



L.I.S.A. EA4014
62, av Notre Dame du Lac
49000 Angers - France
Tél. : +33 (0)2.41.22.65.60



C.H.R.U. d'Angers
Pôle Imagerie
49000 Angers - France
Tél : +33 (0)2.41.22.65.00

Stage de MASTER2 Signaux et Images en Biologie et Médecine

Sujet du stage :

Compression contour-texture des images médicales

Dates : du 5 février au 31 août 2007

Auteur : Emmanuel SZPIEGA

Responsable de stage : Christine CAVARO-MENARD

Synthèse

SUJET

La compression a pour but de résoudre les problèmes grandissants de transmission et d'archivage de données numériques. Dans le domaine de l'imagerie médicale, ces problèmes sont liés à la quantité toujours croissante d'images acquises quotidiennement de part l'amélioration technique des systèmes d'acquisition et le développement de l'imagerie en médecine.

L'algorithme de compression que nous avons réalisé est basé sur un codage de type contour-texture et a été adapté aux images médicales dont la particularité est d'être monochromes et plus riches en niveaux de gris que les images naturelles.

RESULTATS

Le système de compression que nous avons développé est réversible, c'est à dire que l'image finale est un duplicata exact de l'image initiale. Le fond de l'image étant souvent inutile au diagnostic, une version du système de compression (compression quasi-exacte) permet de ne pas coder ce fond, afin d'obtenir un meilleur taux de compression.

Trois étapes constituent notre processus de codage : une étape de segmentation de l'image en régions homogènes, une étape de codage des contours et enfin une étape de codage de la texture de chaque région.

La segmentation est obtenue après classification floue des pixels (Fuzzy C-means généralisée) en un nombre de classes prédéfini par l'utilisateur.

Le codage des contours est réalisé en utilisant l'algorithme de Freeman ou le codage JBIG.

Le codage de la texture de chaque région est un codage prédictif. Il en résulte une matrice "erreur de prédiction" caractérisée par une intensité moyenne centrée en zéro et un écart à la moyenne très faible.

Un codage selon l'algorithme de Huffman est alors effectué pour former un train binaire contenant les contours et la texture ("erreur de prédiction") de l'image.

Le système de compression développé donne de bons résultats si nous nous intéressons à des images synthétiques géométriques. Cependant, les taux de compression obtenus sur des IRM cérébrales ne sont pas ceux attendus. Pour cela nous avons décomposé l'image en trois composantes (la structure géométrique, la texture et le bruit) selon la méthode développée par Aujol.

La programmation du système de compression a été réalisée sous l'environnement MATLAB. Le processus a été testé sur des IRM cérébrales codées sur 16 bits signés.

Remerciements

Avant de débiter ce rapport, je souhaite remercier toutes les personnes qui ont eu un rôle plus ou moins important dans le bon déroulement de mon stage d'initiation à la recherche.

Je tiens donc à remercier le Professeur Jean-Jacques LE JEUNE, responsable du pôle Imagerie du CHU d'Angers, de m'avoir permis de réaliser ce stage dans son service et Monsieur Jean-Louis FERRIER, directeur du LISA de m'avoir accueilli dans son laboratoire.

Je tiens également à remercier Madame Christine CAVARO-MÉNARD, maître de conférences à l'université d'Angers, pour sa gentillesse et pour m'avoir permis d'effectuer ce stage.

J'adresse aussi un grand merci à toutes les personnes du laboratoire qui m'ont aidées dans la réalisation de ce stage et aussi pour leur accueil et leur disponibilité.

Merci à mes collègues : Estelle TAUPIN, Richard SIMONNET, Pierre ETOUGH, stagiaires I3 de l'ESEO, Dorothée BESNIER, Kévin NGUYEN, stagiaires au LISA, Julien CHAIGNEAU, ingénieur de recherche en traitement d'images, Xavier BATY et Vincent ROULLIER, doctorants.

Sommaire

Introduction	9
1 Contexte de la compression contour-texture	10
1.1 La compression	10
1.1.1 But	10
1.1.2 Classification des techniques de compression	10
1.1.3 Problématique	10
1.1.4 La compression basée sur un modèle contour-texture	11
1.2 La classification	15
1.2.1 But	15
1.2.2 Classification d'une image numérique	15
1.2.3 Classification sur les niveaux de gris d'une image	15
1.2.4 Techniques de classification	16
1.2.5 La classification floue	16
1.2.6 La méthode de classification Fuzzy C-means	16
1.3 Compression basée sur une décomposition structure-texture de l'image	19
1.3.1 Travaux de Jean-François Aujol	19
1.3.2 Travaux de Jérôme Gilles	24
2 Travail réalisé	28
2.1 Système de compression mis en œuvre	28
2.1.1 Description	28
2.1.2 Résultats	35
2.2 Utilisation de la décomposition d'image pour le prétraitement	47
2.2.1 Principe	47
2.2.2 Résultats	47
2.3 Méthode de compression quasi-exact	52
2.3.1 Principe	52
2.3.2 Résultats obtenus sans prétraitement	52
2.3.3 Résultats obtenus avec prétraitement	54
Discussion	58
Bibliographie	60
A Codage et décodage des contours de l'image classifiée	61
A.1 Codage de l'image classifiée	61

A.2	Décodage de l'image classifiée	64
A.2.1	Construction d'une matrice 3D contenant toutes les régions	64
A.2.2	Recherche des intersections	65
A.2.3	Construction de l'image classifiée	66

Table des figures

1.1	Modèle de codage contour-texture.	11
1.2	Codage de Freeman.	13
1.3	Décomposition $f = u + v + w$	19
1.4	Décomposition de l'image Barbara avec $\lambda = 0.1, \mu = 60$	20
1.5	Image compressée avec l'algorithme d'Aujol(10% des coefficients pour u et v).	20
1.6	Décomposition d'une image médicale avec $\lambda = 0.1, \mu = 10, \sigma = 5$	24
1.7	Exemple de carte ν_2 (à gauche l'image initiale, à droite la fonction de pondération correspondante).	25
1.8	Décomposition d'une image médicale avec $\lambda = 0.1, \mu_1 = 10, \mu_2 = 5$, et une taille de fenêtre d'analyse de 15×15 pixels.	26
1.9	Fonction de pondération ν_2 avec une taille de fenêtre d'analyse de 15×15 pixels et sa version pyramidale $\tilde{\nu}_2$	27
1.10	Décomposition d'une image médicale (deuxième algorithme de Gilles) avec $\lambda = 0.1, \mu = 10, \mu_2 = 5$, et une taille de fenêtre d'analyse de 15×15 pixels.	27
2.1	Image géométrique ayant subi un très léger bruit gaussien.	28
2.2	Schéma général du système de compression.	29
2.3	Schéma général du système de décompression.	30
2.4	Classification par histogramme.	31
2.5	Centroides obtenus sur l'histogramme après classification.	31
2.6	Image classifiée.	32
2.7	Les 6 classes de l'image après classification.	32
2.8	Au moins un des pixels déjà transmis appartient à la même classe.	33
2.9	Aucun pixel déjà transmis n'appartient à la même classe.	34
2.10	Erreur de prédiction et histogramme de l'image géométrique.	34
2.11	Taux de compression en fonction du nombre de classes (codage de Freeman et codage JBIG).	35
2.12	Moyenne et écart type de l'erreur de prédiction en fonction du nombre de classes.	36
2.13	Schéma général des systèmes de compression LJPEG et JPEG-LS.	36
2.14	Erreur de prédiction et histogramme de l'image géométrique avec la méthode LJPEG (prédicteur 8). Taux de compression obtenu : 1.56 :1. Les autres prédicteurs donnent des taux de compression inférieurs.	37
2.15	Erreur de prédiction et histogramme de l'image géométrique avec la méthode JPEG-LS. Taux de compression obtenu : 1.64 :1.	37
2.16	Image testée et son histogramme.	39
2.17	Taux de compression en fonction du nombre de classes.	40
2.18	Nombre total de régions en fonction du nombre de classes.	40
2.19	Les classes de l'image médicale après classification (initialisation du nombre de classes à 16).	41
2.20	Moyenne et écart type de l'erreur de prédiction en fonction du nombre de classes.	42
2.21	Pourcentage d'octets de la trame de données réservés au codage contour et texture en fonction du nombre de classes avec le codage de Freeman et le codage JBIG.	42

2.22	Erreur de prédiction et histogramme de l'image médicale avec la méthode propre. Taux de compression obtenu : 1.87 :1.	44
2.23	Erreur de prédiction et histogramme de l'image médicale avec la méthode LJPEG (prédicteur 8). Taux de compression obtenu : 2.47 :1.	44
2.24	Erreur de prédiction et histogramme de l'image médicale avec la méthode JPEG-LS. Taux de compression obtenu : 2.50 :1.	45
2.25	Schéma général du système de compression en utilisant la décomposition.	47
2.26	Les classes de l'image médicale après classification (avec prétraitement d'Aujol et initialisation du nombre de classes à 16).	51
2.27	Construction du masque.	52
2.28	Les classes de l'image médicale après classification (initialisation du nombre de classes à 16) et utilisation d'un masque.	54
A.1	Extraction et codage des contours.	61
A.2	Suivi et extraction des contours sur l'image géométrique.	62
A.3	Trame contenant les frontières de chaque région.	63
A.4	Détermination des points contours.	64
A.5	Formation de la matrice 3D.	65
A.6	Pas d'intersection.	65
A.7	La région X contient la région Y.	66
A.8	La région Y contient la région X.	66
A.9	Formation de l'image classifiée.	67

Liste des tableaux

1.1	<i>Prédicteurs LJPEG (Lossless JPEG)</i>	14
2.1	<i>Taux de compression, moyenne et écart type obtenus pour l'image géométrique</i> . . .	38
2.2	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale avec différentes initialisations du nombre de classes</i>	43
2.3	<i>Taux de compression, nombre d'octets du train binaire nécessaire pour coder l'information contour/texture, moyenne et écart type obtenus pour l'image médicale</i>	46
2.4	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec prétraitement d'Aujol) avec différentes initialisations du nombre de classes</i>	48
2.5	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec prétraitement de Gilles) avec différentes initialisations du nombre de classes</i>	49
2.6	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec le deuxième algorithme de prétraitement de Gilles) avec différentes initialisations du nombre de classes</i>	50
2.7	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée avec différentes initialisations du nombre de classes</i> . . .	53
2.8	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée (avec prétraitement d'Aujol) avec différentes initialisations du nombre de classes</i>	55
2.9	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée(avec prétraitement de Gilles) avec différentes initialisations du nombre de classes</i>	56
2.10	<i>Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec le deuxième prétraitement de Gilles) avec différentes initialisations du nombre de classes</i>	57
A.1	<i>Formation de l'image classifiée</i>	66

Introduction

En imagerie médicale, le développement des systèmes d'acquisition a engendré une augmentation des données acquises quotidiennement et pose un problème dans la réalisation pratique des PACS (Picture Archiving and Communication Systems). La compression (en réduisant le volume représentatif des données) est une alternative de faible coût à l'augmentation systématique des capacités des lignes de transmission et des supports de stockage.

Les algorithmes de compression sont nombreux et chaque type a ses spécificités et sera plus ou moins adapté à une image ou à une vidéo donnée. Les formats JPEG et JPEG2000 les plus utilisés en compression d'images, sont basés respectivement sur des transformations globales cosinus et ondelettes. Or en imagerie médicale, chaque région a une importance plus ou moins grande dans le processus diagnostique et la compression peut être adaptée suivant la région. C'est pourquoi nous avons développé et testé une méthode basée sur un codage contour-texture permettant de coder chaque région indépendamment des autres. Dans notre projet, nous nous sommes limités au développement d'un algorithme de compression réversible de type contour-texture pour ne pas avoir à évaluer la qualité de l'image. L'évaluation des performances du système est alors uniquement basée sur l'étude du taux de compression obtenu. Ces taux de compression ont été comparés à ceux issus des formats de compression réversibles tels que LJPEG et JPEG-LS.

Dans le premier chapitre, nous présenterons les principes généraux de la compression et plus spécifiquement ceux de la compression contour-texture, de la classification floue et de la décomposition d'une image suivant trois composantes.

Dans le second chapitre, nous présenterons tout d'abord le système de compression réversible développé et les améliorations de ce système en utilisant une étape de prétraitement et un masque. Nous analyserons les résultats obtenus et les problèmes rencontrés pour conclure sur les perspectives du projet.

Chapitre 1

Contexte de la compression contour-texture

1.1 La compression

1.1.1 But

L'objectif de la compression est de résoudre les problèmes de transmission et d'archivage des données numériques. Ces problèmes sont liés au volume très important d'informations numériques acquises quotidiennement en imagerie médicale. Cependant la réduction de ce volume ne doit pas entraîner de modifications dans l'interprétation qualitative des images et la valeur des paramètres représentant l'état de l'organe étudié.

1.1.2 Classification des techniques de compression

Nous pouvons classer les méthodes de compression suivant deux catégories :

- les méthodes de codage pixel ou syntaxiques : l'information est analysée pixel par pixel, et la redondance de l'information contenue dans l'image est exploitée afin d'optimiser le codage ;
- les méthodes de codage global ou sémantiques : l'information est codée suivant les caractéristiques globales de l'image comme les contours et la texture de l'image.

Les méthodes de compression peuvent être également qualifiées de réversible ou irréversible.

Les méthodes réversibles produisent un duplicata exact de l'image originale. Ce critère est primordial dans le domaine médical.

Les méthodes irréversibles, quant à elles, permettent de ne garder que la partie la plus informative contenue dans l'image initiale. Ces méthodes donnent des taux de compression bien plus élevés que les méthodes réversibles. Dans le domaine médical, elles peuvent être utilisées seulement si elles ne modifient pas l'interprétation diagnostique.

1.1.3 Problématique

Les principaux formats de compression sont JPEG basé sur la transformée en cosinus discrète et JPEG2000 basé sur les ondelettes.

Le problème de JPEG et JPEG2000 est qu'ils sont basés sur des transformations codant l'ensemble de l'image. Or, chaque région anatomique (en imagerie anatomique) ou fonctionnelle (en imagerie fonctionnelle) est plus ou moins importante dans le processus diagnostique.

C'est pourquoi nous nous sommes intéressés à une méthode de compression contour-texture où chaque zone est délimitée par ses contours et où chaque région peut être codée indépendamment des

autres par des méthodes réversibles ou irréversibles selon leur importance dans le processus diagnostique. Certaines zones telles que le fond de l'image peuvent également ne pas être transmises, le reste de l'image est alors récupéré grâce au codage des contours des autres régions.

Dans la littérature, la compression basée sur le codage de régions et de texture s'applique essentiellement en vidéo MPEG4 [1, 2, 3] et en 3D [4]. Le développement de cette méthode dans la vidéo n'est pas adaptée au domaine médical car cette norme fait l'hypothèse d'une méthode de segmentation capable de distinguer tous les objets constituant la scène. Pour les images 3D, un maillage est nécessaire et cette méthode est totalement adaptée au domaine médicale dans le sens où on peut intégrer plusieurs volumes (possibilité d'avoir plusieurs rendus volumiques). Cette méthode travaille essentiellement sur le maillage en effectuant des déformations.

1.1.4 La compression basée sur un modèle contour-texture

La compression basée sur un modèle contour-texture représente l'image de manière symbolique. Nous obtenons alors une mosaïque de régions adjacentes définies chacune par :

- une composante contour correspondant aux frontières de la région ;
- une composante de texture caractérisant l'intérieur de la région.

La figure 1.1 nous montre les fonctions nécessaires à la mise en œuvre d'un modèle de codage contour-texture.

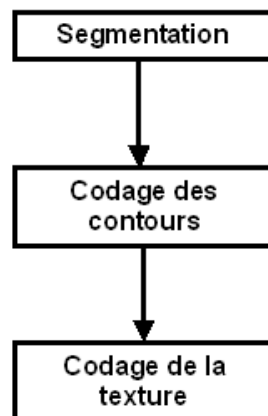


FIG. 1.1 – Modèle de codage contour-texture.

Tout d'abord, l'image initiale est segmentée de façon à obtenir une image binaire des points contour. Les régions doivent être fermées et correspondent aux objets de la scène.

Ensuite, les points contours sont codés à l'aide d'un descripteur des contours. Ce descripteur peut être soit exact (codage de Freeman), soit approximé par des segments de droite ou des polygones.

Enfin, la texture est codée par un descripteur. Pour un codage réversible, les résultats optimaux en terme de taux de compression sont obtenus par codage prédictif [5].

1.1.4.1 La segmentation

La segmentation est généralement utilisée pour l'aide au diagnostic (calcul de surface, volume,...).

Le but de la segmentation est de diviser une image en plusieurs régions contenant chacune une activité homogène, c'est à dire une texture spécifique. Les frontières de chaque région correspondent aux contours.

Il existe trois grandes classes de segmentation :

- la segmentation par détection de contours : les contours obtenus sont alors souvent non fermés ;
- la segmentation par modèles déformables : ce processus est long et complexe et nécessite une initialisation du modèle proche du résultat recherché ;
- la segmentation par tests d'homogénéité de région : le contour est alors défini comme la frontière d'une région homogène.

La nature des images (liée au système d'acquisition et à l'élément étudié) demande des techniques de segmentation spécifiques : il n'existe donc pas de méthode de segmentation universelle. En fait, la méthode de segmentation est souvent choisie en fonction des caractéristiques de l'image, des primitives que l'on souhaite extraire (contours, régions, textures), de l'analyse à effectuer et des contraintes d'exploitation.

