *fptr;
long double coef[991];
    if((fptr=fopen("coefPAL","r"))==NULL)
        printf("LE FICHER EST VIDE");
    for (int n1=1;n1<=990;n1++){
        fscanf(fptr,"%lf\n",&coef[n1]);}
col=int(U1*input->cols-(U1*input->cols)%D1)/D1;
image *fD1=new image(input->rows,col);
    for( int j=0;j<fD1->rows;j++)
    {
        for( int i=0;i<fD1->cols;i++)
        {
            py=0;
            for(int n=1;n<=21 ;n++)
            {
                x=U1*n+(D1*i)%U1;
                gm=U1*coef[x];
                p=(i*D1-(i*D1)%U1)/U1-n;
                xy=input->pixread(p,j);
                py=py+gm*xy;
            }
            fD1->pixref(i,j)=py;
        }
    }
fclose(fptr);
image *fD1c1=new image(fD1->rows,col);
    for( j=0;j<fD1c1->rows;j++)
    {
        for( int i=0;i<fD1c1->cols;i++)
        {
            fD1c1->pixref(i,j)=fD1->pixread(i+8,j);}
    }
delete fD1;
if(C==cname[1] || C==cname[2]) fD1c=dirac.horizontal(fD1c1,2);
else fD1c=dirac.horizontal(fD1c1,1);
delete fD1c1;
} // Fin de la routine

void ShowUse(char *str)
{
    printf("\n usage: %s -i input -o output -s start frame -n number of frame -e option
d'encodeage -d option de decodage\n",str);
    exit(1);
}
```

FIN DU PROGRAMME

PROGRAMME DE L'ENCODEUR DECODEUR NTSC A 3 DIMENSIONS
ADAPTATIVE

```
#include <stdio.h>
extern "C" char tolower(char);
extern "C" double atof(char *);
#include <Ops/Ops.h>
#include <math.h>

float abs(float y);
image *adaptationver(image *Ntscpr);
image *adaptationtmp(image *Ntscpa,image *Ntscpr);
image *conv135143(int frame,char *input,char *comp);
image *encoder(int frame,char *input,int opt);
image *multi(image *input,float fact);
image *decoder(image *ntscpa,image *ntscfu);
image *filtmp2d(image *past,image *present);
image *diagonal30(image *input);
image *diagonal120(image *input);
image *oblique(image *input);
void luminance(image *ntscpa,image *ntscpr,image* &resuhOB,image* &lum);
image *spatial1(image *input);
image *spatial2(image *input);
image *demodulation(int n,image *input,char C);
image *matricage(image *cI,image *cQ,char C);
image *matricageY(image *inputY);
void convfD1(image *cY,char C,image* &fD1c);
void ShowUse(char *str);

main(int argc,char **argv)
{
// Initialisation des variables
char input[60];
char output[60];
input[0]=0;output[0]=0;
int nbframe=1,startframe=1,CurrArg=0;
int opt=0;
if(argc<5) ShowUse(argv[0]);
while(++CurrArg<argc)
{
if((argv[CurrArg] [0] !='-')) ShowUse(argv[0]);
switch(tolower(argv[CurrArg] [1]))
{
case'i'://Input sequence
strcpy(input,argv[++CurrArg]);
{
header h(input);
nbframe=h.length;
if(strcmp(h.type0,"YUV") !=0)
error("Ce n,est pas une sequence YUV");
}
}
break;
case'o'://Output sequence
strcpy(output,argv[++CurrArg]);
```

Annexe

```
break;
  case's':
    startframe=atoi(argv[++CurrArg]);
break;
  case'n':
    nbframe=atoi(argv[++CurrArg]);
    if(nbframe<3){
printf("\nN.B.\n\nle nombre de frame doit etre (-n nbframe>=3) Recommencez\n\n");
exit(0);}

break;
  case 'e': //option d'encodage
    opt=atoi(argv[++CurrArg]);
if(opt!=0 & opt!=1 & opt!=2){
  printf("\noption (ope) mal choisie tapez 0 1 ou 2\n\n");exit(-1);}

  break;
default:
  cerr<<"option invalide "<<argv[CurrArg]<<'\n';
}
}

image *NTSC,*chromatpa,*chromatpr,*chromat,*LUMA,*Yest,*Uest,*Vest,*I,*Q;
image *NTSCpa,*NTSCpr,*cfD1y,*cfD1u,*cfD1v;
char comp[2]='I','Q';
char cname[3]='Y','U','V';
  printf("\nlongueur de la sequence=%d\n",nbframe);
  printf("traitement du frame ");

// début du traitement

  for(int frame_no=startframe;frame_no<=nbframe;frame_no++)
  {
    printf("%.d",frame_no); fflush(stdout);

    NTSC = encoder(frame_no,input,opt);
    if(frame_no==1){ NTSCpa=NTSC;}
    if(frame_no==2){ NTSCpr=NTSC;}
    luminance(NTSCpa,NTSCpr,chromatpa,LUMA);delete NTSCpa;NTSCpa=NTSCpr;}
    if(frame_no>2){ NTSCpr=NTSC;}
    luminance(NTSCpa,NTSCpr,chromatpr,LUMA);delete NTSCpa;
    chromat=decoder(chromatpa,chromatpr);delete chromatpa;
    Yest= matricageY(LUMA); delete LUMA;
    // Yest->display("Luminance");
    I=demodulation(frame_no,chromat,comp[0]);
    Q=demodulation(frame_no,chromat,comp[1]);delete chromat;
    Uest=matricage(I,Q,cname[1]);
    //Uest->display("U 1ere composante de chrominance ");
    Vest=matricage(I,Q,cname[2]);delete I;delete Q;
    //Vest->display("V 2eme composante de chrominance");
    convfD1(Yest,cname[0],cfD1y);delete Yest;
    cfD1y->display("interfer luma");
    convfD1(Uest,cname[1],cfD1u);delete Uest;
    cfD1u->display("interfer chroma U");
    convfD1(Vest,cname[2],cfD1v);delete Vest;
    cfD1v->display("interfer chroma V");
    if (frame_no==3){cfD1y->header(output,YUV,nbframe-(startframe+1));
      cfD1y->write();}
    else {cfD1y->append(output);}
```

Annexe

```
    cfD1u->append(output);
    cfD1v->append(output);
    delete cfD1y;delete cfD1u;delete cfD1v;
    NTSCpa=NTSCpr;
    chromatpa=chromatpr;
}
}
printf("\n... fini! \n");

}

// Routine de l'encodeur

image *encoder(int frame,char *input,int opt){
int m=frame;
char *cname[] = { "Y","U","V" };
    image *fD2,*cY,*cU,*cV;
    for (int c=0; c<3; c++){
        fD2 = conv135143(frame,input,cname[c]);
        if(cname[c]=="Y") cY=fD2;
        if(cname[c]=="U") cU=fD2;
        if(cname[c]=="V") cV=fD2;
    }
int su1=16,su2=128;
image *y=new image(cY->rows,cY->cols);
image *I= new image(cY->rows,cY->cols);
image *Q=new image(cY->rows,cY->cols);
    for( int j=0;j<cY->rows;j++)
    {
        for( int i=0;i<cY->cols;i++)
        {
            y->pixref(i,j)=.5913 *(cY->pixread(i,j)-su1) ;
            I->pixref(i,j)=.5961 *(cU->pixread(i,j)-su2) - 0.2750 *(cV->pixread(i,j)-su2);
            Q->pixref(i,j)=0.3874 *(cU->pixread(i,j)-su2) + .42260 *(cV->pixread(i,j)-su2);
        }
    }
delete cU;delete cV;delete cY;delete fD2;
image *IO=I->deinterlace(0);
image *I1=I->deinterlace(1);delete I;
image *Q0=Q->deinterlace(0);
image *Q1=Q->deinterlace(1);delete Q;
filter fi("i_filter.flt");
filter fq("q_filter.flt");
image *If0=fi.horizontal(*IO);delete IO;
image *If1=fi.horizontal(*I1);delete I1;
image *Qf0=fq.horizontal(*Q0);delete Q0;
image *Qf1=fq.horizontal(*Q1);delete Q1;
image *Cm0=new image(*Qf0,image::nocopy);
image *Cm1=new image(*Qf0,image::nocopy);

//modulation
for( j=0;j<Qf0->rows;j++)
{
    for( int i=0;i<Qf0->cols;i++)
    {
        Cm0->pixref(i,j)=Qf0->pixread(i,j)*sin(PI/2*i)*cos(PI*(-m+j))+If0-
        >pixread(i,j)*cos(PI/2*i)*cos(PI*(-m+j));
    }
}
```

Annexe