Dans notre cas, les méthodes par test d'homogénéité de région apparaissent les plus pertinentes pour un processus automatique et rapide. En effet, la détection de contours nécessite une fermeture de contours complexe et les modèles déformables nécessitent une intervention de l'utilisateur dans le sens où il doit initialiser un modèle proche du résultat souhaité.

Segmentation par test d'homogénéité de régions

Le principe de ce type de segmentation est de trouver les liens entre les pixels selon le degré de similitude de leur activité. Les pixels peuvent ensuite être reliés entre eux suivant leur connexité.

Il existe deux principales méthodes de segmentation par test d'homogénéité de région : la segmentation par croissance des régions et la segmentation par histogramme.

Segmentation par croissance des régions

Le principe de ce type de segmentation est de choisir un certain nombre de pixels connexes dont l'activité peut être caractérisée par une fonction. Ensuite, un pixel voisin est inclus dans cette région si son activité est suffisamment proche. La fonction est mise à jour et le processus est répété tant que tous les points n'appartiennent pas à une région ou à un contour.

Le problème de ce type de segmentation est que le résultat dépend de la position du premier point et du critère d'acceptation. Ces derniers sont difficiles à définir.

Segmentation par histogramme

Le principe de la segmentation par histogramme est de déterminer les vallées de l'histogramme et d'attribuer un pixel dans une classe si son intensité est comprise entre les deux seuils caractérisant cette classe.

La précision de cette segmentation dépend de l'aptitude des seuils à définir des régions spatialement homogènes.

Les processus de segmentation par histogramme les plus connus sont :

- les seuillages : ils peuvent être binaires (non applicable sur des images bruitées), adaptatifs (si les pics de l’histogramme ne sont pas séparables), ou multi-niveaux (si les objets de l’image sont différenciables par leurs niveaux de gris) ;
- l’algorithme EM (Expectation-Maximisation) : cette méthode permet d’approximer l’histogramme par une somme de gaussiennes (équation 1.1) :

$$\hat{H}(I_x) = \sum_{n=1}^N \alpha_n \cdot G(I_x, \mu_n, \sigma_n) \quad (1.1)$$

- la classification floue : ce processus de segmentation sera celui utilisé dans notre algorithme car les images médicales peuvent être considérées comme des données imparfaites (voir la section 1.2).

1.1.4.2 Le codage des contours selon Freeman

Le principe du codage de Freeman est de décrire les arcs formant un contour par une suite de vecteurs de taille élémentaire et de direction choisie dans un ensemble fini. La direction d’un arc est codée sur 3 bits par une valeur comprise entre 0 et 7 dans le sens trigonométrique.

La figure 1.2 nous montre à gauche, la notation de Freeman suivant la direction du parcours, et à droite, un exemple de contour à coder.

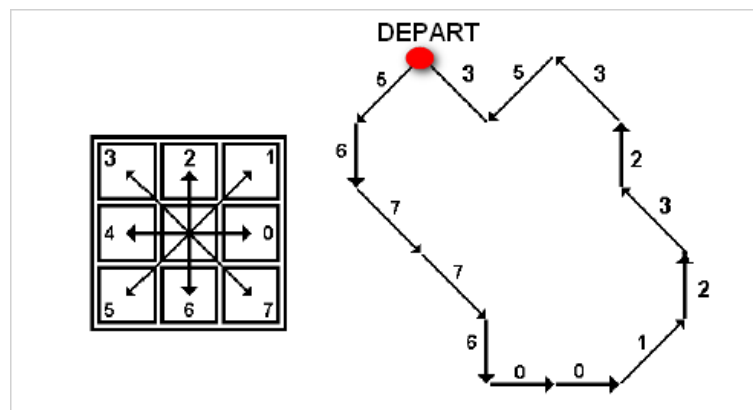


FIG. 1.2 – Codage de Freeman.

La description du contour représenté sur la figure s’effectue de la manière suivante :

- tout d’abord, les coordonnées du point de départ sont transmises. Ce point se situe le plus en haut à gauche dans la matrice ;
- ensuite, le codage des contours est réalisé en se déplaçant dans le sens trigonométrique. Ici nous obtenons comme vecteur de Freeman : 5 6 7 7 6 0 0 1 2 3 2 3 5 3

Nous utiliserons le différentiel de Freeman consistant à centrer le contenu du vecteur de Freeman autour de zéro.

1.1.4.3 Le codage prédictif

Ce type de codage est également connu sous le nom de MICD (Modulation par Impulsion Codée Différentielle) ou DPCM (Differential Pulse Coded Modulation).

Le principe du codage prédictif est de prédire la valeur de l'amplitude d'un pixel par la connaissance des pixels voisins. Ceci peut se faire grâce à la forte corrélation spatiale présente dans une image. Nous transmettons ainsi la différence entre la valeur réelle et la valeur prédite qui est appelée "Erreur de prédiction".

Deux standards de compression réversible sont basés sur le codage prédictif : le codage LJPEG avec prédicteur fixe et le codage LOCO-I avec prédicteur adaptatif.

Le standard LJPEG

Le standard LJPEG (Lossless JPEG) propose plusieurs prédicteurs représentés dans le tableau 1.1. Un seul prédicteur est choisi pour toute l'image et ce choix doit minimiser l'erreur quadratique moyenne (voir tableau 1.1).

Contexte :				1	A
				2	B
				3	C
	C	B	D	4	A+B-C
	A	X		5	A+(B-C)/2
				6	B+(A-C)/2
				7	(A+B)/2
				8	(A+B+C+D)/4

TAB. 1.1 – Prédicteurs LJPEG (Lossless JPEG)

Le standard LOCO-I

Le standard LOCO-I (Low complexity Lossless Compression for Images) également appelé JPEG-LS utilise un prédicteur adaptatif (équation 1.2) :

$$X = \begin{cases} \min(A, B) & \text{si } C \geq \max(A, B) \\ \max(A, B) & \text{si } C \leq \min(A, B) \\ A + B - C & \text{sinon} \end{cases} \quad (1.2)$$

Contexte :	C	B
	A	X

Par rapport au standard LJPEG, cette méthode a pour avantage de s'adapter aux changements de régions dans les zones de contours.

1.2 La classification

1.2.1 But

Les méthodes de classification ont pour but de regrouper les éléments d'un ensemble en un nombre restreint de classes.

La qualité de la classification peut être jugée sur la base des deux critères suivants :

- les classes générées doivent être les plus différentes possibles les unes des autres vis-à-vis de certaines caractéristiques ;
- chaque classe doit être la plus homogène possible vis-à-vis de ces caractéristiques.

Le premier critère traduit une inertie inter-classe la plus grande possible, et le deuxième critère traduit une inertie intra-classe la plus petite possible.

1.2.2 Classification d'une image numérique

La classification consiste donc à classer des individus en exploitant leur similarité, contrairement à l'approche frontière qui recherche les dissimilarités.

Les individus d'une image numérique peuvent représenter :

- les pixels : les attributs permettant de quantifier la similarité entre deux pixels peuvent être soit ponctuels comme par exemple le niveau de gris du pixel étudié, soit calculés sur un voisinage du pixel ;
- une sous-image de taille fixe sur laquelle sont calculés les attributs (l'image est alors découpée en plusieurs sous blocs) ;
- une région obtenue par une méthode de segmentation manuelle ou automatique.

1.2.3 Classification sur les niveaux de gris d'une image

La classification utilisée pour ce projet considère les individus comme les pixels eux-mêmes. Les classes sont construites à partir des niveaux de gris des pixels.

Chaque pixel appartient à une seule classe et est classifié indépendamment de ses voisins. En effet, les critères de similarité sont en général des différences d'intensité qui ne tiennent pas compte de la corrélation spatiale des pixels.

Une image 512x512 contient 262 144 pixels. Si nous choisissons un nombre de classes égal à 16, nous aurons besoin de calculer 4 194 304 distances pour chaque itération, ce qui demande un temps de calcul très important. C'est pourquoi, en traitement d'images, la classification s'effectue souvent sur l'histogramme de l'image.

Dans notre processus, nous avons pris le niveau de gris comme attribut pour la classification. La méthode est ainsi qualifiée de monodimensionnelle, par opposition aux méthodes multidimensionnelles qui prennent en compte plusieurs attributs.

1.2.4 Techniques de classification

Les différentes techniques de classification peuvent être qualifiées de :

- technique automatique ou non automatique : les classes sont obtenues au moyen d’algorithmes formalisés dans le cas des techniques automatiques, et par des méthodes subjectives ou visuelles faisant appel à l’initiative d’un expert dans l’autre cas ;
- technique hiérarchique et non hiérarchique : des suites de partitions en classes de plus en plus vastes sont produites pour les méthodes hiérarchiques alors que pour les méthodes non hiérarchiques, une partition en un nombre fixé de classes est directement créé ;
- technique de classification supervisée ou non ;
- technique floue ou statistique ;

1.2.5 La classification floue

L’algorithme de classification que nous avons utilisé est basé sur la logique floue. La logique floue est une théorie mathématique prenant en compte l’incertitude¹ et l’imprécision²[6]. Elle a été introduite par Zadeh et s’appuie sur la théorie des sous-ensembles flous. La modélisation d’un sous ensemble d’un univers donné est réalisé avec des fonctions d’appartenance. La logique floue est une extension de la théorie des ensembles classiques. Zadeh a montré que cette théorie des sous-ensembles flous se réduit à la théorie des sous-ensembles classiques dans le cas où les fonctions d’appartenance considérées prennent des valeurs binaires (c’est à dire 0 ou 1).

La logique floue se distingue de la logique booléenne puisqu’elle permet à une condition d’être en un autre état que "vrai" ou "faux", elle donne des degrés de vérité.

Elle possède des applications dans de nombreux domaines :

- le contrôle automatique ;
- la prédiction d’évènements ;
- l’aide à la décision ;
- l’interprétation d’images ;
- ...

Cette théorie est adaptée à l’imagerie médicale car les images peuvent être considérées comme des données imparfaites. En effet, les systèmes d’acquisition donnent des mesures incertaines et les renseignements humains sont imprécis.

1.2.6 La méthode de classification Fuzzy C-means

1.2.6.1 Principe

La méthode de classification Fuzzy C-means a été développée par Dunn en 1973 et approuvée par Bezdek en 1981. Elle est basée sur la minimisation de la fonction objective suivante (équation 1.3) :

$$J_m(U, V, Y) = \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^m \cdot d_{ik}^2 \quad (1.3)$$

avec :

- $Y = (y_1, y_2, \dots, y_n)$, l’ensemble des données à analyser ;

¹L’incertitude est un coefficient apporté au fait qu’une proposition soit vraie ou fausse

²L’imprécision est relative au contenu d’une proposition

- $V = (v_1, v_2, \dots, v_c)$, l'ensemble des centroïdes (c étant le nombre de classes) ;
- $U = (\mu_{ik})$, une matrice $c \times n$ où μ_{ik} est l'appartenance du pixel y_k à la classe i ;
- $m > 1$, le degré de fuzzyfication, plus m augmente, plus les classes se chevauchent ;
- d_{ik} , la distance euclidienne du pixel y_k au centroïde v_i .

1.2.6.2 Algorithme

Le principe de l'algorithme de classification Fuzzy C-means est le suivant :

1. Initialisation du nombre de classes c , de la précision ϵ et des centroïdes V_i .
2. Mise à jour de la fonction d'appartenance et des centroïdes de chaque classe :
 - $\mu_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{m-1}}}$;
 - $V_i = \frac{\sum_{k=1}^n \mu_{ik} \cdot y_k}{\sum_{k=1}^n \mu_{ik}}$;
3. Si le critère de précision ϵ n'est pas atteint, l'étape 2 est répétée. Le critère de précision correspond à la quasi-immobilisation des centroïdes entre deux itérations.

1.2.6.3 Les fonctions de classification

La classification floue probabiliste considère que pour chaque point k , la somme des degrés d'appartenance doit être égale à 1 ($\sum_{i=1}^c \mu_{ik} = 1$).

Afin d'alléger l'influence de la contrainte probabiliste, l'algorithme de classification c-Means possibiliste a été proposé par Krishnapuram [7]. La fonction objective est définie suivant l'équation 1.4 :

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c \mu_{ik}^m \cdot d_{ik}^2 + \sum_{i=1}^c \eta_i \sum_{k=1}^n (1 - \mu_{ik})^m \quad (1.4)$$

où η_i est un nombre positif.

La mise à jour de la fonction d'appartenance est donnée par :

$$\mu_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{\eta_i}\right)^{\frac{1}{m-1}}} \quad (1.5)$$

avec :

$$\eta_i = K \frac{\sum_{j=1}^n \mu_{ij}^m \cdot d_{ik}^2}{\sum_{j=1}^n \mu_{ij}^m} \quad (1.6)$$

Comme défini dans [7], nous avons choisi K égal à 1. La mise à jour des centroïdes est donnée par :

$$V_i = \frac{\sum_{k=1}^n \mu_{ik} \cdot y_k}{\sum_{k=1}^n \mu_{ik}} \quad (1.7)$$

Pour notre étude nous utilisons une autre fonction objective proposée par Michel Menard [8] (classification floue généralisée) incluant l'information de Tsallis [9]. Cette information permet de prendre en compte les points situés loin des centroïdes alors que les informations classiques (Shannon, Fisher...) considèrent seulement les voisins locaux.

Approche probabiliste

Nous devons minimiser la fonction objective suivante :

$$J(U, V; Y) = \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^m \cdot d_{ik}^2 + \frac{1}{\lambda(m-1)} \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^m - \frac{1}{\lambda} \sum_{k=1}^n \gamma_k \left(\sum_{i=1}^c \mu_{ik} - 1 \right) \quad (1.8)$$

Le premier terme représente la moyenne quadratique, équivalent à la fonction objective de Bezdek (équation 1.3). Le deuxième terme est l'entropie de Tsallis quand $\sum_{i=1}^c \mu_{ik} = 1$, avec $\lambda = 1$. Le dernier terme est la contrainte probabiliste, avec γ_k un terme de pondération de la contrainte. La minimisation de l'équation 1.8 donne la distribution normalisée de Tsallis (équation 1.9) :

$$\mu_{ik} = \frac{1}{Z_m} [1 + \lambda(m-1)d_{ik}^2]^{-\frac{1}{m-1}} \quad (1.9)$$

avec :

$$Z_m = \sum_{j=1}^c [1 + \lambda(m-1)d_{jk}^2]^{-\frac{1}{m-1}} \quad (1.10)$$

L'équation de mise à jour des centroïdes est la suivante (Eq. 2.17) :

$$V_i = \frac{\sum_{k=1}^n \mu_{ik} \cdot y_k}{\sum_{k=1}^n \mu_{ik}} \quad (1.11)$$

L'algorithme généralisé probabiliste est appelé en anglais Probabilistic Generalized c-Means (FGcM).

Approche possibiliste

Pour la fonction possibiliste généralisée, le terme probabiliste $\sum_{k=1}^n \gamma_k (\sum_{i=1}^c \mu_{ik} - 1)$ est remplacé par le terme de contrainte possibiliste $\frac{1}{\lambda} \sum_{k=1}^n \mu_{ik}$. Nous obtenons ainsi :

$$J(U, V; Y) = \sum_{i=1}^c \sum_{k=1}^n \mu_{ik}^m \cdot d_{ik}^2 + \frac{1}{\lambda(m-1)} \sum_{i=1}^c \sum_{k=1}^n (\mu_{ik}^m - \mu_{ik}) - \frac{1}{\lambda} \sum_{i=1}^c \sum_{k=1}^n \mu_{ik} \quad (1.12)$$

Le deuxième terme est l'entropie de Tsallis. Le dernier terme est la contrainte possibiliste. La mise à jour de la fonction d'appartenance est donnée par :

$$\mu_{ik} = \frac{1}{[1 + \lambda(m-1)d_{ik}^2]^{-\frac{1}{m-1}}} \quad (1.13)$$

La mise à jour des centroïdes se fait comme suit :

$$V_i = \frac{\sum_{k=1}^n \mu_{ik} \cdot y_k}{\sum_{k=1}^n \mu_{ik}} \quad (1.14)$$

L'algorithme généralisé possibiliste est appelé en anglais Possibilistic Generalized c-Means (PGcM).

Dans notre algorithme nous utilisons successivement les algorithmes FGcM et PGcM. En effet, la contrainte probabiliste donne systématiquement des centroïdes (représentants de chaque classe) proche de la valeur optimale et permet d'initialiser l'algorithme possibiliste. En effet, ce dernier a besoin d'une initialisation proche des vrais centroïdes pour être stable.

1.3 Compression basée sur une décomposition structure-texture de l'image

Récemment, Aujol a proposé un algorithme de compression appliqué à des images contenant à la fois des structures et des textures. Il propose ainsi une étape de décomposition de l'image et utilise des algorithmes de compression différents pour ces deux composantes.

Cette décomposition nous a paru intéressante pour notre algorithme de compression car les structures peuvent être assimilées aux régions de l'image.

1.3.1 Travaux de Jean-François Aujol

La thèse de Jean-François Aujol [10] est un travail en mathématiques appliquées, qui aborde quelques problèmes en analyse d'images (restauration d'images, segmentation, compression) et utilise des outils mathématiques spécifiques, entre autre la décomposition d'une image en trois composantes : information géométrique, information de texture et bruit.

Le modèle proposé par Aujol décompose l'image en trois composantes : $f = u + v + w$ avec :

- La composante u contenant l'information géométrique qui peut être considérée comme une esquisse de l'image originale, c'est à dire les bords de l'image et les niveaux de gris moyens (figure 1.3).
- La composante v comportant l'information de texture de l'image : elle correspond aux détails de l'image autre que le bruit éventuel. La texture est un champs de l'image qui apparaît à l'observateur comme un domaine homogène sous certaines conditions ; nous pouvons aussi parler de phénomène pseudo-périodique pour définir une texture (figure 1.3).
- La composante w représentant le bruit présent dans l'image : $u + v$ représente alors une version restaurée de l'image f et la composante w est nulle dans le cas où l'image n'est pas bruitée (figure 1.3).

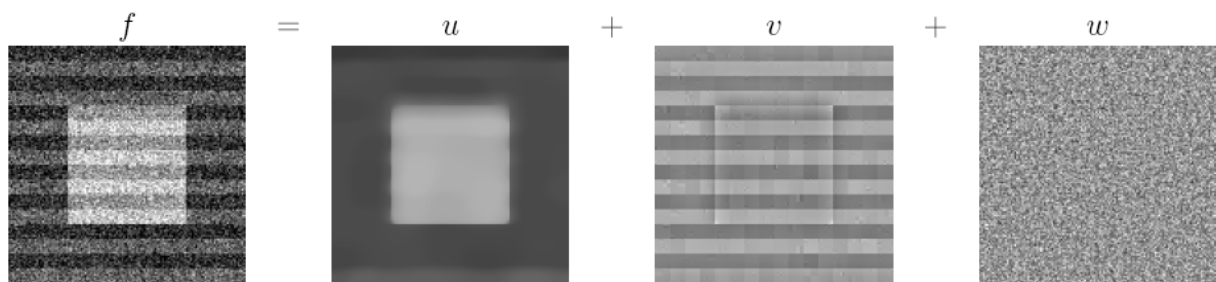


FIG. 1.3 – Décomposition $f = u + v + w$.

L'approche de J-F Aujol permet d'utiliser des espaces mathématiques adaptés à chaque composante :

- l'espace BV (Bounded Variation) des fonctions à variations bornées pour u ;
- l'espace G proche du dual de BV pour les textures représentées par v ;
- et un espace de Besov d'exposant négatif E pour le bruit.

Une étude mathématique complète est effectuée pour les différents modèles proposés et deux applications concrètes sont présentées dans [10] :

- une première en restauration d'images RSO (Radar à synthèse d'ouverture) qui ont la particularité d'être fortement bruitées (bruit de chatoiement ou "speckle") ;

- et une seconde en compression d'images naturelles (images contenant à la fois des structures et des textures).

Algorithme de compression

Le principe de l'algorithme de compression d'Aujol est de décomposer l'image originale f suivant deux composantes u (composante géométrique) et v (composante oscillante) comme le montre la figure 1.4.

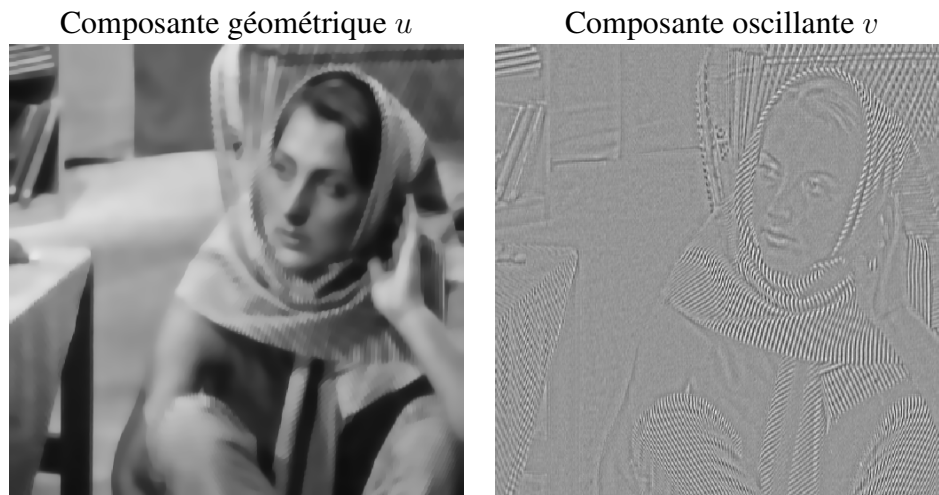


FIG. 1.4 – Décomposition de l'image Barbara avec $\lambda = 0.1$, $\mu = 60$.

Afin d'avoir une décomposition exacte, Aujol rajoute le résidu $f - u - v$ à la composante géométrique.

Il utilise un algorithme de compression par ondelettes biorthogonales pour la composante oscillante et l'algorithme "Edge Adapted" pour la composante géométrique [10].

Les résultats de compression sont comparés en terme de nombre de coefficients et non en terme de taux de compression (voir figure 1.5 et ??).



FIG. 1.5 – Image compressée avec l'algorithme d'Aujol (10% des coefficients pour u et v).

1.3.1.1 Décomposition de l'image

Quelques définitions

Une image numérique est un vecteur à deux dimension de taille $N \times N$. Soit X l'espace euclidien $\mathbb{R}^{N \times N}$ et $Y = X \times X$. X est munit du produit scalaire :

$$(u, v)_X = \sum_{1 \leq i, j \leq N} u_{i,j} v_{i,j} \quad (1.15)$$

et de la norme :

$$\|u\|_X = \sqrt{(u, u)_X} \quad (1.16)$$

Version discrète de l'opérateur gradient

Si $u \in X$ le gradient ∇u est un vecteur de Y donné par :

$$(\nabla u)_{i,j} = ((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2) \quad (1.17)$$

avec

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N \\ 0 & \text{si } i = N \end{cases} \quad (1.18)$$

et

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N \\ 0 & \text{si } j = N \end{cases} \quad (1.19)$$

La variation totale de u est donnée par :

$$J(u) = \sum_{1 \leq i, j \leq N} |(\nabla u)_{i,j}| \quad (1.20)$$

Version discrète de l'opérateur divergence

Aujol définit l'opérateur divergence par analogie avec le cadre continu en posant $div = -\nabla^*$, où ∇^* est l'opérateur adjoint de ∇ . Ainsi, pour tout $p \in Y$ $u \in X$, $(-div p, u)_X = (p, \nabla u)_Y$. L'opérateur de divergence est donné par :

$$(div(p))_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 1 < i < N \\ p_{i,j}^1 & \text{si } i = 1 \\ -p_{i-1,j}^1 & \text{si } i = N \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 1 < j < N \\ p_{i,j}^2 & \text{si } j = 1 \\ -p_{i,j-1}^2 & \text{si } j = N \end{cases} \quad (1.21)$$

Algorithme de projection de Chambolle

Aujol se sert de l'algorithme de projection de Chambolle [11] pour projeter l'image f sur les espaces G et BV .

Le calcul de $P_{\lambda K}(f)$ (projection de f sur un ensemble convexe fermé K) peut s'effectuer par une méthode point fixe. Cette méthode est un algorithme itératif consistant à initialiser $p_{i,j}^0$ à 0 et à répéter l'opération suivante un certain nombre de fois :

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(\text{div}(p^n) - f/\lambda))_{i,j}}{1 + \tau|\nabla(\text{div}(p^n) - f/\lambda)|_{i,j}} \quad (1.22)$$

Théorème [11] : si $\tau \leq \frac{1}{8}$, alors $\lambda \text{div}(p^n)$ converge vers $P_{\lambda K}(f)$ quand $n \rightarrow \infty$.

Ainsi, pour projeter sur l'espace BV , nous remplacerons λK par G_λ et pour projeter sur l'espace G , nous remplacerons λK par G_μ .

Seuillage doux en ondelettes

Aujol utilise une opération de seuillage doux des coefficients d'ondelette $\beta_{j,n}$ afin d'effectuer un débruitage d'image. Cette opération est appelée en anglais "Wavelet Soft Thresholding" (elle sera notée WST par la suite).

Les coefficients seuillés $\tilde{\beta}_{j,n}$ d'une fonction f pour un seuil δ seront donnés par :

$$\tilde{\beta}_{j,n} = \begin{cases} \beta_{j,n} - \delta & \text{si } \beta_{j,n} > \delta \\ 0 & \text{si } |\beta_{j,n}| \leq \delta \\ \beta_{j,n} + \delta & \text{si } \beta_{j,n} < -\delta \end{cases} \quad (1.23)$$

Le choix de δ est donné dans le paragraphe 1.3.1.3.

Principe de la décomposition d'Aujol

Le principe est de minimiser la fonctionnelle suivante :

$$\inf_{(u,v,w) \in X^3} (F(u, v, w)) \quad (1.24)$$

avec

$$F(u, v, w) = J(u) + J^*\left(\frac{v}{\mu}\right) + B^*\left(\frac{w}{\delta}\right) + \frac{1}{2\lambda} \|f - u - v - w\|^2 \quad (1.25)$$

Afin de résoudre l'équation 1.24, nous considérons les trois problèmes suivant :

– v et w étant fixés, on cherche u solution de :

$$\inf_{u \in X} \left(J(u) + \frac{1}{2\lambda} \|f - u - v - w\|_X^2 \right) \quad (1.26)$$

La solution de cette équation est donnée par :

$$\hat{u} = f - v - w - P_{G_\lambda}(f - v - w). \quad (1.27)$$

– u et w étant fixés, on cherche v solution de :

$$\inf_{v \in G_\mu} (\|f - u - v - w\|_X^2) \quad (1.28)$$

La solution de cette équation est donnée par :

$$\hat{v} = P_{G_\mu}(f - u - w). \quad (1.29)$$

– u et v étant fixés, on cherche w solution de :

$$\inf_{w \in E_\delta} (\|f - u - v - w\|_X^2) \quad (1.30)$$

La solution de cette équation est donnée par :

$$\hat{w} = P_{E_\delta}(f - u - v) = f - u - v - WST(f - u - v, \delta). \quad (1.31)$$

1.3.1.2 Algorithme

L'algorithme utilisé par Jean-François Aujol se divise en trois étapes :

1. Initialisation de la précision ϵ , du nombre maximal d'itérations et des trois composantes :

$$u_0 = v_0 = w_0 = 0 \quad (1.32)$$

2. Mise à jour des trois composantes :

$$w_{n+1} = f - u_n - v_n - WST(f - u_n - v_n, \delta) \quad (1.33)$$

$$v_{n+1} = P_{G_\mu}(f - u_n - w_{n+1}) \quad (1.34)$$

$$u_{n+1} = f - v_{n+1} - w_{n+1} - P_{G_\lambda}(f - v_{n+1} - w_{n+1}) \quad (1.35)$$

3. Tant que le critère d'arrêt n'est pas atteint, l'étape 2 est répétée. Le critère d'arrêt correspond soit à la quasi-immobilisation des trois composantes entre deux itérations ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$), soit au nombre maximal d'itérations.

1.3.1.3 Choix des paramètres

A notre connaissance, aucune publication ne fait état de la décomposition d'Aujol sur les images médicales. Après avoir implémenté le processus, nous avons défini de manière empirique les paramètres les plus adaptés à nos images.

Le seuil δ est choisi en supposant que le bruit dans l'image est de nature gaussienne [10]. La valeur théorique de Donoho [12, 13, 10] pour le seuillage des coefficients en ondelettes est donnée par :

$$\delta = \sigma \sqrt{2 \log(N^2)}. \quad (1.36)$$

Où σ est l'écart type du bruit et N la taille de l'image ($N \times N$ pixels).

Aujol rajoute un paramètre η compris entre 0 et 1 permettant de jouer sur la valeur du seuil :

$$\delta = \eta \sigma \sqrt{2 \log(N^2)}. \quad (1.37)$$

J'ai choisi d'utiliser l'ondelette de Haar et un nombre de trois décompositions dans mes expériences. La valeur η jouant sur le seuil sera égale à 1 par la suite. Pour une image IRM, l'écart type du bruit σ peut être défini à l'aide du fond de l'image contenant exclusivement du bruit.

Le paramètre λ permettant d'obtenir la composante géométrique doit être choisi petit, Jean-françois Aujol a pris comme valeur $\lambda \leq 1$ dans ses expériences [10]. Plus la valeur de λ est faible, plus la norme du résidu $f - u - v - w$ sera petite. J'ai pris comme valeur : $\lambda = 0.1$.

Le paramètre μ lié à la texture reste difficile à évaluer selon Aujol. Plus ce paramètre est élevé, plus v contient de l'information et donc u se trouve plus moyennée [14]. Cependant, lorsque ce paramètre est trop élevé par rapport à λ , la composante géométrique u perd un peu de sa structure géométrique. J'ai choisi après test comme valeur : $\mu = 10$.

J'ai choisi comme paramètres de projection $\tau = \frac{1}{8}$ et un nombre de 500 itérations (au-delà, il n'y pas visiblement pas de grand changement).

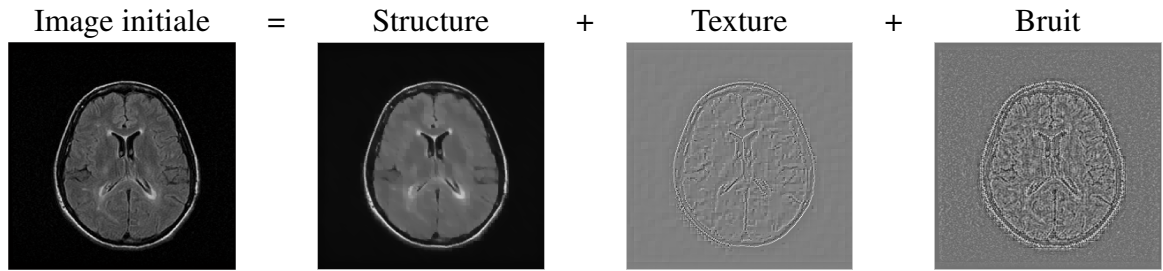


FIG. 1.6 – Décomposition d'une image médicale avec $\lambda = 0.1$, $\mu = 10$, $\sigma = 5$.

Nous remarquons que la composante texture est peu informative. En effet, il n'existe pas de texture déterministe en imagerie médicale

1.3.2 Travaux de Jérôme Gilles

Gilles propose également une décomposition d'image suivant trois composantes et propose une fusion de son algorithme avec celui d'Aujol. Les images utilisées par Gilles sont naturelles [15].

1.3.2.1 Principe de la décomposition de Gilles

Le principe est de minimiser la fonctionnelle suivante :

$$F_{\lambda, \mu_1, \mu_2}(u, v, w) = J(u) + J^*\left(\frac{v}{\mu_1}\right) + J^*\left(\frac{v}{\mu_2}\right) + B^*\left(\frac{w}{\delta}\right) + \frac{1}{2\lambda} \|f - u - \nu_1 v - \nu_2 w\|_{L^2}^2 \quad (1.38)$$

La composante géométrique u est prise dans l'espace BV alors que les composantes v (texture) et w (bruit) sont prises dans l'espace G puisqu'elles peuvent être vues comme des fonctions oscillantes.

ν_1 et ν_2 sont des coefficients de régularisation adaptatif (avec $\nu_i \in]0; 1[$). Ainsi, dans une zone de l'image contenant que du bruit, ν_1 doit être proche de 1 pour modéliser l'absence de texture et ν_2 doit être proche de 0 pour prendre en compte le bruit.

Par contre, dans une zone de l'image contenant de la texture et du bruit : ν_1 doit être proche de 0 pour renforcer la texture et ν_2 doit être proche de 1 pour modéliser l'absence de bruit.

Jérôme Gilles fait l'hypothèse que ces deux fonctions sont complémentaires et donc que $\nu_2 = 1 - \nu_1$. Il effectue un calcul de variance locale pour construire une carte des zones texturées ou non. $\nu_2(i, j)$ correspond alors à la variance calculée sur une fenêtre carrée centrée en $f(i, j)$. La taille de cette fenêtre doit être choisie afin d'obtenir le meilleur compromis entre l'estimation correcte de la variance et la localité dans l'image (figure 1.7).



FIG. 1.7 – Exemple de carte ν_2 (à gauche l'image initiale, à droite la fonction de pondération correspondante).

1.3.2.2 Algorithme

L'algorithme utilisé par Jérôme Gilles peut se résumer en trois étapes :

1. Calcul de ν_1 et ν_2 à partir de f ; initialisation de la précision ϵ , du nombre maximal d'itérations et des trois composantes :

$$u_0 = v_0 = w_0 = 0 \quad (1.39)$$

2. Mise à jour des trois composantes :

$$w_{n+1} = P_{G_{\mu_2}} \left(\frac{f - u_n - \nu_1 v_n}{\nu_2} \right) \quad (1.40)$$

$$v_{n+1} = P_{G_{\mu_1}} \left(\frac{f - u_n - \nu_2 w_{n+1}}{\nu_1} \right) \text{ avec } \mu_1 = \mu \quad (1.41)$$

$$u_{n+1} = f - \nu_1 v_{n+1} - \nu_2 w_{n+1} - P_{G_\lambda}(f - \nu_1 v_{n+1} - \nu_2 w_{n+1}) \quad (1.42)$$

3. Tant que le critère d'arrêt n'est pas atteint, l'étape 2 est répétée. Le critère d'arrêt correspond soit à la quasi-immobilisation des trois composantes entre deux itérations ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$), soit au nombre maximal d'itérations.

1.3.2.3 Choix des paramètres pour nos images

Gilles a rajouté comme paramètres dans son algorithme la taille de la fenêtre d'analyse et le paramètre μ_2 utilisé pour obtenir la composante de bruit : il faut donc le choisir plus petit que μ_1 . J'ai choisi après test comme valeur : $\mu_2 = 5$ (ce qui correspond en fait à l'écart-type du fond de l'image).