```
Cm1->pixref(i,j)=Qf1->pixread(i,j)*sin(PI/2*i)*cos(PI*(-m+j))+If1-
>pixread(i,j)*cos(PI/2*i)*cos(PI*(-m+j));
}}
delete Qf0;delete Qf1;delete If0;delete If1;
filter fcr("notch5.flt");
filter fver("verticalentsc.flt");
image *NTSC=new image(y->rows,y->cols);
if(opt==2){
    image *Cm=Cm0->interlace(*Cm1);delete Cm0;delete Cm1;
    image *Y_C=*y-*Cm;delete Cm;
    image *Y_C0=Y_C->deinterlace(0);
    image *Y_C1=Y_C->deinterlace(1); delete Y_C;
    image *Y_C0f=fcr.horizontal(*Y_C0); delete Y_C0;
    image *Y_C1f=fcr.horizontal(*Y_C1);delete Y_C1;
    image *Y_C0fv=fver.vertical(*Y_C0f); delete Y_C0f;
    image *Y_C1fv=fver.vertical(*Y_C1f); delete Y_C1f;
    image *Y_Cf=Y_C0fv->interlace(*Y_C1fv);delete Y_C0fv;delete Y_C1fv;
    NTSC=*y-*Y_Cf;delete Y_Cf;}
if(opt==0){
    image *Cm00=fcr.horizontal(*Cm0);delete Cm0;
    image *Cm11=fcr.horizontal(*Cm1);delete Cm1;
    image *Cm=Cm00->interlace(*Cm11);delete Cm00;delete Cm11;
    NTSC=new image(*Cm,image::docopy);delete Cm;}
if(opt==1){
    image *Cm=Cm0->interlace(*Cm1); delete Cm0;delete Cm1;
    image *Y_C=*y-*Cm;delete Cm;
    image *Y_C0=Y_C->deinterlace(0);
    image *Y_C1=Y_C->deinterlace(1);delete Y_C;
    image *Y_C0f=fcr.horizontal(*Y_C0);delete Y_C0;
    image *Y_C1f=fcr.horizontal(*Y_C1); delete Y_C1;
    image *Y_Cf=Y_C0f->interlace(*Y_C1f);delete Y_C1f;delete Y_C0f;
    NTSC=*y-*Y_Cf;delete Y_Cf;}
delete y;
return NTSC;

} //Fin de la routine de l'encodeur
```

// Routine du décodeur

```
image *decoder(image *ntscpa,image *ntscpr){
filter fh("notch5.flt");
filter fv("VHPver2.flt");
filter dirac(1);
image *ntscpah=fh.horizontal(*ntscpa);
image *ntscprh=fh.horizontal(*ntscpr);
image *ntscpah0=ntscpah->deinterlace(0);
image *ntscpah1=ntscpah->deinterlace(1);delete ntscpah;
image *ntscpah0_30=oblique(ntscpah0);delete ntscpah0;
image *ntscpah1_30=oblique(ntscpah1);delete ntscpah1;
image *ntscpah0_120=dirac.horizontal(ntscpah0_30);delete ntscpah0_30;
image *ntscpah1_120=dirac.horizontal(ntscpah1_30);delete ntscpah1_30;
image *ntscpah0_120v=fv.vertical(ntscpah0_120);delete ntscpah0_120;
image *ntscpah1_120v=fv.vertical(ntscpah1_120);delete ntscpah1_120;
image *ntscpatmp=ntscpah0_120v->interlace(*ntscpah1_120v);delete
ntscpah0_120v;delete ntscpah1_120v;
```

Annexe

```
image *ntscprh0=ntscprh->deinterlace(0);
image *ntscprh1=ntscprh->deinterlace(1);delete ntscprh;
image *ntscprh0_30=oblique(ntscprh0);delete ntscprh0;
image *ntscprh1_30=oblique(ntscprh1);delete ntscprh1;
image *ntscprh0_120=dirac.horizontal(ntscprh0_30);delete ntscprh0_30;
image *ntscprh1_120=dirac.horizontal(ntscprh1_30);delete ntscprh1_30;
image *ntscprh0_120v=fv.vertical(ntscprh0_120);delete ntscprh0_120;
image *ntscprh1_120v=fv.vertical(ntscprh1_120);delete ntscprh1_120;
image *ntscprtmp=ntscprh0_120v->interlace(*ntscprh1_120v);delete
ntscprh0_120v;delete ntscprh1_120v;
image *inter=*ntscprtmp-*ntscpatmp;delete ntscpatmp;delete ntscprtmp;
image *resul=multi(inter,0.5);delete inter;
return resul ;
```

}//Fin de la routine du décodeur

// Routine de la démodulation

```
image *demodulation(int n,image *input,char C){
char compl[2]={'I','Q'};
filter f1("demodul.flr");
image *input0=input->deinterlace(0);
image *input1=input->deinterlace(1);
image *evenfield=new image(*input0,image::nocopy);
image *oddfield=new image(*input0,image::nocopy);
    for(int j=0;j<input0->rows;j++){
        for(int i=0;i<input0->cols;i++){
            if(C==compl[1]){
                evenfield->pixref(i,j)=2*input0->pixread(i,j)*(sin((i*PI/2)))*cos(PI*(-n+j));
                oddfield->pixref(i,j)=2*input1->pixread(i,j)*(sin((i*PI/2)))*cos(PI*(-n+j));
            }
            else{
                evenfield->pixref(i,j)=2*input0->pixread(i,j)*(cos((i*PI/2)))*cos(PI*(-n+j));
                oddfield->pixref(i,j)=2*input1->pixread(i,j)*(cos((i*PI/2)))*cos(PI*(-n+j));
            }
        }
    }
delete input0;delete input1;

image *evenfield0=f1.horizontal(*evenfield);delete evenfield;
image *oddfield1=f1.horizontal(*oddfield);delete oddfield;
image *resu=evenfield0->interlace(*oddfield1);delete oddfield1;delete evenfield0;
return resu;
}
```

// Fin de la routine de démodulation

// Routine de matricage

```
image *matricage(image *cI,image *cQ,char C){
char cname[2]={'U','V'};
int su1=128;
image *res=new image(cI->rows,cI->cols);
    for(int j=0;j<cI->rows;j++){
        for(int i=0;i<cI->cols;i++){
            if(C==cname[0]){
                res->pixref(i,j)=1.1790 *cI->pixread(i,j)+ 0.7672 *cQ->pixread(i,j)+su1;
            }
            else{
                res->pixref(i,j)=-1.0808 *cI->pixread(i,j)+ 1.6630 *cQ->pixread(i,j)+su1;
            }
        }
    }
}
```

Annexe

```
return res;
}
image *matricageY(image *inputY){
int su2=16;
image *Y=new image(inputY->rows,inputY->cols);
for(int j=0;j<inputY->rows;j++){
for(int i=0;i<inputY->cols;i++){
Y->pixref(i,j)=1.6912*inputY->pixread(i,j)+su2;}}
return Y;
}
```

// Fin de la routine de matricage

// Routine de la conversion fréquentielle du format D1 en Format D 2

```
image *conv135143(int frame,char *input,char *comp){
image *in=new image(frame,input,comp);
int col,pp,xx,U1=35,D1=33;
float pu,xu,gm;
filter f("updown.flr");
image *C135;

if(comp=="U" || comp=="V") C135=f.ihorizontal(*in,2);
else C135=new image(*in,image::docopy);delete in;
```

```
FILE *fptr;
long double coef[601];
if((fptr=fopen("coefntsc","r"))==NULL)
printf("LE FICHER EST VIDE");
for (int n1=0;n1<=600;n1++){
fscanf(fptr,"%lf\n",&coef[n1]);}
```

```
//Passage de 13.5MHz a 14.31MHz
col=(U1*C135->cols-(U1*C135->cols)%D1)/D1;
col+=1;
image *fD2=new image(C135->rows,col);
```

```
for( int j=0;j<fD2->rows;j++)
{
for( int i=0;i<fD2->cols;i++)
{
pu=0;
for(int n=0;n<=16 ;n++)
{
xx=U1*n+(D1*i)%U1;
gm=35*coef[xx];
pp=(i*D1-(i*D1)%U1)/U1-n;
xu=C135->pixread(pp,j);
pu=pu+gm*xu;
}
fD2->pixref(i,j)=pu;
}
```

```
}delete C135;
fclose(fptr);
image *fD2c=new image(fD2->rows,col);
for( j=0;j<fD2c->rows;j++)
{
```

Annexe

```
        for( int i=0;i<fD2c->cols;i++)
        {
            fD2c->pixref(i,j)=fD2->pixread(i+8,j);}
delete fD2;
return fD2c;
```

// Fin de la routine de conversion fréquentielle

// Routine du filtrage verticale temporelle

```
image *filtmp2d(image *past,image *present){
int A=16,Ce=32;
image *tmp=new image(present->rows,present->cols);
double p0;

for(int j=0;j<present->rows;j++){

    iff((j%2)!=0){

        for(int i=0;i<present->cols;i++){

            p0=Ce*(present->pixread(i,j)-past->pixread(i,j))+A*(-present->pixread(i,j-2)-present->pixread(i,j+2)+past->pixread(i,j-2)+past->pixread(i,j+2));

            tmp->pixref(i,j)=1./128*p0;}}
        else{

            for(int i=0;i<present->cols;i++){

                p0=Ce*(present->pixread(i,j)-past->pixread(i,j))+A*(-present->pixread(i,j-2)-present->pixread(i,j+2)+past->pixread(i,j-2)+past->pixread(i,j+2));
                tmp->pixref(i,j)=1./128*p0;}}
            }
return tmp;
}
// Fin de la routine.
```

// Routine de la conversion du format D2 en format D1.