Pour avoir un bon compromis entre une estimation correcte de la variance et la localité dans l'image, j'ai choisi d'utiliser une taille de fenêtre d'analyse de 15×15 pixels. La figure 1.8 montre la décomposition d'une image médicale avec l'algorithme de Jérôme Gilles

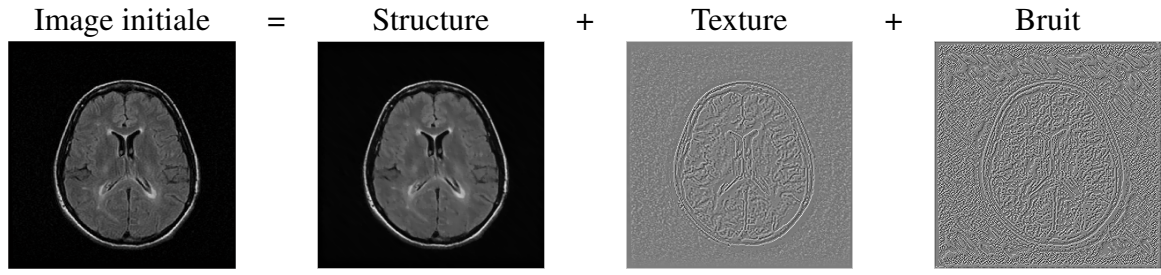


FIG. 1.8 – Décomposition d'une image médicale avec $\lambda = 0.1$, $\mu_1 = 10$, $\mu_2 = 5$, et une taille de fenêtre d'analyse de 15×15 pixels.

1.3.2.4 Extension du modèle de Gilles à celui d'Aujol

Dans sa thèse, Gilles met en évidence le fait que le bruit est mieux modélisé dans l'algorithme de Aujol mais que cet algorithme rejette une plus grande part d'information texture dans la composante bruit.

Gilles a donc proposé une unification des deux modèles dont l'algorithme numérique est le suivant :

1. Calcul de ν_1 , ν_2 et $\tilde{\nu}_2$ (version pyramidale de ν_2 comme le montre la figure 1.9) à partir de f ; initialisation de la précision ϵ , du nombre maximal d'itérations et des trois composantes :

$$u_0 = v_0 = w_0 = 0 \quad (1.43)$$

2. Mise à jour des trois composantes :

$$w_{n+1} = \frac{f - u_n - \nu_1 v_n}{\nu_2} - \frac{\lambda}{\delta \nu_2^2} WST \left(\frac{\delta \nu_2}{\lambda} (f - u_n - \nu_1 v_n), \frac{2\delta^2 \tilde{\nu}_2^2}{\lambda} \right) \quad (1.44)$$

$$v_{n+1} = P_{G_\mu} \left(\frac{f - u_n - \nu_2 w_{n+1}}{\nu_1} \right) \quad (1.45)$$

$$u_{n+1} = f - \nu_1 v_{n+1} - \nu_2 w_{n+1} - P_{G_\lambda} (f - \nu_1 v_{n+1} - \nu_2 w_{n+1}) \quad (1.46)$$

3. Tant que le critère d'arrêt n'est pas atteint, l'étape 2 est répétée. Le critère d'arrêt correspond soit à la quasi-immobilisation des trois composantes entre deux itérations ($\max\{|u_{n+1} - u_n|, |v_{n+1} - v_n|, |w_{n+1} - w_n|\} \leq \epsilon$), soit au nombre maximal d'itérations.

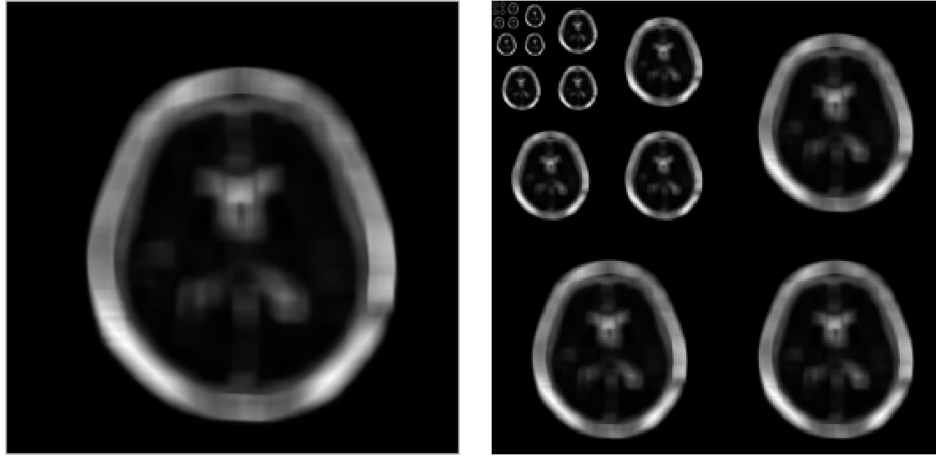


FIG. 1.9 – Fonction de pondération ν_2 avec une taille de fenêtre d'analyse de 15×15 pixels et sa version pyramidale $\tilde{\nu}_2$.

La figure 1.10 montre la décomposition d'une image médicale avec le deuxième algorithme de Jérôme Gilles.

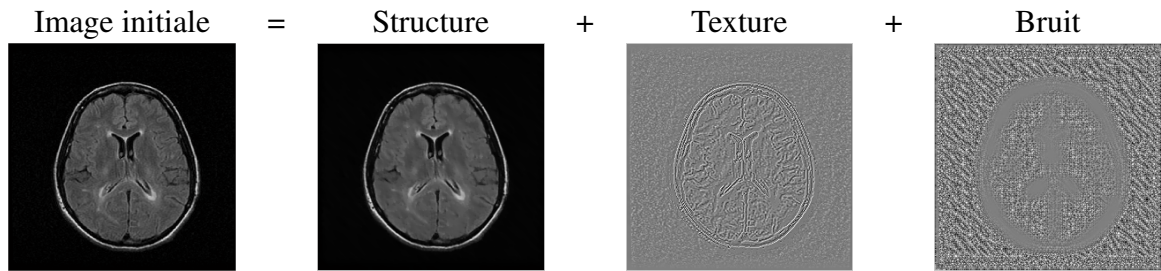


FIG. 1.10 – Décomposition d'une image médicale (deuxième algorithme de Gilles) avec $\lambda = 0.1$, $\mu = 10$, $\mu_2 = 5$, et une taille de fenêtre d'analyse de 15×15 pixels.

Nous voyons que la composante bruit de l'image médicale contient une partie (correspondant à la carte ν_2) sans "bruit". Cela est dû au fait que Gilles considère la texture et le bruit de l'image comme deux composantes complémentaires. Ainsi, carte ν_2 pondère uniquement la composante texture.

Chapitre 2

Travail réalisé

L'algorithme de compression que j'ai développé sous MATLAB, est basé sur un codage contour-texture réversible, c'est à dire que l'image reconstruite après compression est un duplicata exact de l'image initiale. Cependant le système mis en œuvre peut être modifié afin de réaliser une compression irréversible, plus performante en terme de taux de compression.

Tout d'abord, dans ce chapitre, je décrirai les phases de compression et de décompression du système de compression réversible réalisé et les résultats qui en découlent.

Ensuite, je parlerai des améliorations apportés à ce système de compression afin d'augmenter le taux de compression. Ainsi, nous présenterons les résultats obtenus d'une part, dans le cas d'un système de compression quasi-exact utilisant un masque et d'autre part, dans le cas d'une décomposition suivant trois composante structure, texture et bruit, selon les travaux d'Aujol et de Gilles.

2.1 Système de compression mis en œuvre

2.1.1 Description

Les phases de compression et de décompression du système de compression développé seront illustrées par des schémas et par les résultats obtenus sur une image géométrique simple (figure 2.1) au cours des différentes étapes. C'est une image 64x64 codée sur 8 bits non signés ayant subi un très léger bruitage gaussien.

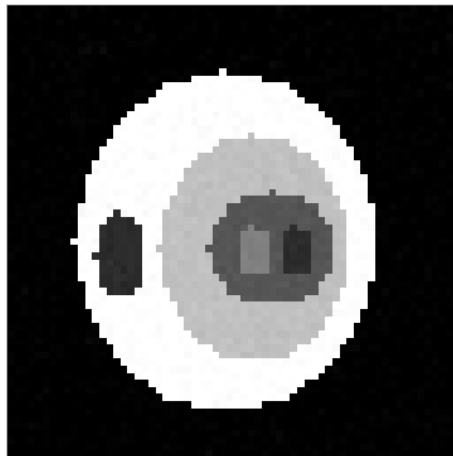


FIG. 2.1 – Image géométrique ayant subi un très léger bruit gaussien.

Les schémas synthétiques des figures 2.2 et 2.3 représentent respectivement le schéma général de compression et le schéma général de la phase de décompression du système développé.

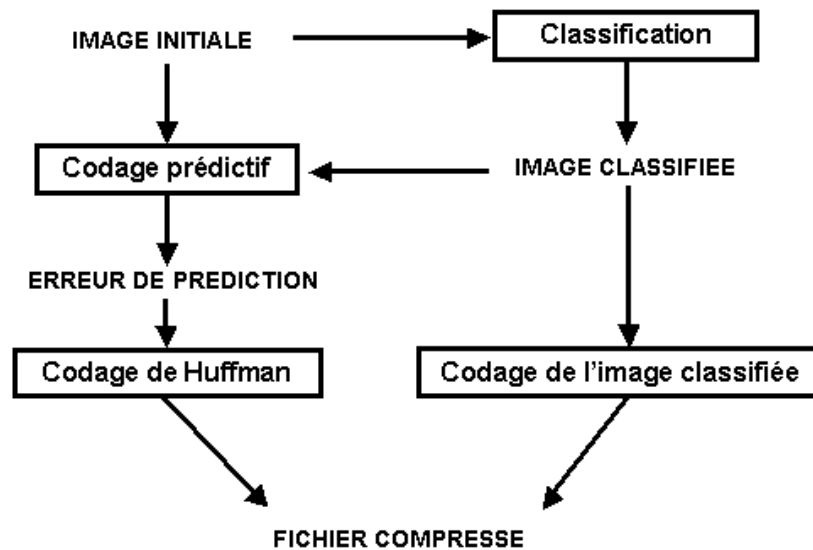


FIG. 2.2 – Schéma général du système de compression.

Pour la phase de compression (figure 2.2), l'image initiale est tout d'abord classifiée (en terme d'intensité) par les algorithmes de classification FGCM et PGCM, suivant un nombre de classes prédéfini par l'utilisateur.

L'image classifiée est ensuite codée de manière réversible, soit en utilisant le codage JBIG (Joint Bi-level Image experts Group), soit en utilisant le codage de Freeman (Annexe A.1). Le codage JBIG est une méthode de compression réversible destiné aux images binaires.

La texture de chaque région est codée de manière réversible par codage prédictif. La matrice "erreur de prédiction" caractérisée par une intensité moyenne centrée en zéro et un écart à la moyenne très faible, est codée selon l'algorithme d'Huffman. Le codage de Huffman est une méthode de type statistique utilisant une structure arborescente. Les symboles à coder sont classés par ordre de probabilité décroissante. Plus les symboles ont une forte probabilité d'apparition, plus ils seront codés par un nombre réduit de bits.

Nous obtenons ainsi un fichier compressé contenant d'une part l'information texture et d'autre part l'information contour.

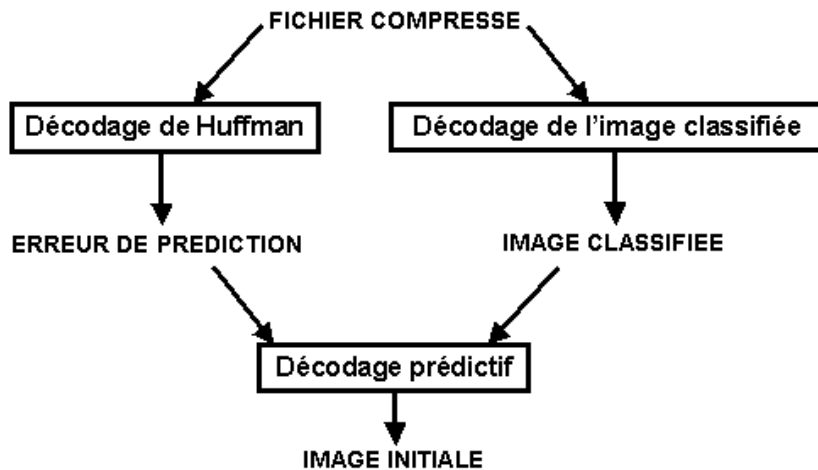


FIG. 2.3 – Schéma général du système de décompression.

La phase de décompression (figure 2.3) nécessite tout d'abord un décodage de Huffman pour reconstruire la matrice "erreur de prédiction". La reconstruction de l'image classifiée est réalisée en utilisant un décodage de Freeman ou un décodage JBIG (cela dépend de ce que nous avons utilisé durant la phase de compression).

Grâce à l'image classifiée reconstruite et à la matrice "erreur de prédiction", nous pouvons reconstruire l'image initiale à l'aide de la fonction de décodage prédictif utilisée pour le codage.

Le système de compression étant réversible, nous obtenons exactement la même image que celle représentée sur la figure 2.1.

2.1.1.1 Classification par histogramme

Le schéma de principe de l'algorithme de classification utilisé est donné figure 2.4.

Afin d'éliminer une partie du bruit, l'image originale est prétraitée par un filtrage médian. L'algorithme de classification est effectué sur l'histogramme de l'image obtenue. Comme précisé précédemment (chapitre 1.2.6.3 page 17), nous appliquons d'abord l'algorithme généralisé probabiliste, et ensuite, l'algorithme généralisé possibiliste.

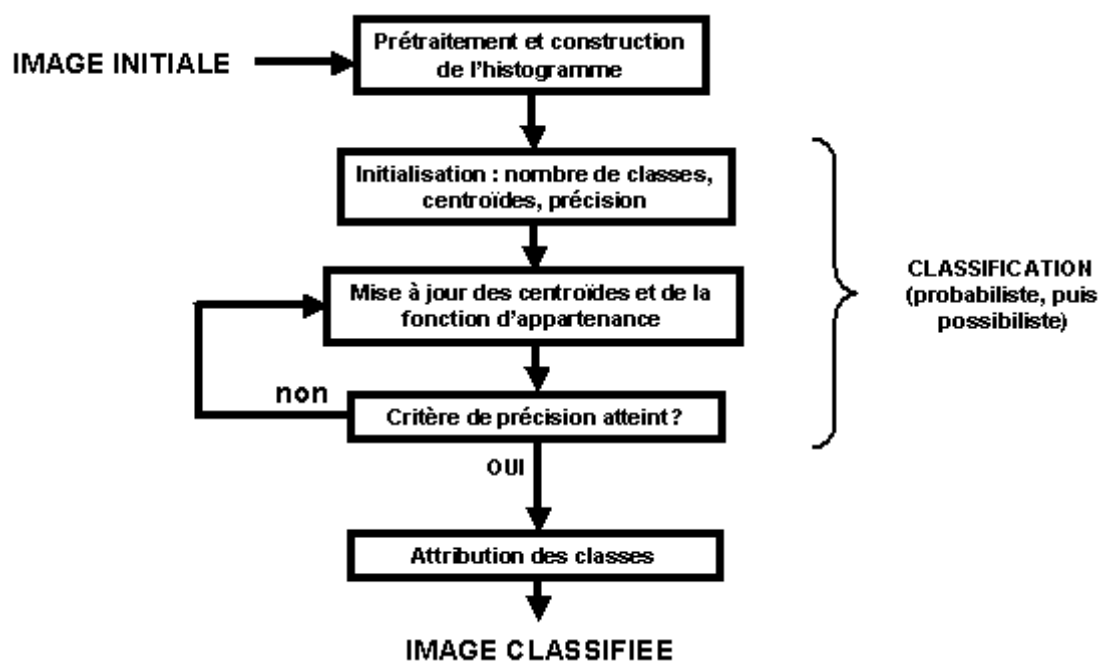


FIG. 2.4 – Classification par histogramme.

La figure 2.5 représente l'histogramme de l'image figure 2.1. Le nombre de classes pour l'algorithme de classification a été initialisé à 10.

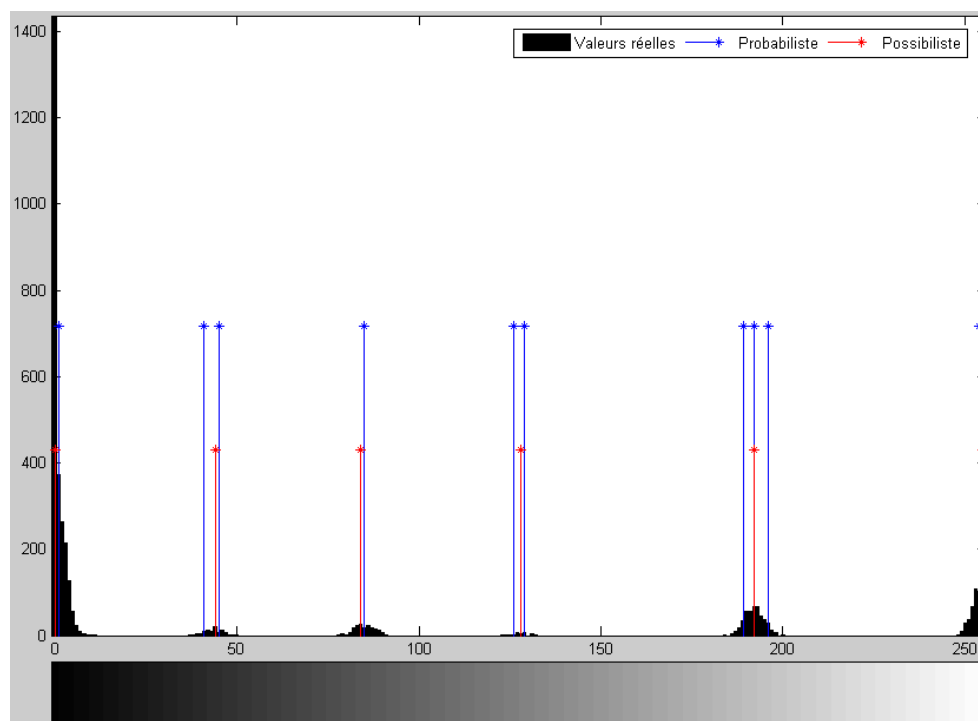


FIG. 2.5 – Centroïdes obtenus sur l'histogramme après classification.

Ces 10 centroïdes sont déplacés vers les centroïdes optimaux par l'algorithme PGCM (probabiliste) puis ajustés au mieux par l'algorithme FGCM (possibiliste). Il ne reste alors que 6 centroïdes

du fait que certains centroïdes se trouvent confondus après avoir été arrondis.

Une fois que le critère de précision est atteint, il suffit d'attribuer à chaque pixel une classe grâce à la fonction d'appartenance obtenue.

Nous obtenons alors l'image classifiée (figure 2.6).

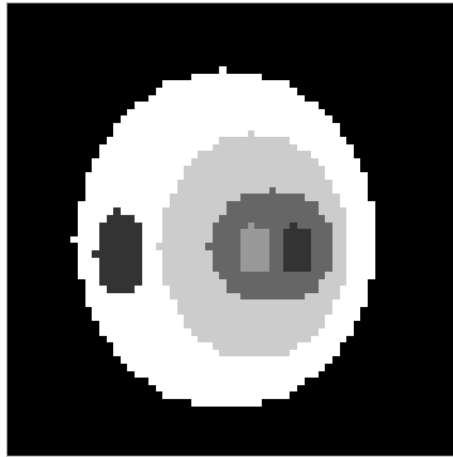


FIG. 2.6 – Image classifiée.

L'image classifiée peut être décomposée suivant 6 classes comme le montre la figure 2.7.

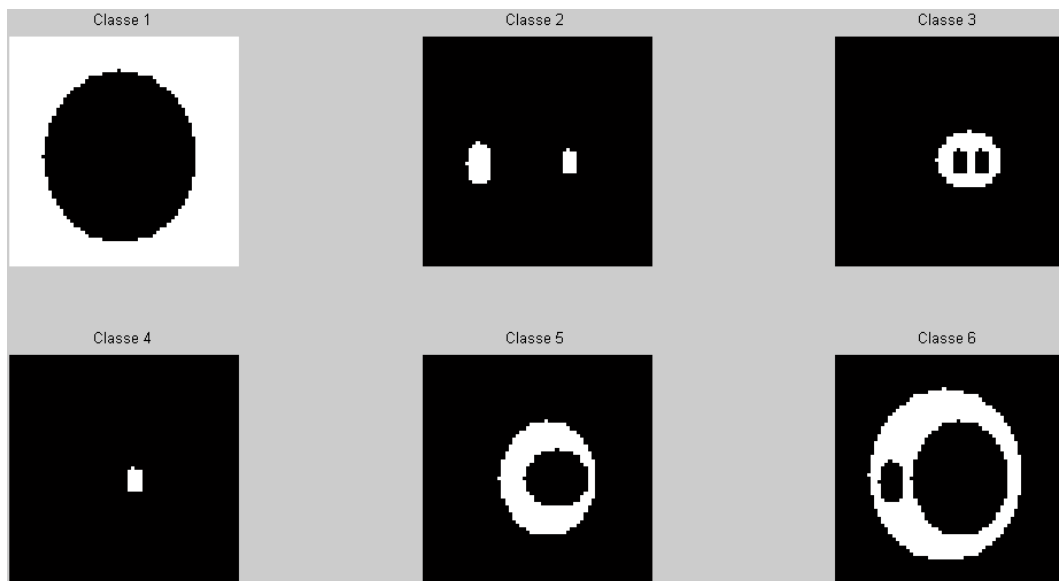


FIG. 2.7 – Les 6 classes de l'image après classification.

Nous remarquons que seule la classe 2 contient 2 régions (2 composantes non connexes), alors que toutes les autres en contiennent une seule, ce qui fait en tout 7 régions à coder.

Le codage et le décodage de l'image classifiée est présenté en Annexe A.

En utilisant le codage JBIG, chaque classe subit une binarisation. En fait, le codage JBIG est réalisé grâce à un exécutable qui convertit une image au format '.pbm' (Portable Bitmap) en une image au format '.jbg' (JBIG). Le décodage est également réalisé à l'aide d'un exécutable. Ces executables ont été créés sous Dev C++ à partir de fichiers sources téléchargés.

2.1.1.2 Codage prédictif par région

Ce type de codage fait intervenir non seulement l'image originale, mais aussi l'image obtenue après classification.

Contexte : soit les pixels voisins A, B, C, D, du pixel à coder.

C	B	D
A	X	

X sera égal :

- à la valeur moyenne des pixels voisins appartenant à la même classe (figure 2.8) *si au moins un des pixels déjà transmis appartient à la même classe* ;

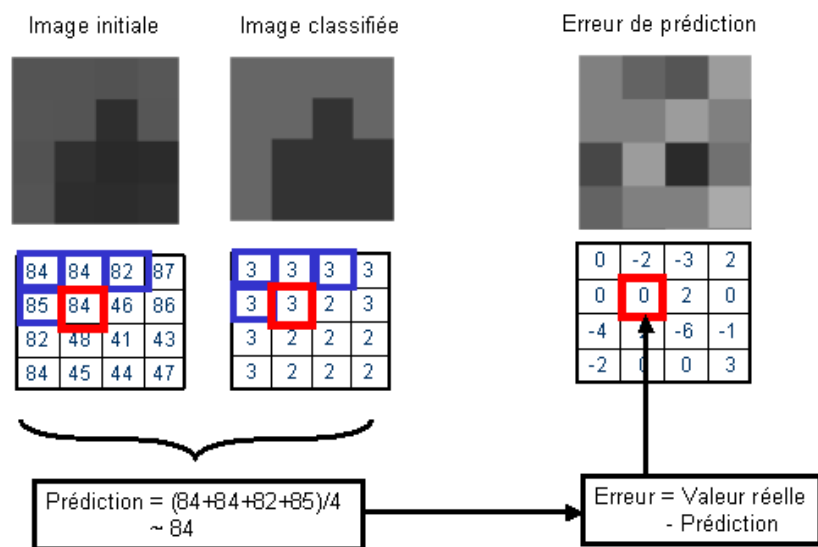


FIG. 2.8 – Au moins un des pixels déjà transmis appartient à la même classe.

- à la valeur moyenne de la classe de X (figure 2.9) *si aucun pixel voisin déjà transmis appartient à la même classe*.

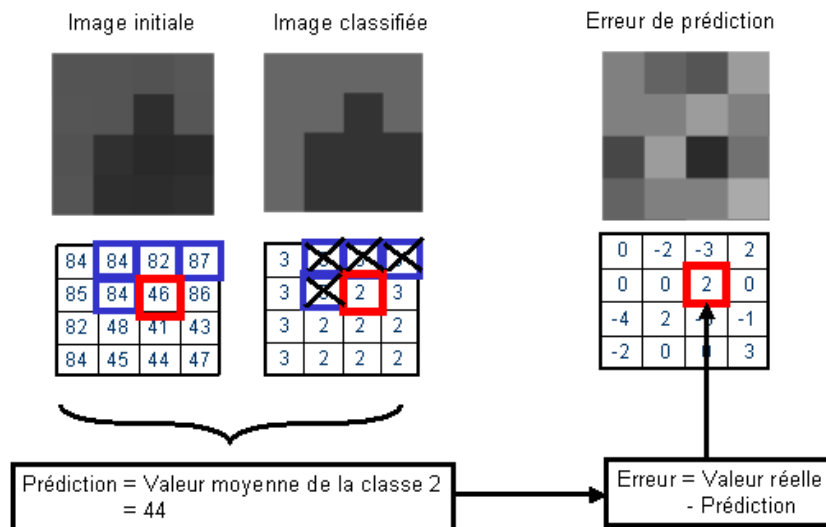


FIG. 2.9 – Aucun pixel déjà transmis n'appartient à la même classe.

Ce codage permet de définir la matrice d'erreurs de prédiction dont les valeurs sont proches de zéros (figure 2.10).

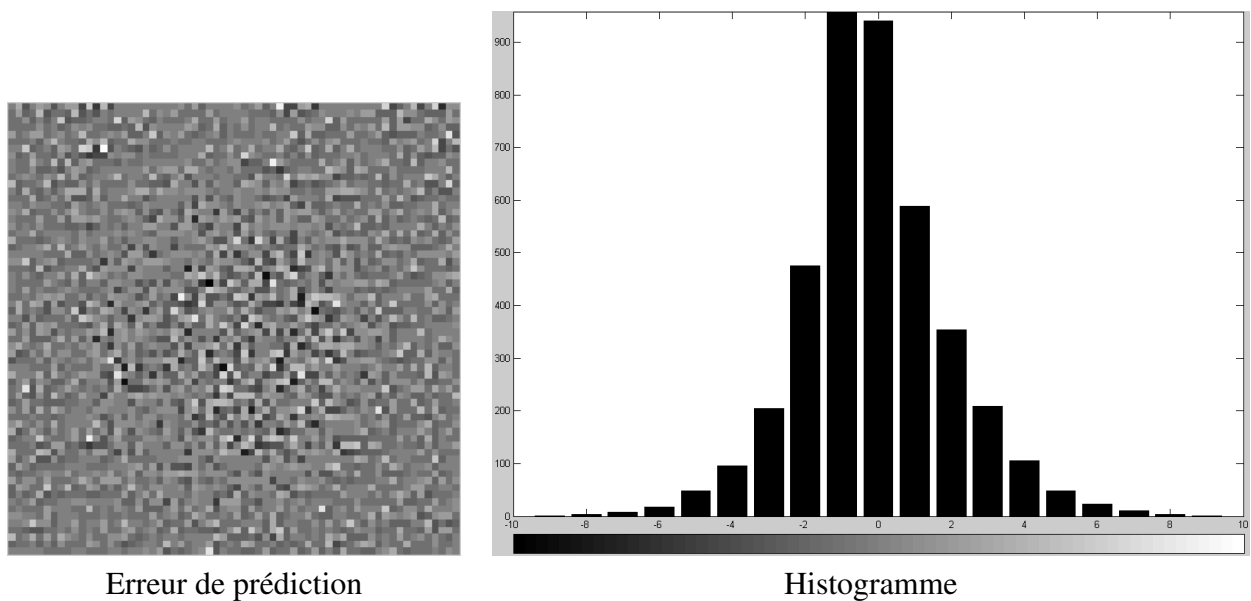


FIG. 2.10 – Erreur de prédiction et histogramme de l'image géométrique.

2.1.2 Résultats

Dans cette section nous nous intéressons aux résultats obtenus en prenant tout d'abord, le cas de l'image géométrique étudiée précédemment et ensuite, le cas d'une image médicale.

2.1.2.1 Cas d'une image géométrique

Influence du nombre de classes sur les résultats

La figure 2.11 illustre l'évolution du taux de compression en fonction du nombre de classes. Le taux de compression correspond au rapport entre le nombre total de bits nécessaire avant la compression, et le nombre total de bits nécessaire après la compression. Ainsi, plus ce rapport est important, plus le système de compression est performant en terme de réduction de l'information.

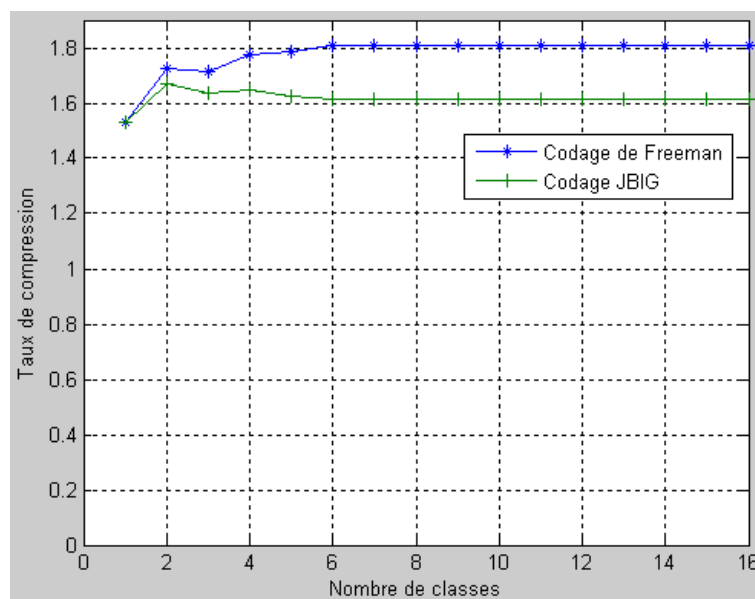


FIG. 2.11 – Taux de compression en fonction du nombre de classes (codage de Freeman et codage JBIG).

En utilisant le système de codage d'image classifiée développé, nous remarquons que le taux de compression subit tout d'abord une croissance et qu'à partir de 6 classes, il reste constant. Cela montre que le nombre optimal de classes pour l'image géométrique est à 6, ce qui correspond au nombre réel de classes utilisées lors de la création de cette image synthétique.

En utilisant le codage JBIG, nous obtenons des taux de compression inférieurs. Le taux de compression maximal obtenu est atteint pour un nombre de classe égal à deux, ce qui ne correspond pas au nombre de classe réel.

La figure 2.12 montre l'évolution de la moyenne et de l'écart type de la matrice des erreurs de prédiction en fonction du nombre de classes.

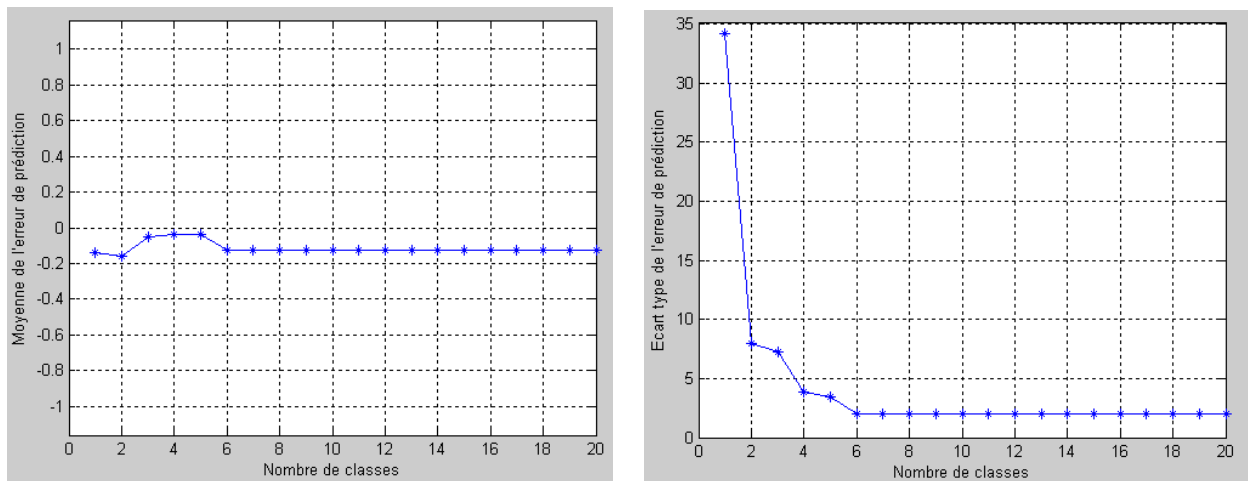


FIG. 2.12 – Moyenne et écart type de l'erreur de prédiction en fonction du nombre de classes.

Nous voyons que la moyenne est assez proche de zéro et varie peu. Par contre, l'écart type diminue nettement en fonction du nombre de classes. Cela explique également pourquoi nous obtenons un meilleur taux de compression à partir de 6 classes. En effet, le codage d'Huffman est d'autant plus performant que l'erreur de prédiction possède un écart à la moyenne très faible.

Comparaison avec les méthodes LJPEG et JPEG-LS

Afin d'évaluer notre processus de compression, nous avons comparé les résultats de notre système à ceux obtenus après codage par les deux standards réversibles de compression LJPEG et JPEG-LS.

Pour ces deux formats, le schéma général de compression est présenté figure 2.13. La différence entre ces méthodes se situe au niveau du prédicteur utilisé : un prédicteur fixe pour LJPEG et un prédicteur adaptatif pour JPEG-LS.

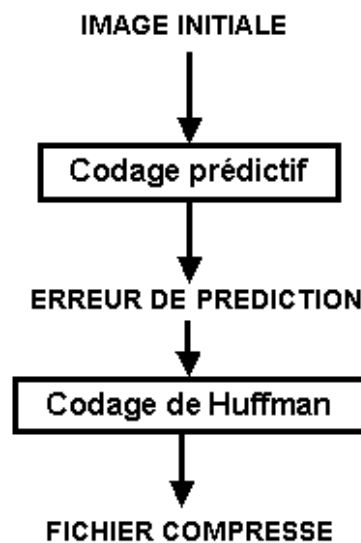


FIG. 2.13 – Schéma général des systèmes de compression LJPEG et JPEG-LS.