```
void convfD1(image *input,char C,image * &fD1c){
int col,p,x,U1=33,D1=35;
float py,gm,xy;
char cname[3]={'Y','U','V'};
filter dirac(1);
FILE *fptr;
long double coef[628];
if((fptr=fopen("coefntscinv","r"))==NULL)
    printf("LE FICHER EST VIDE");
for (int n1=0;n1<=627;n1++){
    fscanf(fptr,"%lf\n",&coef[n1]);}

col=int((U1*input->cols-(U1*input->cols)%D1)/D1);
image *fD1=new image(input->rows,col);
```

Annexe

```
for( int j=0;j<fD1->rows;j++)
{
    for( int i=0;i<fD1->cols;i++)
    {
        py=0;
        for(int n=0;n<=17 ;n++)
        {
            x=U1*n+(D1*i)%U1;
            gm=33*coeff[x];
            p=(i*D1-(i*D1)%U1)/U1-n;
            xy=input->pixread(p,j);
            py=py+gm*xy;
        }
        fD1->pixref(i,j)=py;
    }
}
fclose(fp);
image *fD1c1=new image(fD1->rows,col);
for( j=0;j<fD1c1->rows;j++)
{
    for( int i=0;i<fD1c1->cols;i++)
    {
        fD1c1->pixref(i,j)=fD1->pixread(i+9,j);}
}
delete fD1;
if(C==cname[1] || C==cname[2]) fD1c=dirac.horizontal(fD1c1,2);
else fD1c=dirac.horizontal(fD1c1,1);delete fD1c1;
```

}Fin de la routine

// Routine de l'extraction de la composante de luminance.

```
void luminance(image *ntscpa,image *ntscpr,image* &resu,image* &lum){
filter fv("verticalentsc.flr");
filter fcr("notch5.flr");
image *Cr1D=fcr.horizontal(*ntscpr);

image *ntscpr0=ntscpr->deinterlace(0);
image *ntscpr1=ntscpr->deinterlace(1);
image *ntscpr0v=fv.vertical(ntscpr0);delete ntscpr0;
image *ntscpr1v=fv.vertical(ntscpr1);delete ntscpr1;
image *Cr2D=ntscpr0v->interlace(*ntscpr1v);delete ntscpr0v;delete ntscpr1v;
image *diff=*ntscpr-*ntscpa;
image *ch3D=multi(diff,.5);delete diff;
image *h=adaptationver(ntscpr);
image *k=adaptationtmp(ntscpa,ntscpr);
resu=new image(Cr2D->rows,Cr2D->cols);

for(int j=0;j<resu->rows;j++){
    for( int i=0;i<resu->cols;i++){

        resu->pixref(i,j)=k->pixread(i,j)*h->pixread(i,j)*Cr1D->pixread(i,j)+k->pixread(i,j)*(1-h-
>pixread(i,j))*Cr2D->pixread(i,j)+(1-k->pixread(i,j))*ch3D->pixread(i,j);}
lum=*ntscpr-*resu;delete Cr2D;delete Cr1D;delete h;delete k;delete ch3D;
```

} Fin de la routine

Annexe

// Implémentation du filtre spatiale2

```
image *spatial2(image *input){
image *resu=new image(input->rows,input->cols);
    for(int j=0;j<resu->rows;j++){
        for( int i=0;i<resu->cols;i++){
            resu->pixref(i,j)=1./8*abs(2*(input->pixread(i,j-1)-input->pixread(i,j+1))+(input-
->pixread(i-1,j-1)+input->pixread(i+1,j-1))-(input->pixread(i-1,j+1)+input-
->pixread(i+1,j+1)));})
return resu;
}
}
```

// Implémentation du filtre spatiale 1

```
image *spatial1(image *input){
image *resu=new image(input->rows,input->cols);
    for(int j=0;j<resu->rows;j++){
        for( int i=0;i<resu->cols;i++){
            resu->pixref(i,j)=1./8*abs(4*input->pixread(i,j)+2*(input->pixread(i-1,j)+input-
->pixread(i+1,j))-2*(input->pixread(i,j-1)+input->pixread(i,j+1))-(input->pixread(i-1,j-
1)+input->pixread(i+1,j-1)+input->pixread(i+1,j+1)+input->pixread(i-1,j+1)));})
return resu;
} //routine de filtrage spatiale
```

```
image *multi(image *input,float fact){
image *resu=new image (input->rows,input->cols);
    for(int j=0;j<resu->rows;j++){
        for( int i=0;i<resu->cols;i++){
            resu->pixref(i,j)=fact*input->pixread(i,j);})
return resu;
}
}
```

```
float abs(float y){
float a=0.0;
if (y<0) a=-y;
if(y>=0) a=y;
return a;
}
}
```

// Routine de l'adaptation temporelle

```
image *adaptationtmp(image *Ntscpa,image *Ntscpr){
filter Flp("LP.flt");
pixel ye=.0,teta1=3.0,teta2=10.0;

image *NTSCpaLP=Flp.horizontal(*Ntscpa);
image *NTSCprLP=Flp.horizontal(*Ntscpr);
image *sortie=new image(NTSCpaLP->rows,NTSCpaLP->cols);
image *resu=new image(NTSCpaLP->rows,NTSCpaLP->cols);

    for(int j=0;j<sortie->rows;j++){
        for( int i=0;i<sortie->cols;i++){
            sortie->pixref(i,j)=abs(NTSCpaLP->pixread(i,j)-NTSCprLP->pixread(i,j));})
delete NTSCpaLP;delete NTSCprLP;

    for(j=0;j<resu->rows;j++){
        for( int i=0;i<resu->cols;i++){
            ye=sortie->pixread(i,j);
```

Annexe

```
if(ye<=teta1) resu->pixref(i,j)=0.0;
if(ye>teta2) resu->pixref(i,j)=1.0;
if(ye>teta1 & ye<=teta2) resu->pixref(i,j)=(ye-teta1)/(teta2+1-teta1);}
delete sortie;
//(multi(resu,255))->display("k 3 31");
return resu;
```

```
}//Fin de la routine.
```

```
// Implémentation de l'adaptation verticale.
```

```
image *adaptationver(image *Ntscpr){
float h=.0;
pixel ye;
float teta1=7.0,teta2=15.0;
filter Flp("LP.fit");
image *ntscpb=Flp.horizontal(*Ntscpr);
image *ntscpb0=ntscpb->deinterlace(0);
image *ntscpb1=ntscpb->deinterlace(1);
image *ntscpb0S=spatial1(ntscpb0);delete ntscpb0;
image *ntscpb1S=spatial1(ntscpb1);delete ntscpb1;
image *ntsc1=ntscpb0S->interlace(*ntscpb1S);delete ntscpb0S;delete ntscpb1S;
ntscpb0=ntscpb->deinterlace(0);
ntscpb1=ntscpb->deinterlace(1);delete ntscpb;
ntscpb0S=spatial2(ntscpb0);delete ntscpb0;
ntscpb1S=spatial2(ntscpb1);delete ntscpb1;
image *ntsc2=ntscpb0S->interlace(*ntscpb1S);delete ntscpb0S;delete ntscpb1S;
image *somme=*ntsc1+*ntsc2;delete ntsc1;delete ntsc2;
image *H=new image(somme->rows,somme->cols);
```

```
for(int j=0;j<H->rows;j++){
for(int i=0;i<H->cols;i++){

ye=somme->pixread(i,j);
if(ye<=teta1) h=0.0;
if(ye>teta2) h=1.0;
if(ye>teta1 & ye<=teta2) h=(ye-teta1)/(teta2+1-teta1);
H->pixref(i,j)=h;}}delete somme;
```

```
filter fvb("binaire.fit");
image *H0=H->deinterlace(0);
image *H1=H->deinterlace(1);delete H;
image *H0v=fvb.vertical(*H0);delete H0;
image *H1v=fvb.vertical(*H1);delete H1;
image *Hb=H0v->interlace(*H1v);delete H1v;delete H0v;
image *resu=new image(Hb->rows,Hb->cols);
```

```
for(j=0;j<resu->rows;j++){
for(int i=0;i<resu->cols;i++){
```

```
if(Hb->pixread(i,j)>=1) resu->pixref(i,j)=1;
if(Hb->pixread(i,j)<1) resu->pixref(i,j)=Hb->pixread(i,j);}}delete Hb;
return resu;
```

```
}
```

```
// Implémentation du filtre Diagonale 30°
```

```
image *diagonal30(image *input){
```

Annexe

```
image *resu=new image(input->rows,input->cols);
  for(int j=0;j<resu->rows;j++){
    for( int i=0;i<resu->cols;i++){
      resu->pixref(i,j)=1./64*(50*input->pixread(i,j)+16*(input->pixread(i+2,j-1)+input-
>pixread(i-2,j+1))-9*(input->pixread(i+4,j-2)+input->pixread(i-4,j+2)));}
    return resu;
  }
```

// Implémentation du filtre Diagonale 120°