Les figures 2.14 et 2.15 représentent l'image et l'histogramme des erreurs de prédiction, obtenus respectivement avec la méthode LJPEG et JPEG-LS.

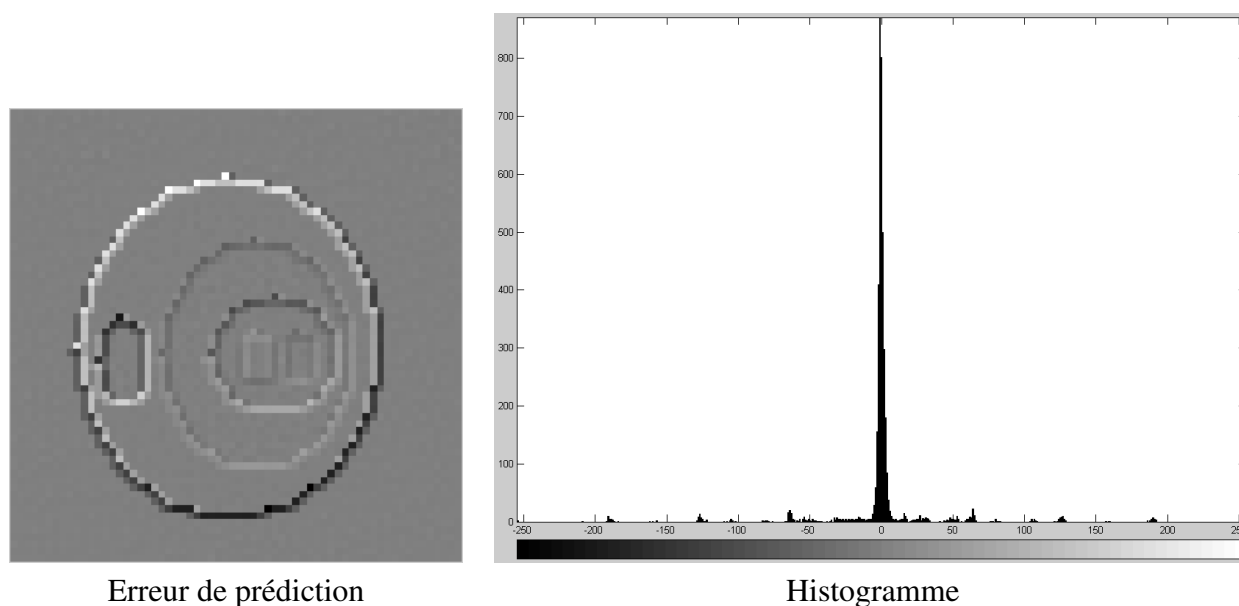


FIG. 2.14 – Erreur de prédiction et histogramme de l'image géométrique avec la méthode LJPEG (prédicteur 8). Taux de compression obtenu : 1.56 :1. Les autres prédicteurs donnent des taux de compression inférieurs.

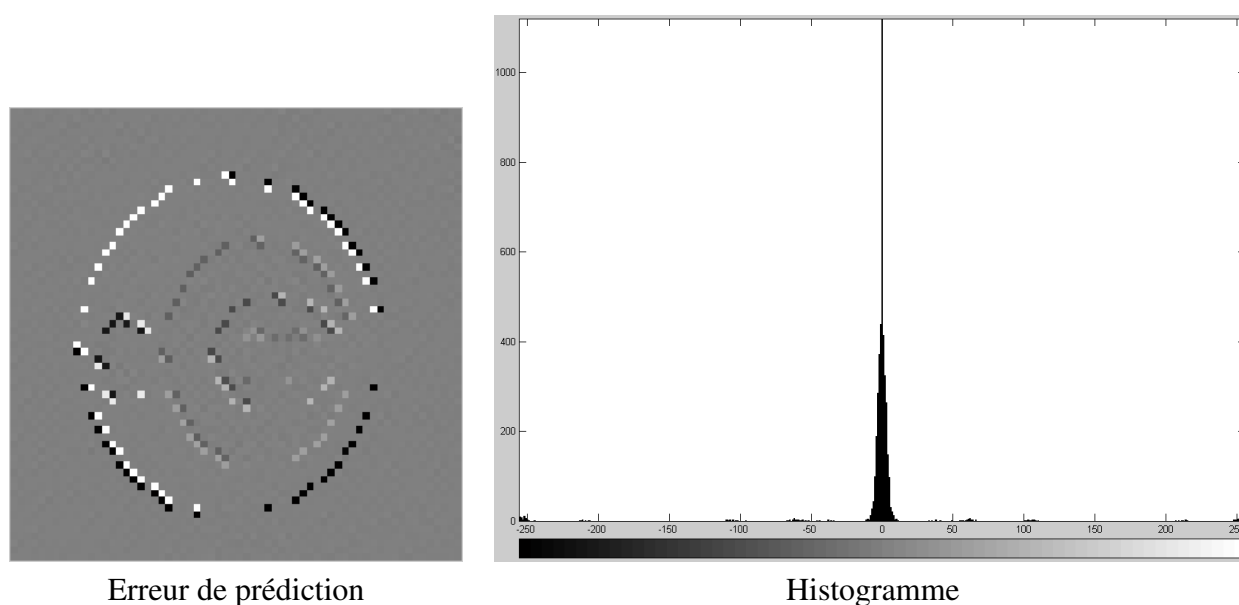


FIG. 2.15 – Erreur de prédiction et histogramme de l'image géométrique avec la méthode JPEG-LS. Taux de compression obtenu : 1.64 :1.

Après comparaison des images erreurs de prédiction obtenues par LJPEG, JPEG-LS et la méthode mise en œuvre (figure 2.10), nous remarquons que LJPEG et JPEG-LS donnent des erreurs de prédiction pouvant atteindre une grande amplitude (250 en valeur absolue).

La forte dynamique des histogrammes des erreurs de prédiction pour les méthodes LJPEG et JPEG-LS explique pourquoi nous obtenons de meilleurs résultats avec la méthode que nous avons développée.

Le tableau 2.1 est un tableau récapitulatif des résultats nous donnant les taux de compression, les moyennes et écarts types obtenus en utilisant notre méthode contour-texture, la méthode LJPEG avec le prédicteur 8 et la méthode JPEG-LS.

	Taux de compression	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
Méthode contour-texture développée avec le codage de Freeman :	1.8084	-0.1223	2.0486
Méthode LJPEG :	1.5607	-0.0984	34.1197
Méthode JPEG-LS :	1.6381	-0.1438	39.4521

TAB. 2.1 – Taux de compression, moyenne et écart type obtenus pour l'image géométrique

Interprétation

Nous pouvons en conclure que pour l'image test, le système de compression développé donne de meilleurs taux de compression que les méthodes LJPEG et JPEG-LS.

Néanmoins, les taux de compression semblent assez faibles. Cela est sans doute dû à la taille de l'image géométrique qui est très faible (64×64 pixels). Du coup, la table des symboles nécessaire pour le codage d'Huffman prend un pourcentage de place plus important dans le train binaire.

Le système de compression peut également être qualifié de non supervisé. En effet, si la valeur d'initialisation du nombre de classes est importante, ce nombre convergera ensuite vers la solution donnant le meilleur taux de compression.

2.1.2.2 Cas d'une image médicale

Image testée

L'image médicale étudiée et son histogramme sont représentés sur la figure 2.16.

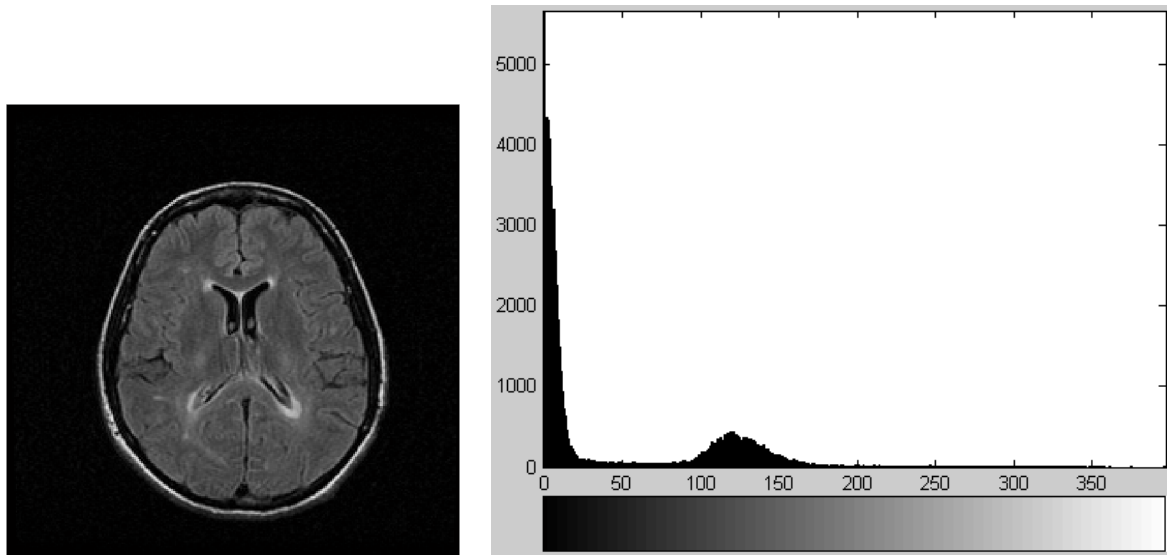


FIG. 2.16 – Image testée et son histogramme.

Cette image est une IRM cérébrale 256x256 codée sur 16 bits signés. Cette image est fortement bruitée et son histogramme nous montre qu'il sera difficile de classifier de façon précise les données intensités. En effet, contrairement à l'image géométrique étudiée précédemment, nous n'observons que deux maxima sur l'histogramme.

Influence du nombre de classes sur les résultats

Les figures 2.17 et 2.18 nous montrent respectivement l'évolution du taux de compression et du nombre total de régions en fonction du nombre de classes.

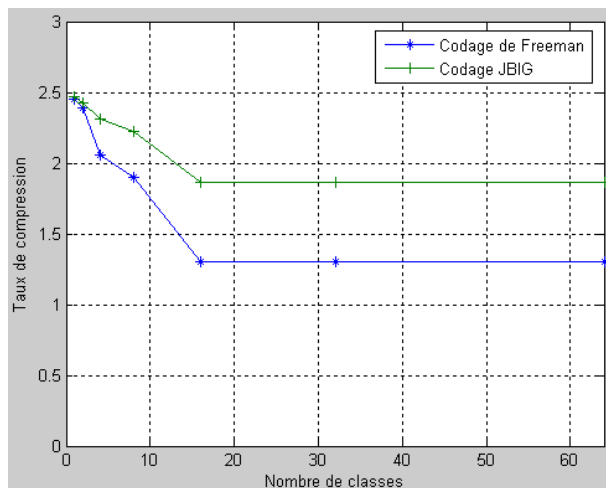


FIG. 2.17 – Taux de compression en fonction du nombre de classes.

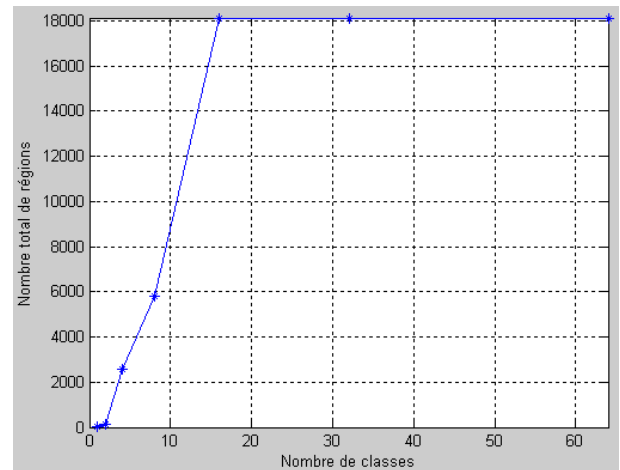


FIG. 2.18 – Nombre total de régions en fonction du nombre de classes.

Comme pour l'image géométrique, le taux de compression devient constant à partir d'un certain nombre de classes. Par contre, que ce soit en utilisant le codage de Freeman ou le codage JBIG, le taux de compression subit cette fois-ci, une diminution avant d'aboutir à une stagnation.

Avec l'utilisation du codage de Freeman, cette diminution s'explique par le fait que le nombre total de régions devient extrêmement important lorsque le nombre de classes augmente (plus de 18 000 régions à partir de 16 classes). Du coup, la trame contenant la description des contours de chaque région prendra de plus en plus de place dans le train binaire et le taux de compression s'en trouvera réduit.

Avec l'utilisation du codage JBIG, cette diminution du taux de compression peut s'expliquer par le fait que chaque classe après binarisation possède une répartition trop aléatoire comme le montre la figure 2.19.

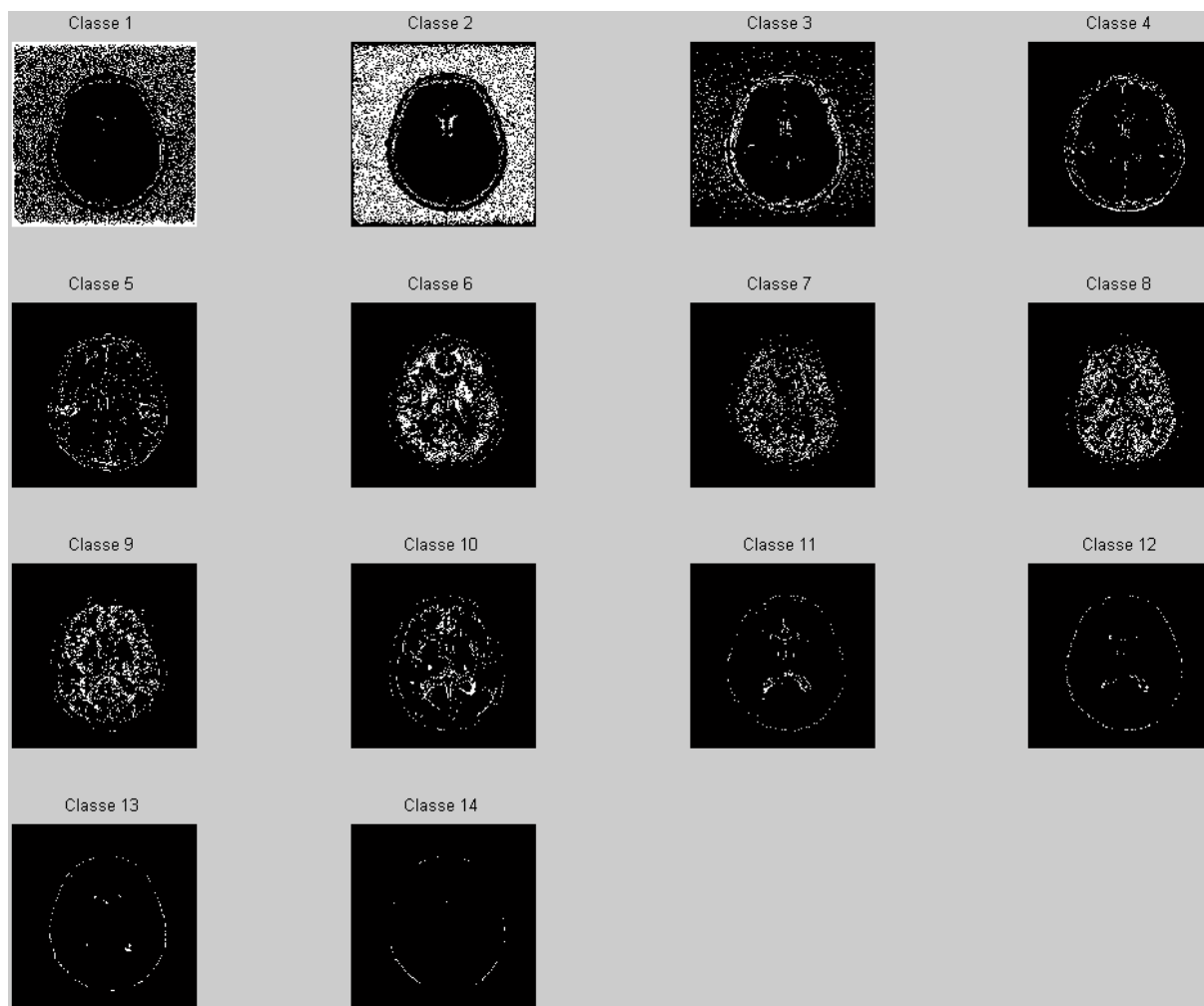


FIG. 2.19 – Les classes de l'image médicale après classification (initialisation du nombre de classes à 16).

La figure 2.20 représente l'évolution de la moyenne et de l'écart type de l'erreur de prédiction en fonction du nombre de classes.