```
image *diagonal120(image *input){
image *resu=new image(input->rows,input->cols);
  for(int j=0;j<resu->rows;j++){
    for( int i=0;i<resu->cols;i++){
      resu->pixref(i,j)=1./64*(50*input->pixread(i,j)+16*(input->pixread(i+2,j+1)+input-
>pixread(i-2,j-1))-9*(input->pixread(i+4,j+2)+input->pixread(i-4,j-2)));}
    return resu;
  }
```

// Implémentation du filtre oblique

```
image *oblique(image *input){
image *resu=new image(input->rows,input->cols);
  for(int j=0;j<resu->rows;j++){
    for( int i=0;i<resu->cols;i++){
      resu->pixref(i,j)=1./8*(4*input->pixread(i,j)-(input->pixread(i+2,j)+input->pixread(i-
2,j)+input->pixread(i,j+1)+input->pixread(i,j-1)));}
    return resu;
  }
```

```
void ShowUse(char *str)
{
  printf("\n usage: %s -i input -o output -s start frame -n number of frame -e encodage
\n",str);
  exit(1);
}
```

FIN DU PROGRAMME

V: IMAGE ORIGINALE



Photo.1.

U: IMAGE ORIGINALE



V: IMAGE ORIGINALE



Photo.2.

V: IMAGE ORIGINALE

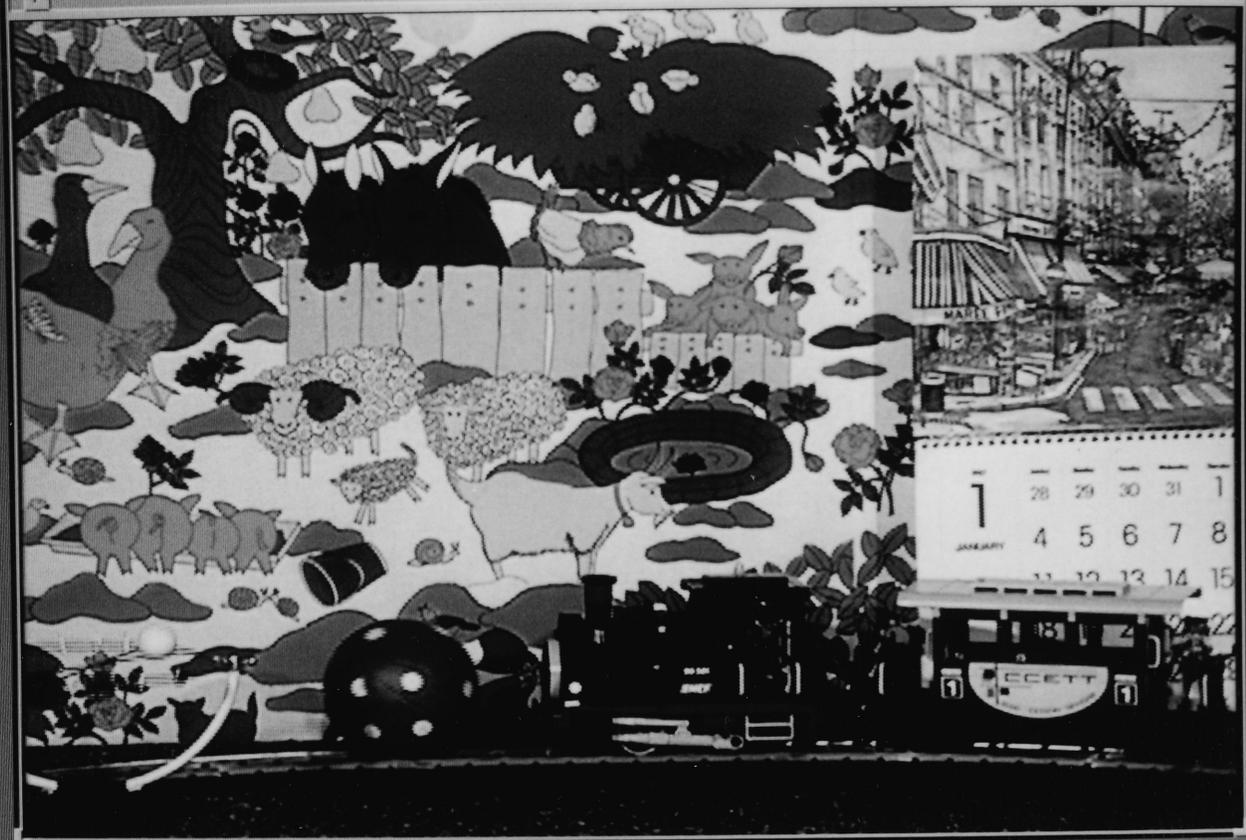


Photo.3.

U: IMAGE ORIGINALE



V: IMAGE ORIGINALE



Photo.4.

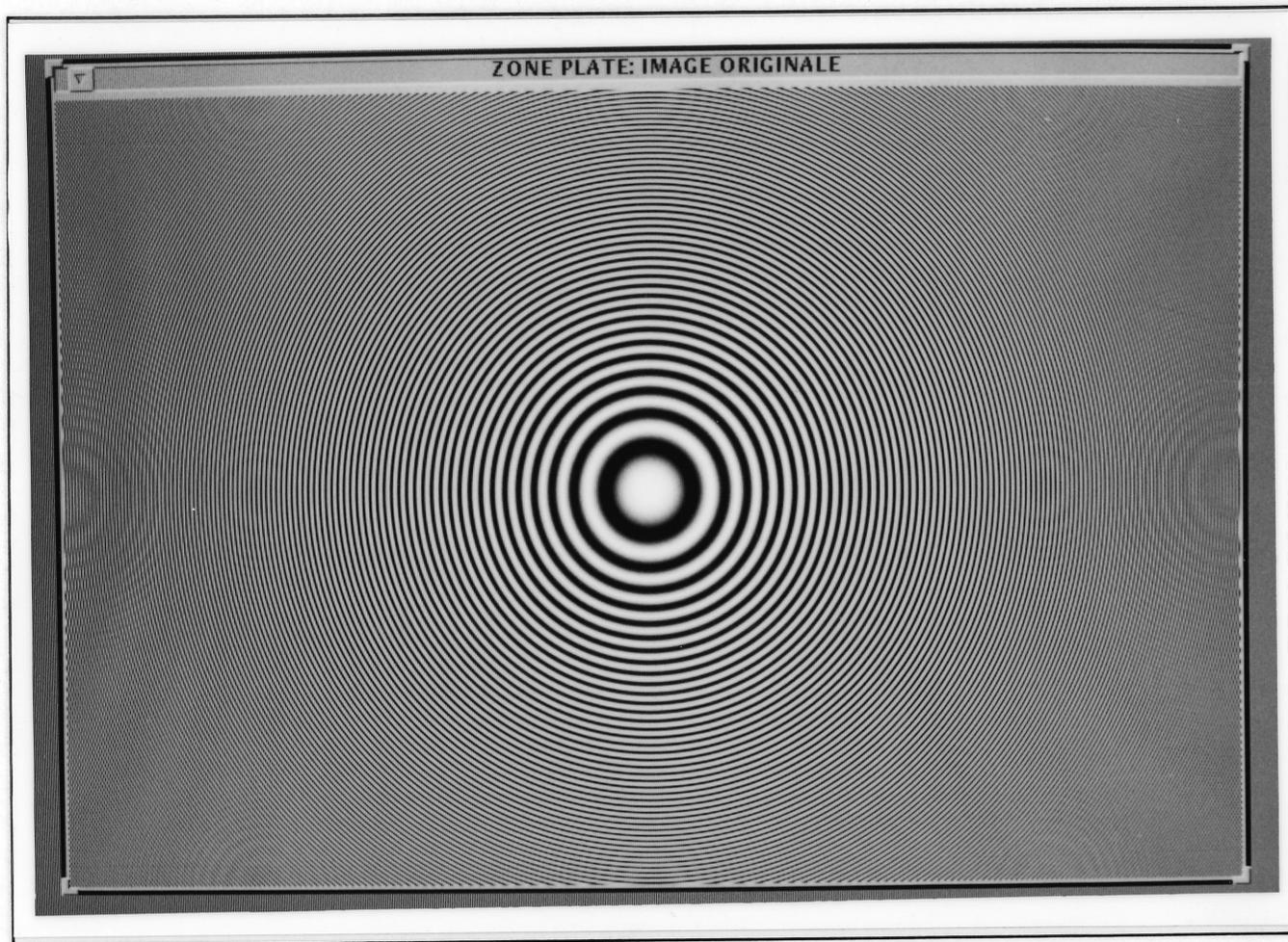


Photo.5.

Y SYSTEME NTSC (ENC,DEC)=(1D,1D)



Photo.6.

U SYSTEME NTSC (ENC,DEC)=(1D,1D)



V SYSTEME NTSC (ENC,DEC)=(1D,1D)



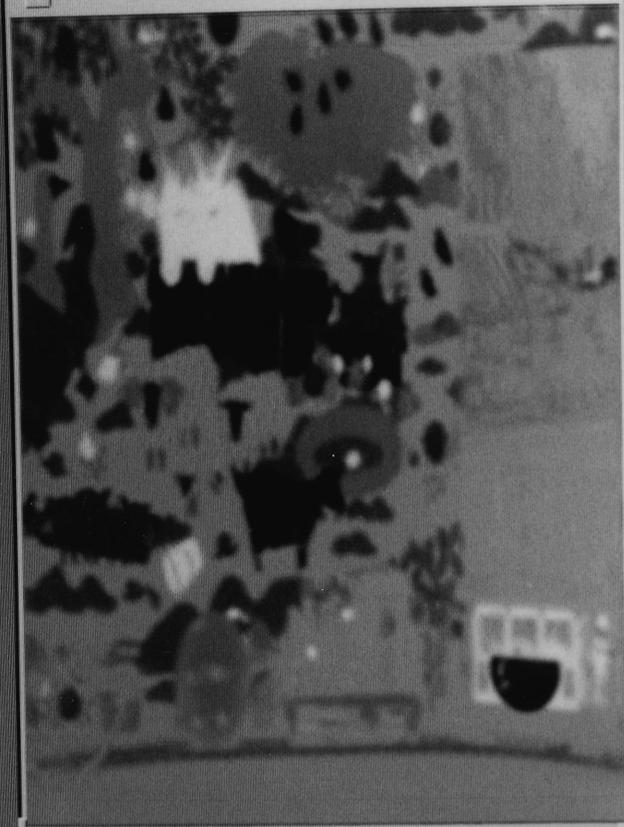
Photo.7.

Y SYSTEME NTSC (ENC,DEC)=(1D,1D)



Photo.8.

U SYSTEME NTSC (ENC,DEC)=(1D,1D)



V SYSTEME NTSC (ENC,DEC)=(1D,1D)



Photo.9.

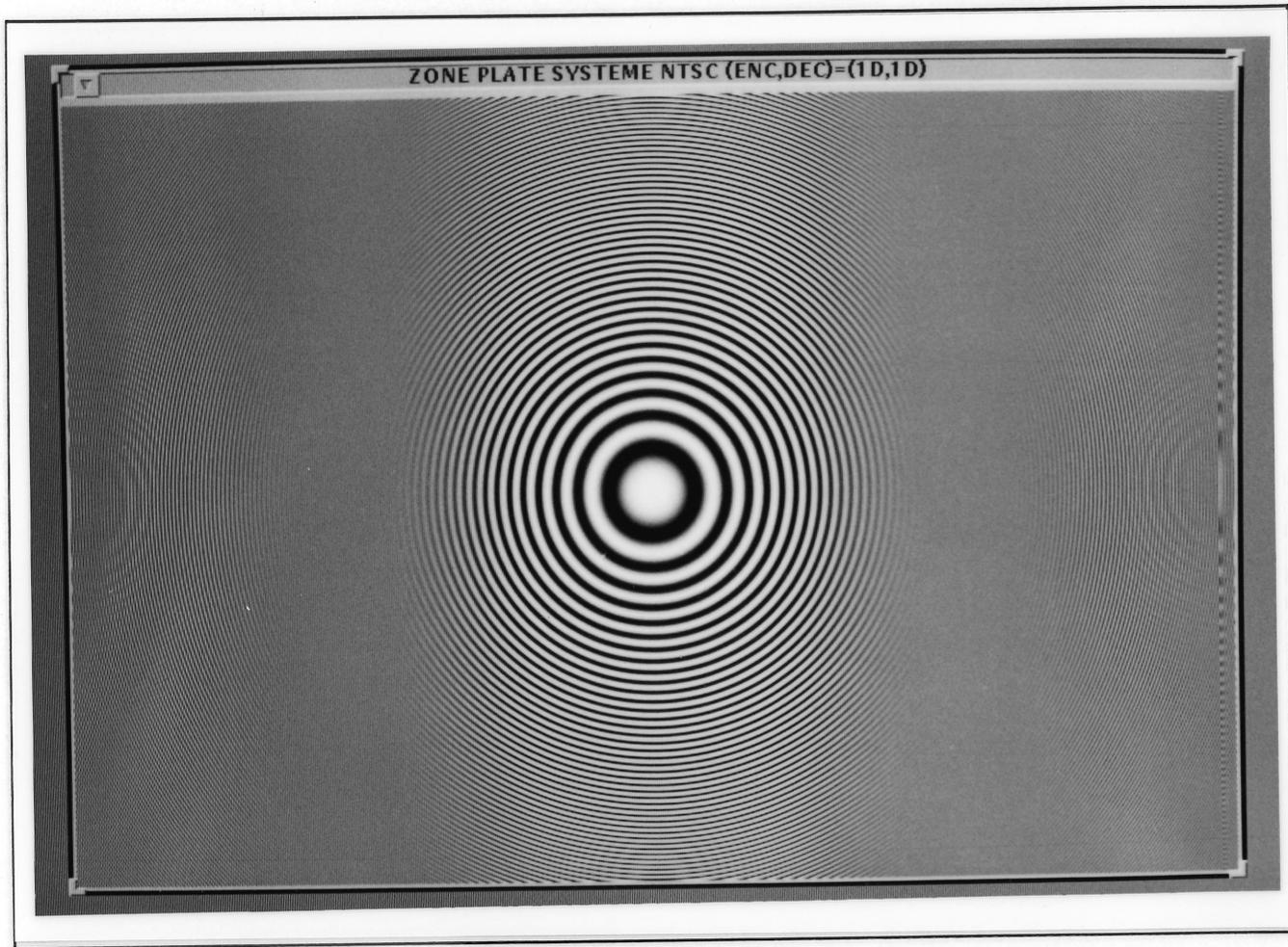


Photo.10.

CROSS LUMA: SYSTEME NTSC (ENC,DEC)=(1D,1D)

Photo. 11.

CROSS U: SYSTEME NTSC (ENC,DEC)=(1D,1D)

CROSS V: SYSTEME NTSC (ENC,DEC)=(1D,1D)

Photo. 12.

Y SYSTEME NTSC (ENC,DEC)=(1D,1D) avec ISOLUMINANCE



Photo.13.

Y SYSTEME NTSC (ENC,DEC)=(1D,1D) avec ISOLUMINANCE



Photo.14.

LUMA: SYSTEME PAL (ENC,DEC)=(1D,1D)



Photo.15.

LUMA: SYSTEME PAL (ENC,DEC)=(1D,1D)



Photo.16.

LUMA: SYSTEME PAL (ENC,DEC)=(1D,1D) AVEC ISOLUMINANCE



Photo.17.

ZONE PLATE: SYSTEME PAL (ENC,DEC)=(1D,1D)

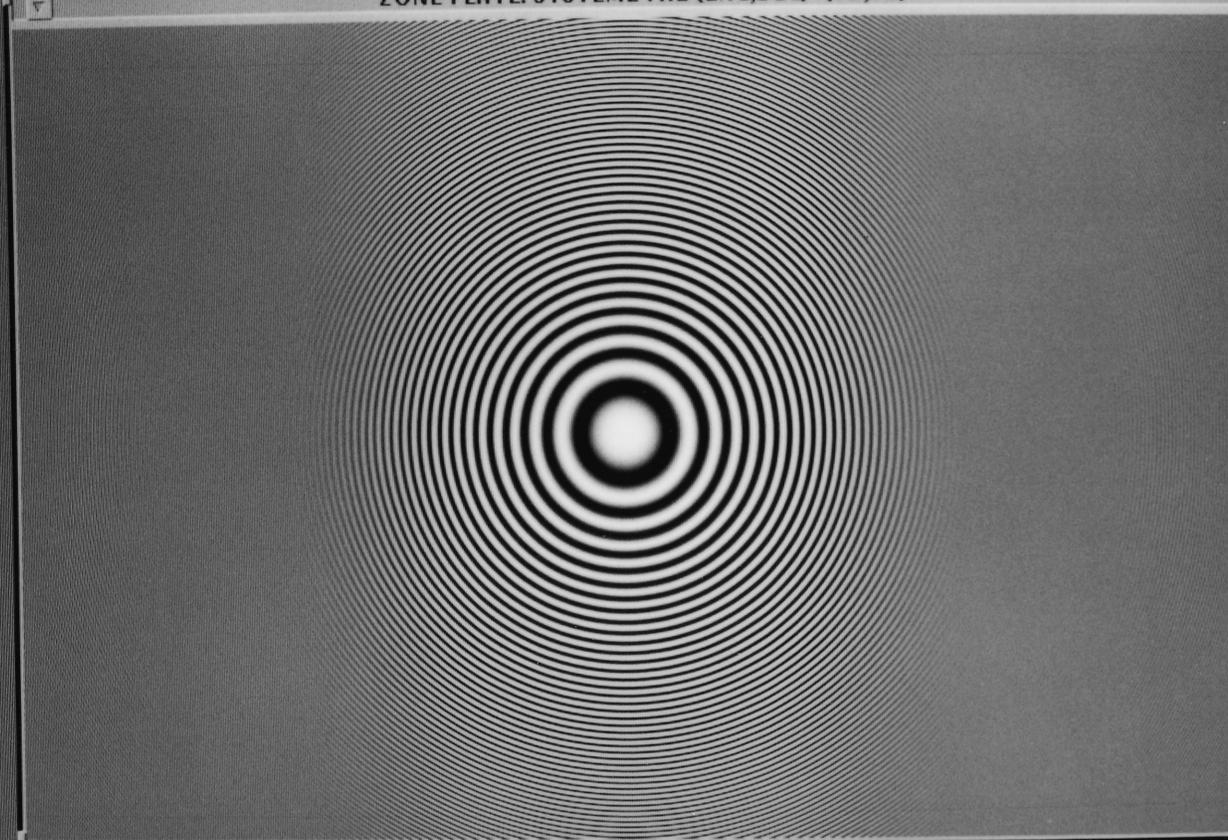


Photo.18.

CROSS LUMA: SYSTEME PAL (ENC,DEC)=(1D,1D)

Photo.19.

CROSS U: SYSTEME PAL (ENC,DEC)=(1D,1D)

CROSS V: SYSTEME PAL (ENC,DEC)=(1D,1D)

Photo.20.

Y SYSTEME NTSC (ENC,DEC)=(2D,2D)



Photo.21.

Y SYSTEME NTSC (ENC,DEC)=(2D,2D)



Photo.22.



Photo.23.

CROSS LUMA: SYSTEME NTSC (ENC,DEC)=(2D,2D)

Photo.24.

cross U: SYSTEME NTSC (ENC,DEC)=(2D,2D)

CROSS V: SYSTEME NTSC (ENC,DEC)=(2D,2D)

Photo.25.

LUMA: SYSTEME PAL (ENC,DEC)=(2D,2D)



Photo.26.

LUMA: SYSTEME PAL (ENC,DEC)=(2D,2D)



Photo.27.

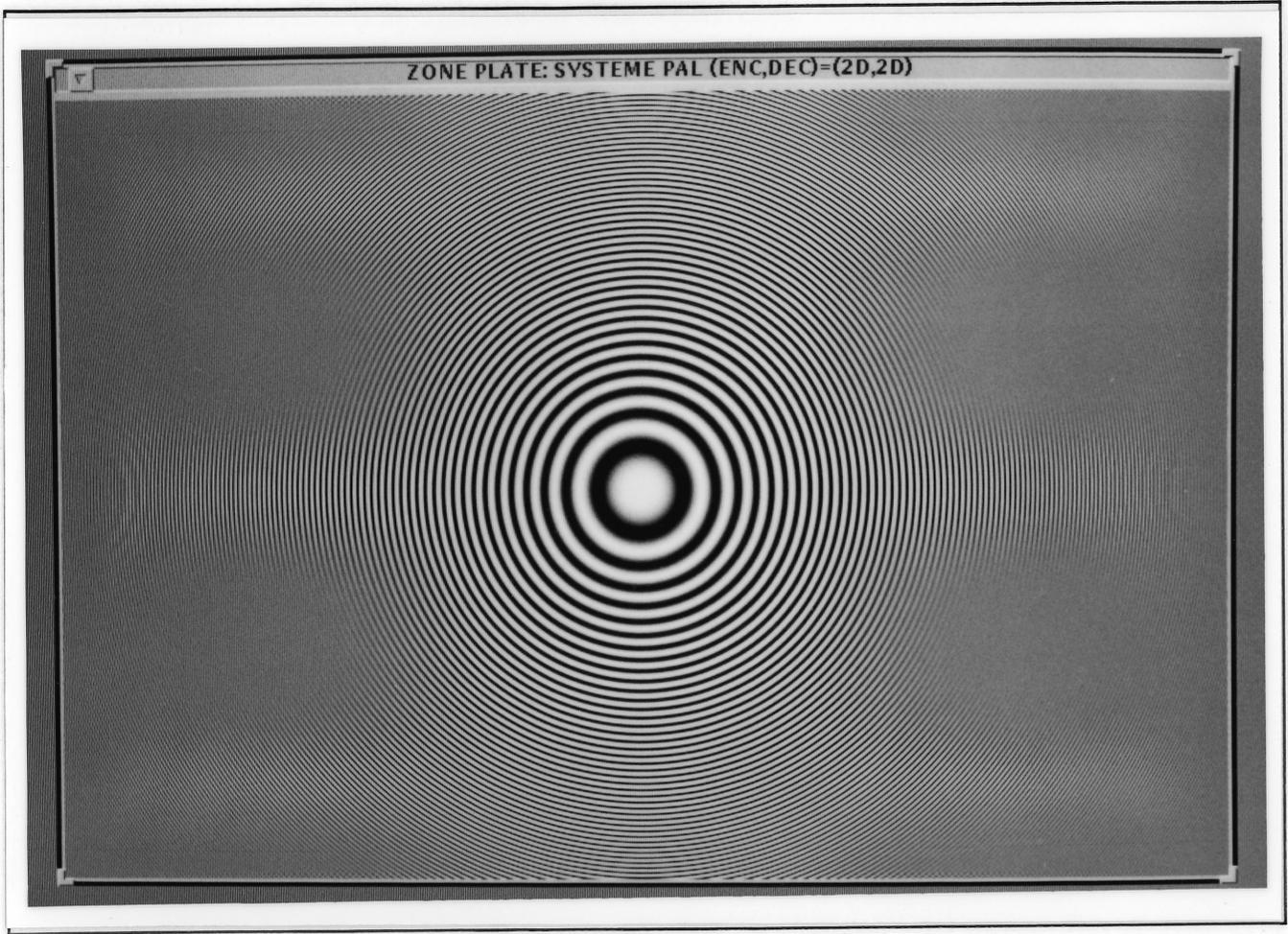


Photo.28.

CROSS LUMA: SYSTEME PAL (ENC,DEC)=(2D,2D)

Photo.29.

CROSS U: SYSTEME PAL (ENC,DEC)=(2D,2D)

CROSS V: SYSTEME PAL (ENC,DEC)=(2D,2D)

Photo.30.

Y SYSTEME NTSC (ENC,DEC)=(3D,3D)



Photo.31.

Y SYSTEME NTSC (ENC,DEC)=(3D,3D)



Photo.32.

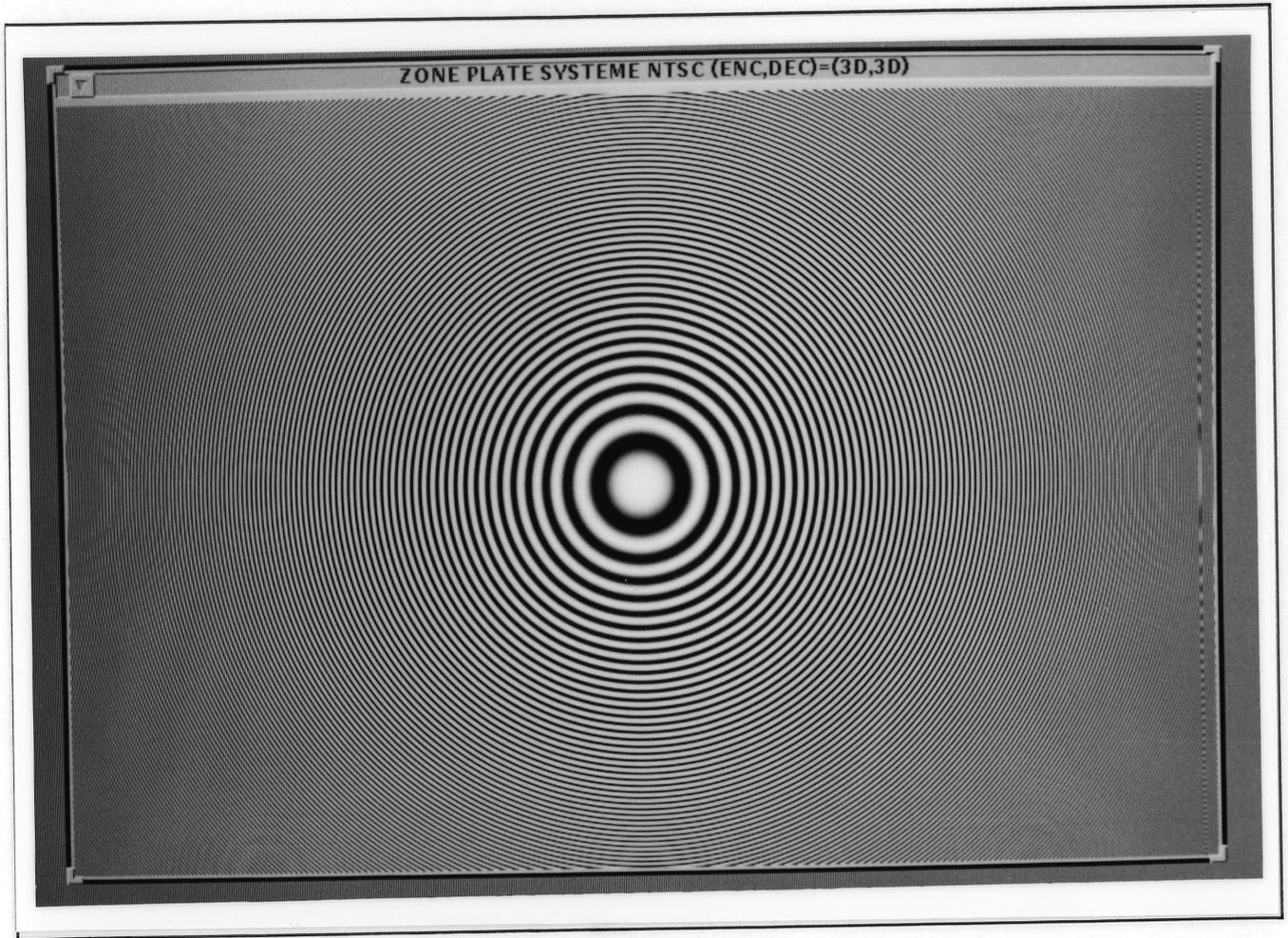


Photo.33.

CROSS LUMA: SYSTEME NTSC (ENC,DEC)=(3D,3D)

Photo. 34.

cross U: SYSTEME NTSC (ENC,DEC)=(3D,3D)

cross V: SYSTEME NTSC (ENC,DEC)=(3D,3D)

Photo. 35.

LUMA: SYSTEME PAL (ENC,DEC)=(3D,3D)



Photo.36.

LUMA: SYSTEME PAL (ENC,DEC)=(3D,3D)



Photo.37.

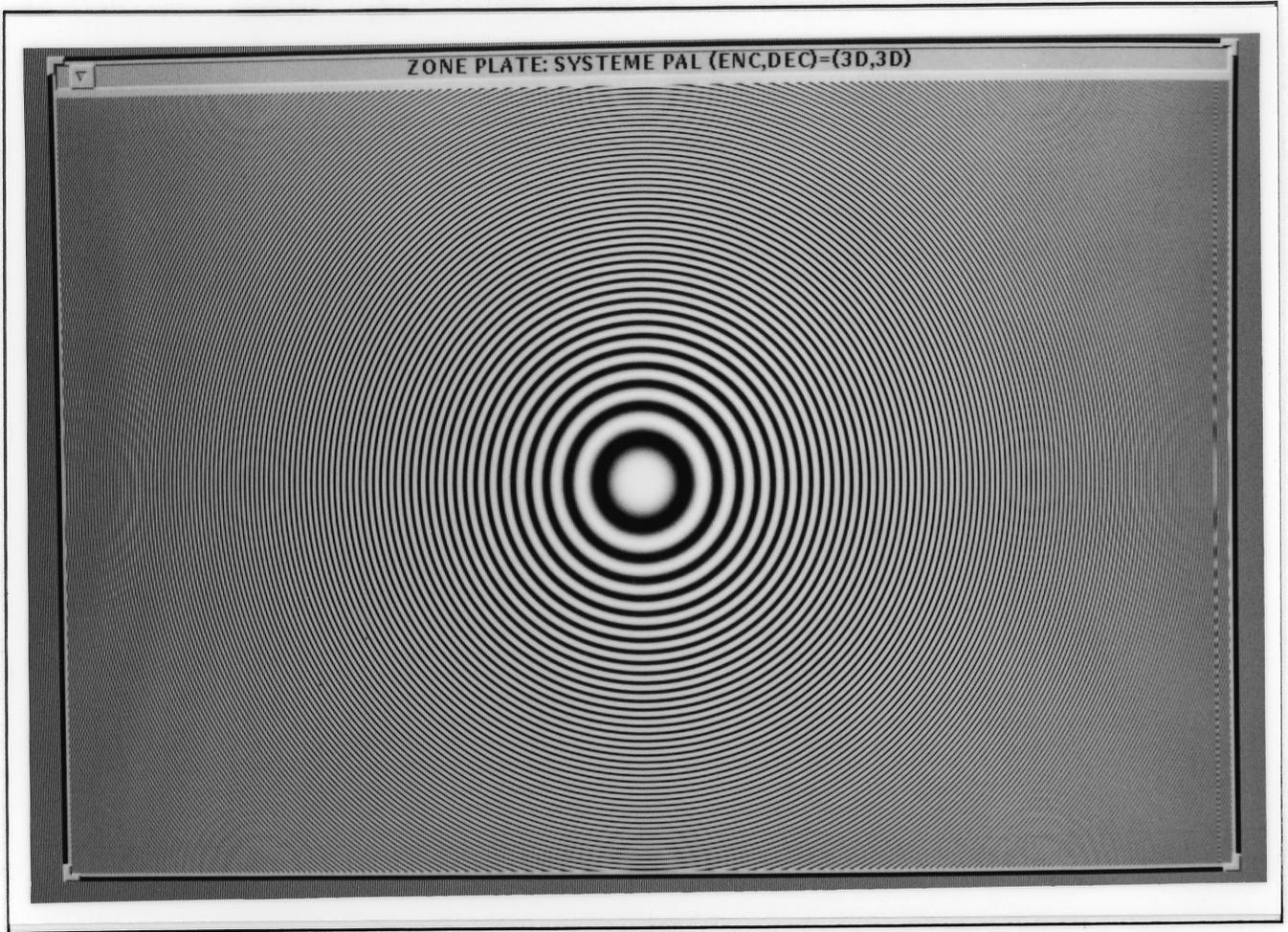


Photo.33.

CROSS LUMA: SYSTEME PAL (ENC,DEC)=(3D,3D)

Photo.39.