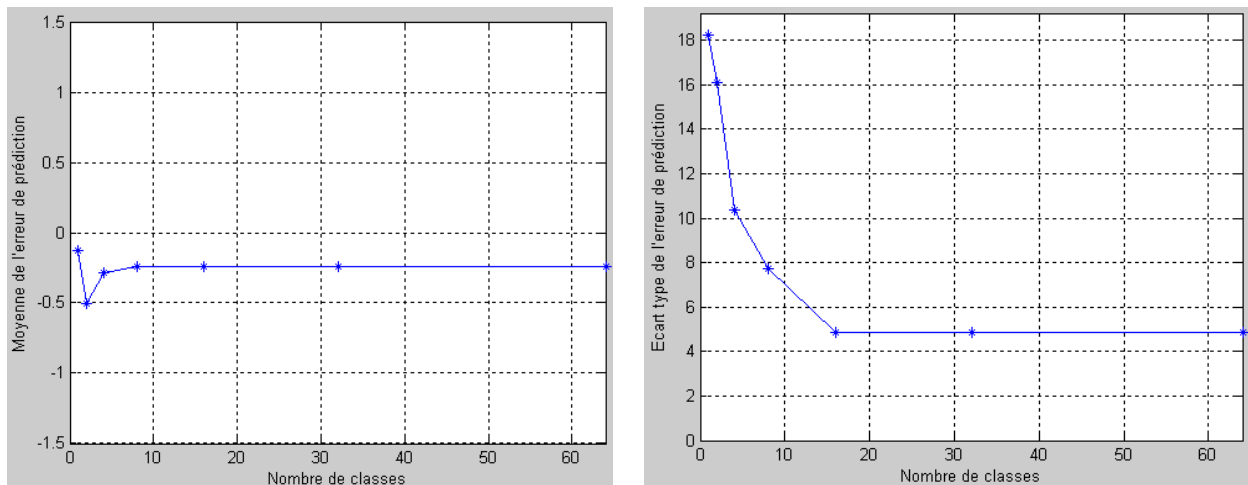


FIG. 2.20 – Moyenne et écart type de l'erreur de prédiction en fonction du nombre de classes.

Ces figures nous confirment bien que la diminution du taux compression est uniquement due au nombre trop important de régions à coder puisque la valeur moyenne de l'erreur de prédiction reste proche de zéro et que l'écart type de l'erreur de prédiction diminue lorsque le nombre de classes augmente. Le codage de la texture par Huffman est donc amélioré. La figure 2.21 nous montre que pour une initialisation du nombre de classes à 16, le fichier compressé contient plus d'information contour avec le codage de Freeman.

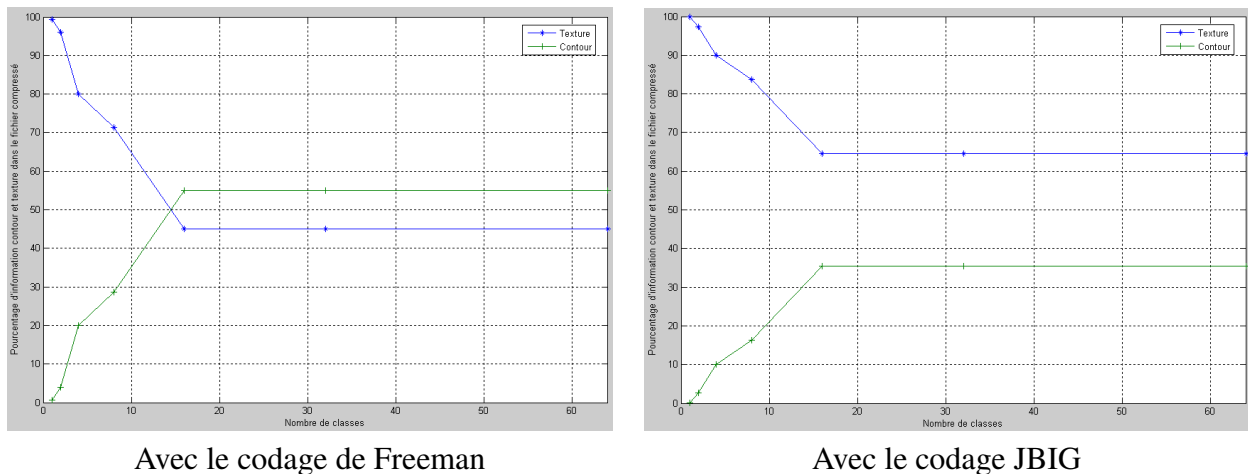


FIG. 2.21 – Pourcentage d'octets de la trame de données réservés au codage contour et texture en fonction du nombre de classes avec le codage de Freeman et le codage JBIG.

Le tableau 2.2 est un tableau récapitulatif de l'évolution des différents paramètres en fonction du nombre de classes.

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	2.4526	2.4660	1	-0.12502	18.224
2	2.3935	2.4235	153	-0.50156	16.121
4	2.0537	2.3086	2598	-0.28214	10.374
8	1.8955	2.2268	5818	-0.23717	7.7278
≥ 16	1.3039	1.8683	18132	-0.241	4.8668

TAB. 2.2 – Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale avec différentes initialisations du nombre de classes

Comparaison avec les méthodes LJPEG et JPEG-LS

Les figures 2.22, 2.23 et 2.24 représentent les erreurs de prédiction obtenues respectivement avec la méthode que nous avons développée, la méthode LJPEG et la méthode JPEG-LS.

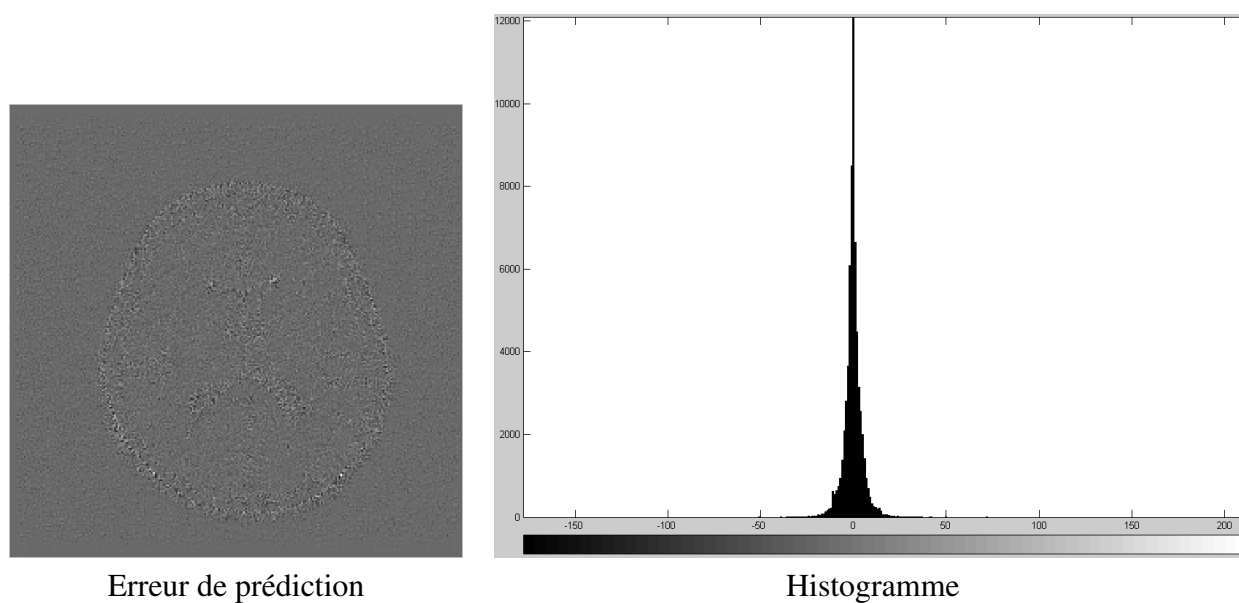


FIG. 2.22 – Erreur de prédiction et histogramme de l'image médicale avec la méthode propre. Taux de compression obtenu : 1.87 :1.

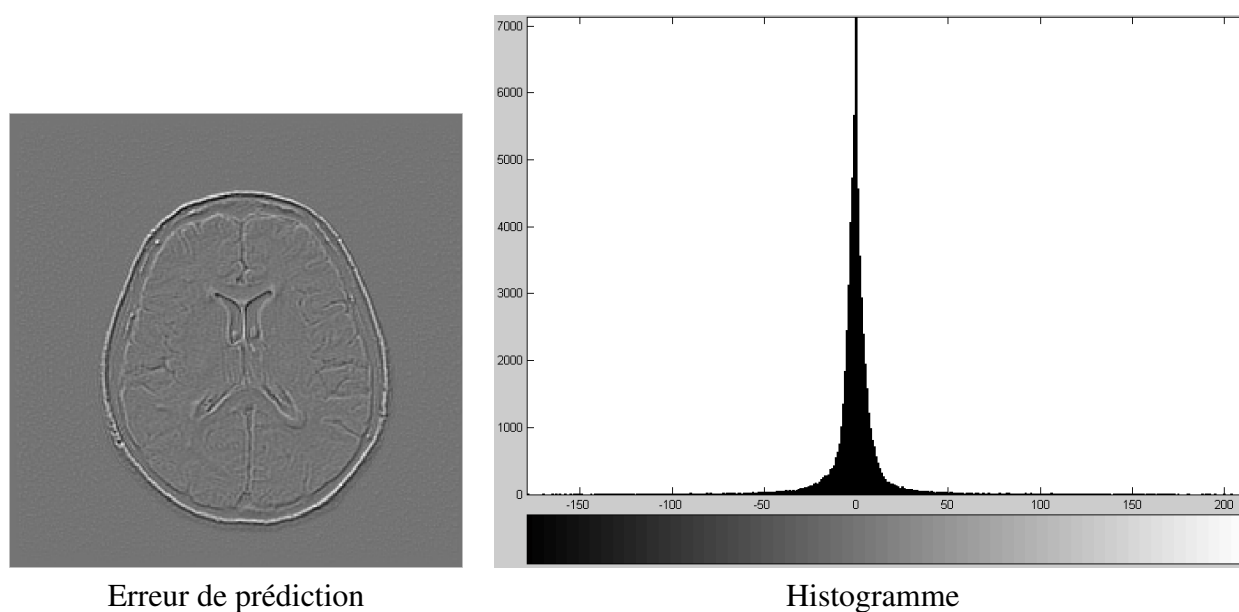


FIG. 2.23 – Erreur de prédiction et histogramme de l'image médicale avec la méthode JPEG (prédicteur 8). Taux de compression obtenu : 2.47 :1.

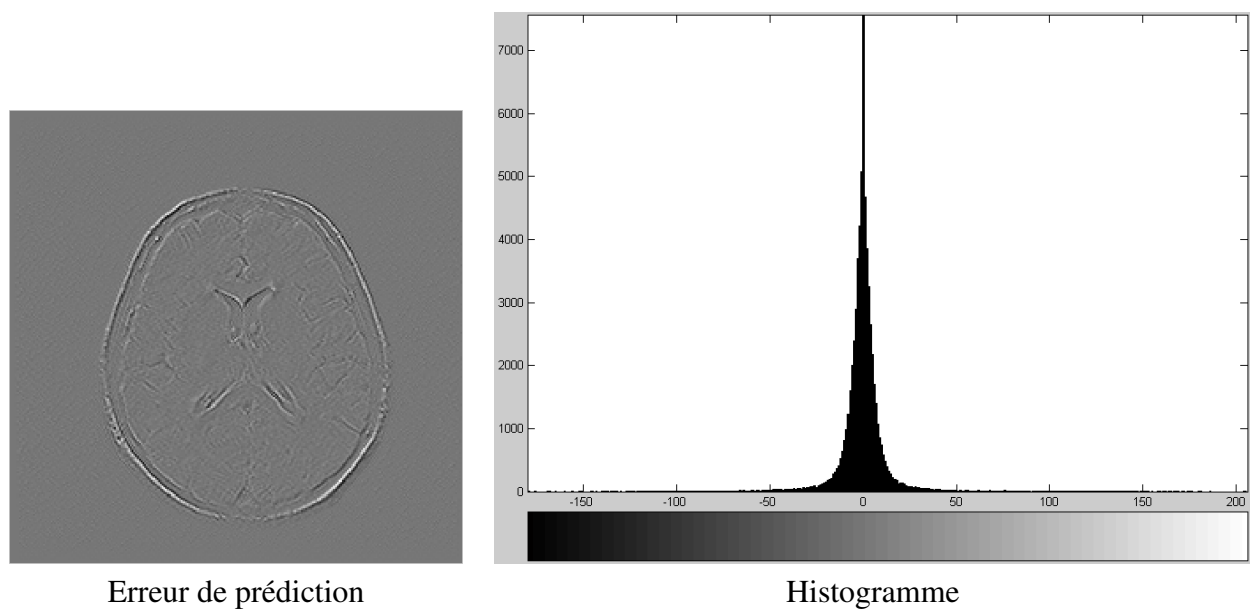


FIG. 2.24 – Erreur de prédiction et histogramme de l'image médicale avec la méthode JPEG-LS. Taux de compression obtenu : 2.50 :1.

Comme pour l'image test, l'écart type sur l'erreur de prédiction obtenu avec notre système est plus faible que ceux obtenus par LJPEG et JPEG-LS. En plus, nous pouvons remarquer que dans l'histogramme de l'erreur de prédiction obtenue avec la méthode prédictive par régions, 1 pixel sur 5 environ a une valeur nulle (contre 1 pixel sur 10 pour les autres méthodes).

Le tableau 2.3 est un tableau récapitulatif des résultats nous donnant les taux de compression, les nombres d'octets nécessaires au codage de la texture d'une part et du contour d'autre part, les moyennes et écarts types obtenus en utilisant la méthode propre, la méthode LJPEG avec le prédicteur 8 et la méthode JPEG-LS.

	Taux de compression	Nombre d'octets du train binaire nécessaires pour coder l'information		Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
		texture	contour (JBIG)		
Méthode propre :	1.8683	45 336	24 819	-0.2410	4.8668
Méthode LJPEG :	2.4705	53 055	0	-0.1226	18.2225
Méthode JPEG-LS :	2.4988	52 454	0	-0.2706	15.6793

TAB. 2.3 – Taux de compression, nombre d'octets du train binaire nécessaire pour coder l'information contour/texture, moyenne et écart type obtenus pour l'image médicale

Interprétation

Le taux de compression obtenu avec la méthode propre (et un nombre de classe à 16) se trouve inférieur à ceux obtenus avec LJPEG et JPEG-LS. Cela est dû au fait que le nombre de régions à coder est très important. En effet, le tableau 2.3 nous montre que le nombre d'octets nécessaires pour coder l'information contour est extrêmement important, alors que le nombre d'octets nécessaires pour coder l'information texture est plus faible que les méthodes LJPEG et JPEG-LS.

Ce grand nombre d'information contour à coder s'explique par le fait que la classification est uniquement fonction des niveaux de gris de l'image et qu'elle ne tient pas compte de l'information spatiale.

Notre système bien que s'adaptant à la texture des structures de l'image n'est pas approprié sans prétraitement adéquat aux images bruitées telles que les images médicales.

Afin d'améliorer notre système, nous avons utilisé les décompositions développées par Aujol et Gilles pour représenter la structure géométrique de l'image et ainsi obtenir un nombre réduit de régions à coder.

Le principe et les résultats obtenus avec ce type de prétraitement fait l'objet de la section suivante.

2.2 Utilisation de la décomposition d'image pour le prétraitement

2.2.1 Principe

Le schéma synthétique de la figure 2.25 représente le schéma général de la phase de compression.

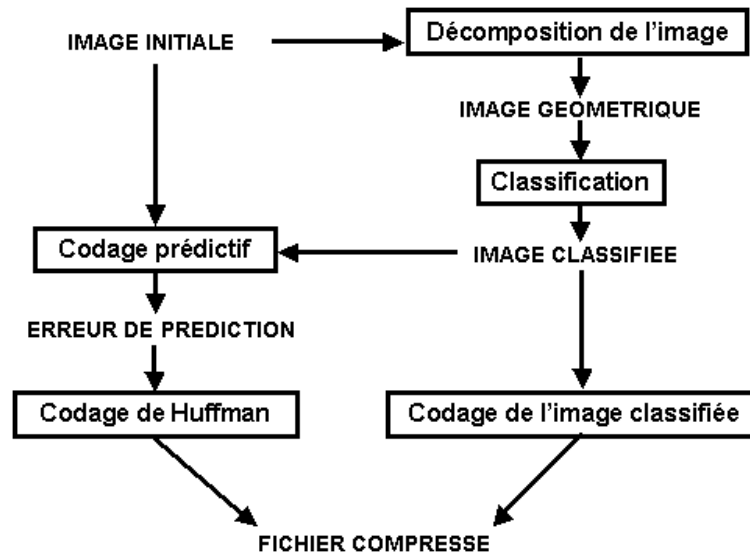


FIG. 2.25 – Schéma général du système de compression en utilisant la décomposition.

Par rapport au schéma représenté sur la figure 2.2, nous avons rajouté une étape de prétraitement consistant à décomposer l'image initiale suivant sa structure géométrique, sa texture et le bruit qu'elle contient.

L'image géométrique subit ensuite une classification par histogramme pour obtenir une image classifiée.

2.2.2 Résultats

Les tableaux 2.4, 2.5 et 2.10 représentent les résultats obtenus respectivement avec l'algorithme de décomposition d'Aujol, avec l'algorithme de décomposition de Gilles, et l'unification des deux algorithmes.

Les choix des différents paramètres ont été décrits dans la section 1.3.