CROSS U: SYSTEME PAL (ENC,DEC)=(3D,3D)

CROSS V: SYSTEME PAL (ENC,DEC)=(3D,3D)

Photo.40.

LUMA: SYSTEME NTSC ADAPTATIVE



Photo.41.

V: SYSTEME NTSC ADAPTATIVE



V: SYSTEME NTSC ADAPTATIVE



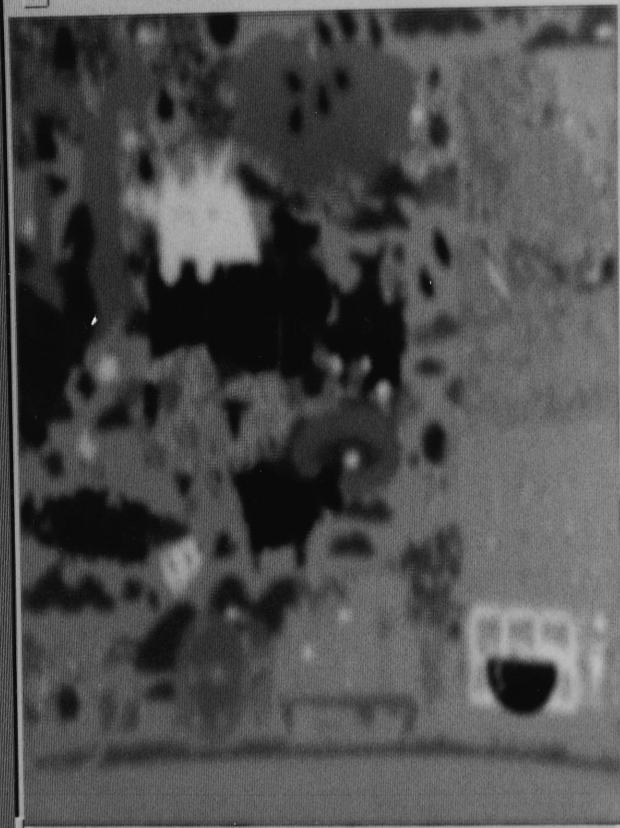
Photo.42.

LUMA: SYSTEME NTSC ADAPTATIVE



Photo.43.

U: SYSTEME NTSC ADAPTATIVE



V: SYSTEME NTSC ADAPTATIVE

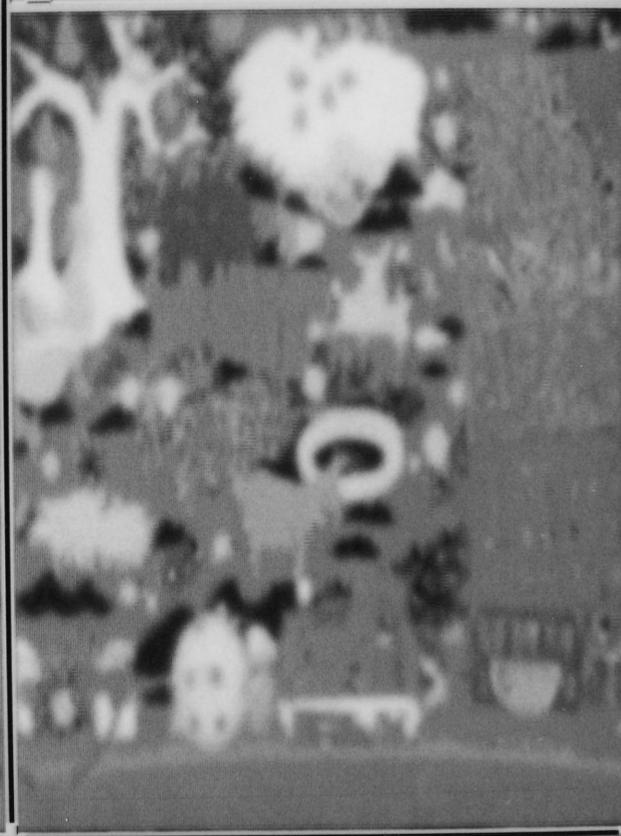


Photo.44.

3

3

3

ZONE PLATE: SYSTEME NTSC ADAPTATIVE

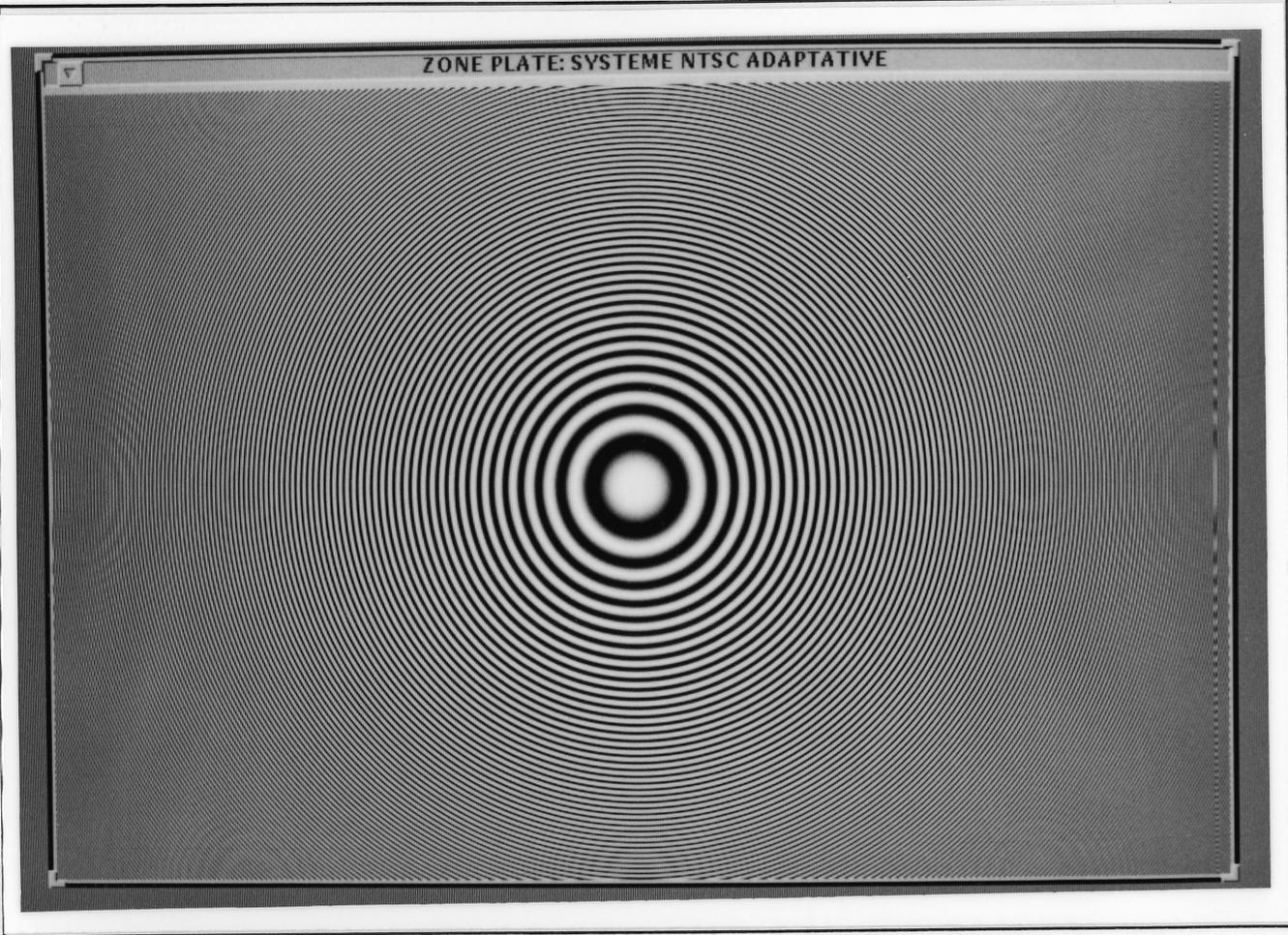


Photo.45.

CROSS LUMA: SYSTEME NTSC ADAPTATIVE

Photo.46.

CROSS U: SYSTEME NTSC ADAPTATIVE

CROSS V: SYSTEME NTSC ADAPTATIVE

Photo.47.

BIBLIOGRAPHIE

- [1] W. K. Pratt " Digital Image Processing " 2nd Edition New York : WILEY C1991
- [2] D. E. Dugeron and R. M. Mersereau, " Multi-dimensional Digital Signal Processing ". Englewood Cliffs N. J Prentice Hall 1984.
- [3] L. Goussot , " Les Systèmes de Télévision en Ondes Métriques et Décimétriques ", Dunod 1987
- [4] P. Mellusson, " Traité Théorique et Pratique de la Reception T.V ", Editions techniques et scientifiques françaises 1978
- [5] R. M. Mersereau and A. V Oppenheim, " Digital Reconstruction of Multi-dimensional Signals from Their Projection" *Proc. IEEE*, Vol 62, pp.1319-1338. Oct. 1974.
- [6] J.Watkinson, " The Art Of Digital Video " Focal Press London & Boston 1990
- [7] J. E. Ouellet et E.Dubois, " Sampling and Reconstruction of NTSC Video Signals at Twice the Color Sub carrier Frequency ". *IEEE Transactions On Communications*, Vol. Com-29, No.12, pp1823-1831 December 1981.
- [8] E. Dubois, M .S. Sabri, J. Y. Ouellet " Three -Dimensional Spectrum and Processing of Digital NTSC Color signals " . *SMPTE . J.* pp 373-378 April 1982
- [9] E. Dubois, P. Faubert, " Two -Dimensional Filters for NTSC Color Encoding and Decoding" *International Broadcasting Convention*, Brington, UK, 19-23 September 1986, Conference Publication pp 252 - 255
- [10] E. Dubois, W. F. Schreiber, " Improvements to NTSC by Multidimensional Filtering " . *SMPTE. J.* pp 446-463. June 1988.
- [11] Drewery, J. O, " The Filtering of Luminance and Chrominance Signals to Avoid Cross-color in a PAL Color System" *BBC Engineering*, Sept. 1976 pp 8-39

- [12] Clarke, C. K. P. " High Quality Decoding for PAL Inputs to Digital YUV studios ".
BBC Research Department Report No. 1982/12
- [13] Clarke, C. K. P. " PAL Decoding: "Multi-dimensional filter Design for Luminance-Chrominance Separation" BBC Research Department Report No 1988/11.
- [14] Teichner, D. " Adaptive Filter Techniques for separation of Luminance and Chrominance in PAL TV Signals " *IEEE Transactions On Consumer Electronics*, Vol. CE-32, No. 3. pp 241-250 August 1986.
- [15] Teichner, D. " Three-Dimensional Pre- and Past-Filtering for PAL TV Signals ".
IEEE Transactions on Consumer Electronics, Vol.34, No.1, pp 205-227 February 1988
- [16] Y. Nakajima, Y. Mizutani, S. Tamaki, H. Ito, T. Kazezawa, T. Murakama, " Improvement of Picture Quality for NTSC and PAL System by Digital Signal Processing ". *IEEE Transactions on Consumer Electronics*, Vol. CE, No.4 pp 642- 653 November 1985.
- [17] T. Fukinuki, Y. Hirano, " Extended Definition TV Fully compatible With Existing Standards ". *IEEE Transactions on Communications*, Vol.Com-32, No.8 pp 984-953 August 1984.
- [18] M. Tsinberg, E. Fisch, " A System for Artifact Free NTSC Encoding and decoding ".
IEEE transaction on Consumer Electronics, Vol. CE-32, No.3 pp 228-237 August 1986.
- [19] T. Yoshida, K. Ashizawa, A. Nishihara." Design and Implementation of Multi-dimensional Y_C Separation Filters for NTSC signals ". *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.2, No.4 pp 373-1992 December 1992.
- [20] Faroudja. Y. C; Roisen. J: " Improving NTSC to Achieve Near RGB Performance ".
SMPTE. Journal, Vol.96, No8, August 1987 pp 750-761.
- [21] Faroudja. Y. C " NTSC and Beyond " *IEEE Transactions on Consumer Electronics*. Vol 34. No. 1 February 1988.

- [22] Ledinh. C.T ; A. Sahed; C. Tremblay " A Non Linear Luminance Compensation Network for Gamma Corrected TV Signals " *IEEE Transactions on Consumer Electronics*. Vol 39, No 4 November 1993 pp 806-814.
- [23] Ledinh. C.T " Separable Diamond Shaped Multidimensional Filters for Composite Video Encoding/Decoding Application " *United states Patent* N.5,150.202 September 1992.
- [24] R. E. Crochiere " A General Program to Perform Sampling Rate Conversion of Data by Rational Ratio " *Bell Laboratories* Murray Hill, NJ 07974
- [25] Digital Signal Processing Committee " Programs for Digital Signal Processing " *IEEE Acoustics, Speech and Signal Processing Society IEEE Press*.
- [26] CCIR Recommendation 601: Encoding Parameters of Digital Television for Studios (4:2:2 & 4:4:4)
- [27] CCIR Recommendations 656 (BC/11 & BD/11) : Interfaces for Digital Component video Signals in 525-line and 625-line television systems (parallel and serial)