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	2.4525	2.4659	1	-0.1252	18.2242
2	2.4265	2.4329	56	-0.47678	16.3717
4	2.2854	2.3836	896	-0.2971	13.0884
8	2.1692	2.3127	1843	-0.12451	11.1288
16	1.9844	2.1741	3366	-0.03241	10.3972

TAB. 2.4 – Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec prétraitement d'Aujol) avec différentes initialisations du nombre de classes

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	2.4526	2.466	1	-0.12508	18.2239
2	2.4179	2.4302	72	-0.50522	16.1613
4	2.1445	2.3493	1451	-0.20695	11.085
8	1.9709	2.251	3546	-0.17523	8.5649
16	1.8356	2.1133	6448	-0.08876	6.9928

TAB. 2.5 – Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec prétraitement de Gilles) avec différentes initialisations du nombre de classes

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	2.4526	2.466	1	-0.12502	18.2237
2	2.4198	2.4295	80	-0.51657	16.1243
4	2.0953	2.3387	1671	-0.22852	10.8491
8	1.9769	2.2526	3732	-0.16135	8.1068
16	1.8711	2.1519	5742	-0.12781	6.9921

TAB. 2.6 – *Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec le deuxième algorithme de prétraitement de Gilles) avec différentes initialisations du nombre de classes*

Ces tableaux nous montrent que même en utilisant la composante géométrique pour réaliser la classification, le taux de compression diminue lorsque le nombre de classes augmente mais reste dans tous les cas supérieur à celui obtenu sans prétraitement. Pour chaque classe de l'image classifiée, nous obtenons en fait donc plus de composantes connexes comme le montre la figure 2.26.

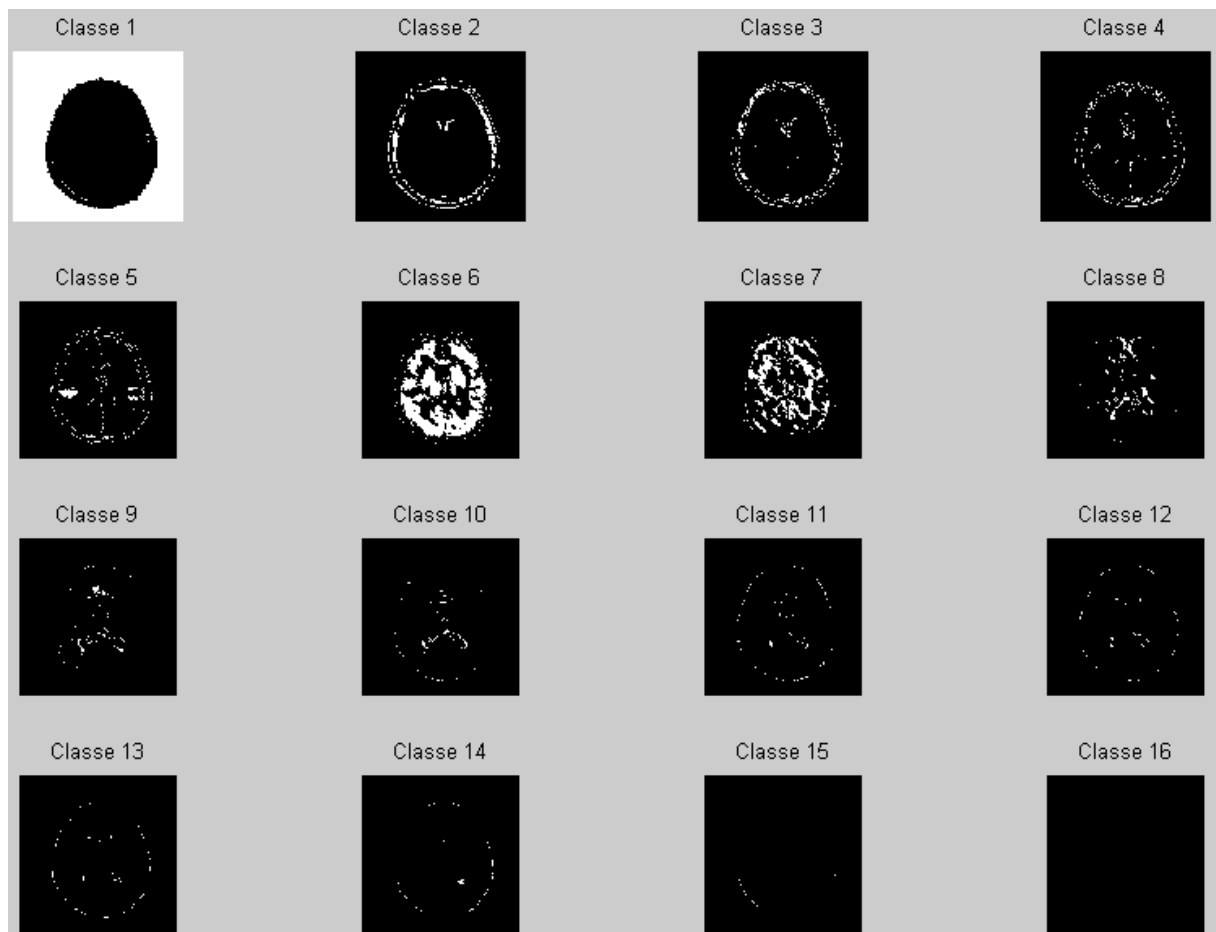


FIG. 2.26 – Les classes de l'image médicale après classification (avec prétraitement d'Aujol et initialisation du nombre de classes à 16).

2.3 Méthode de compression quasi-exact

2.3.1 Principe

Dans les IRM cérébrales, le fond noir de l'image correspond uniquement au bruit et son non-codage peut permettre d'augmenter le taux de compression sans pour autant changer l'interprétation diagnostique.

Nous commençons donc tout d'abord par déterminer un masque à partir de l'image initiale. Ce masque est obtenu en utilisant la classification par histogramme et en choisissant un nombre de classe égal à 2. En effet, la peau délimite l'information utile au diagnostic et possède la particularité d'avoir une intensité élevée dans l'image. Le masque est obtenu en réalisant une fermeture de l'image après classification (figure 2.27). Les pixels du masque seront donc égaux à zéro si ils correspondent au fond noir de l'image et à un dans le cas contraire.

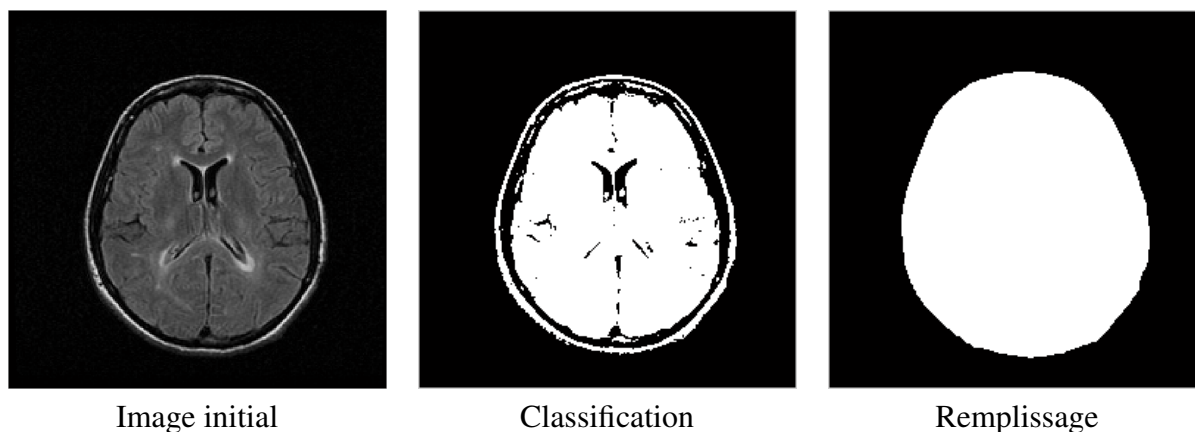


FIG. 2.27 – Construction du masque.

Grâce à ce masque, nous pouvons effectuer une classification sur la partie utile de l'image et ne pas transmettre le fond noir de l'image originale.

2.3.2 Résultats obtenus sans prétraitement

Les résultats obtenus sont représentés dans le tableau 2.7.

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	4.8302	4.8231	1	-0.12315	18.2812
2	4.6542	4.64	131	-0.43169	15.6373
4	3.4645	4.2093	2346	-0.19159	10.2241
8	2.8717	3.8273	6356	-0.20239	7.1616
16	2.5564	3.5702	10040	-0.12987	4.9921
≥ 32	2.402	3.4433	11991	-0.12779	4.5665

TAB. 2.7 – *Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée avec différentes initialisations du nombre de classes*

Nous voyons que nous obtenons de meilleurs taux de compression. En effet, sans l'utilisation d'un masque, la classification est biaisée par le fond noir de l'image contenant que du bruit. De plus, seuls les pixels contenu dans le masque sont transmis (ce qui correspond à environ 40% des pixels de l'image initiale), comme le montre la figure 2.28 avec une initialisation de 16 classes.

Cependant, nous constatons que le taux de compression diminue en fonction du nombre de classes avec les deux méthodes de codage de l'image classifiée.

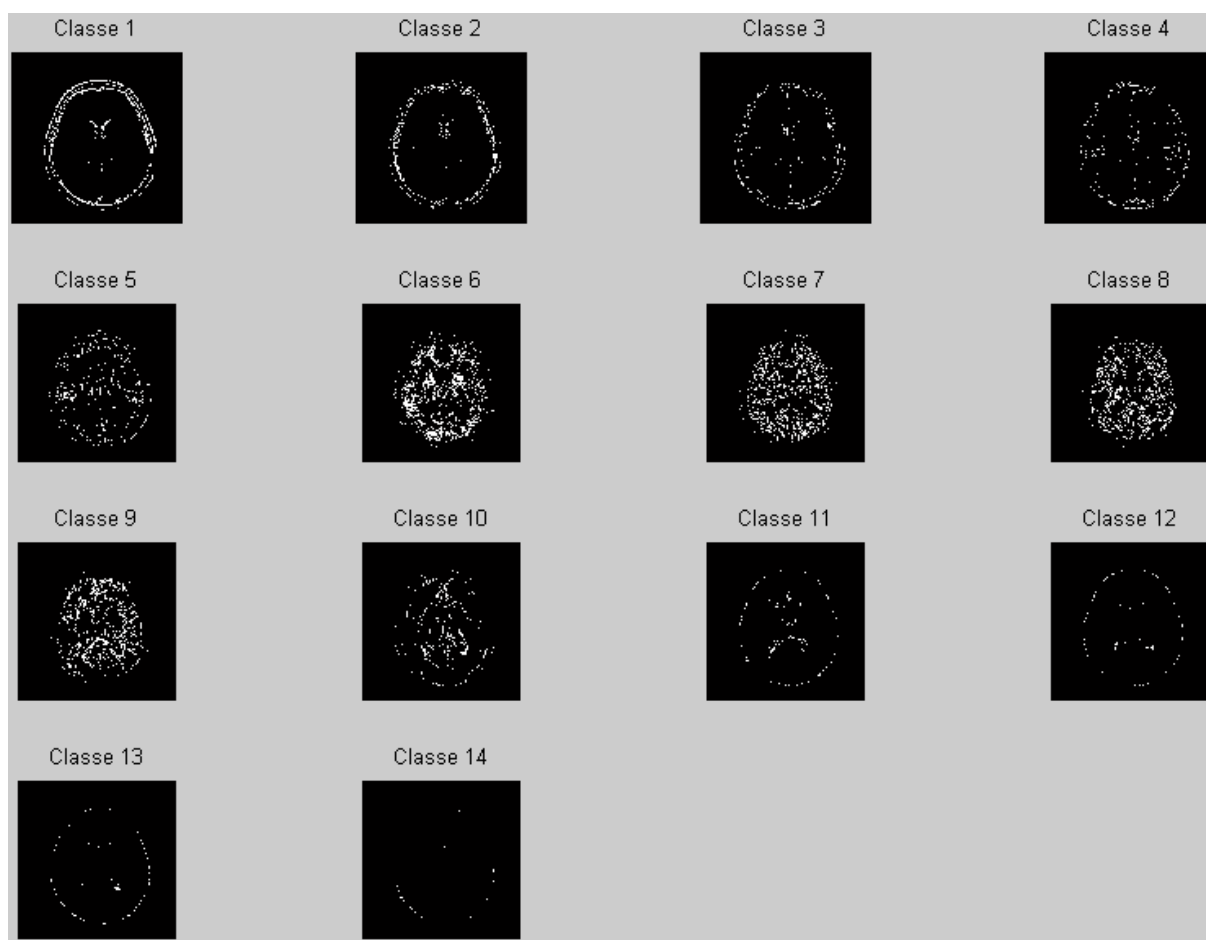


FIG. 2.28 – Les classes de l'image médicale après classification (initialisation du nombre de classes à 16) et utilisation d'un masque.

2.3.3 Résultats obtenus avec prétraitement

Les tableaux 2.8, 2.9 et 2.10 nous donnent l'influence du nombre de classes sur les différents paramètres en utilisant les trois types de décomposition.

Le masque permet en outre d'avoir une estimation du bruit contenu dans l'image. En effet, les valeurs nulles du masque correspondent au fond de l'image qui ne contient que du bruit. Ainsi, l'écart type du bruit σ peut être choisi comme étant égal à l'écart type du fond de l'image.

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	4.8311	4.824	1	-0.12535	18.2946
2	4.6976	4.6848	58	-0.35468	15.6049
4	4.3934	4.5358	781	-0.12094	12.3896
8	3.5183	4.1919	2144	-0.083862	10.748
16	3.2393	3.795	3575	-0.034805	10.3228

TAB. 2.8 – Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée (avec prétraitement d'Aujol) avec différentes initialisations du nombre de classes

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	4.8316	4.8245	1	-0.12517	18.2935
2	4.6939	4.659	93	-0.39131	15.4861
4	3.6462	4.3588	1487	-0.17131	11.0292
8	3.2402	4.0376	3583	-0.16275	8.6236
16	2.8542	3.5884	6532	-0.10031	6.9949

TAB. 2.9 – Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale masquée(avec prétraitement de Gilles) avec différentes initialisations du nombre de classes

Initialisation du nombre de classes	Taux de compression (codage de Freeman)	Taux de compression (codage JBIG)	Nombre total de régions	Moyenne de l'erreur de prédiction	Ecart type de l'erreur de prédiction
1	4.8295	4.8224	1	-0.12425	18.2878
2	4.7023	4.6708	79	-0.44324	15.7434
4	3.5981	4.3247	1668	-0.16972	10.7929
8	3.1433	3.9568	4062	-0.20215	8.3815
16	2.9262	3.7003	5917	-0.14095	6.9183

TAB. 2.10 – *Taux de compression, nombre total de régions, moyenne et écart type obtenus pour l'image médicale (avec le deuxième prétraitement de Gilles) avec différentes initialisations du nombre de classes*

Encore une fois, le taux de compression diminue quand le nombre de classes augmente mais reste supérieur à celui obtenu sans prétraitement. Pour un nombre de classes initialisé à 16, nous obtenons avec un prétraitement et un masque deux fois moins de régions à coder, ce qui reste néanmoins trop important.

Discussion

Le système de compression développé donne des résultats corrects dans le cas où nous nous intéressons à des images géométriques. En effet, nous avons obtenu un système de compression non supervisé apportant des taux de compression légèrement supérieurs à ceux obtenus par les standards JPEG-LS et LJPEG.

Cependant, les résultats obtenus sur l'image médicale sont décevants en réversible. Même en utilisant un prétraitement, chaque classe de l'image classifiée possède une répartition peu homogène spatialement pour obtenir de bon taux de compression.

En effet, la méthode de classification utilisée est monodimensionnelle et pourrait être améliorée en utilisant une information spatiale telle que dans [16].

Le processus de classification peut aussi être amélioré en homogénéisant spatialement l'image obtenue après classification, ce qui permettrait de supprimer les régions non significatives (trop petites pour être codées) et d'avoir moins d'information contour à coder.

Théoriquement, le système serait plus performant en irréversible. La classification permettrait de coder les différentes structures selon leur importance dans le processus diagnostique. Le système de compression a été développé en réversible pour être évalué selon l'unique critère du taux de compression et pour être transformé par la suite en irréversible. L'évolution du système de compression en irréversible devra être évalué également en fonction de la qualité de l'image compressée.

Bibliographie

- [1] J.R. CASAS et L. TORRES. A region-based subband coding scheme. Signal Processing : Image Communication, 1997.
- [2] R. MOLINA A.K. KATSAQQELOS et G.M. CALLICO D. BARRETO, L.D. ALVAREZ. Region-based super-resolution for compression. Multidimensional Systems and Signal Processing, 2007.
- [3] H. ZHANG et F. BOSSEN. Region-based coding of motion fields for low-bitrate video compression. Proceedings - International Conference on Image Processing, ICIP, 2004.
- [4] M. NADAR D.P NAIDICH C.L. NOVAK et M.W. MARCELIN B.L. Odry, K. Krishnan. The effect of 3d region-based compression on the performance of an automatic lung nodule detection system. Progress in Biomedical Optics and Imaging - Proceedings of SPIE, 2005.
- [5] CAVARO-MENARD Christine et NAIT-ALI Amine. Compression des images et des signaux médicaux. Livre, 2007.
- [6] Jean-Charles ATINE. Méthode d'apprentissage floue : application à la segmentation d'images biologiques. Thèse de Doctorat, 2005.
- [7] R. Krishnapuram. A possibilistic approach to clustering. IEEE Trans. Fuzzy Systems, 1993.
- [8] M. Ménard and M. Eboueya. Extreme physical information and objective functions in fuzzy clustering. Fuzzy Sets and Systems, Volume 128, Issue 3, 16 June 2002, Pages 285-303.
- [9] C. Tsallis. Entropic nonextensivity : a possible measure of complexity. Chaos, Solitons and Fractals, Volume 13, Issue 3, March 2002, Pages 371-391.
- [10] J.F. Aujol. Contribution à l'analyse de texture en traitement d'images par approches variationnelles et equations au dérivées partielles. Thèse de Doctorat, 2004.
- [11] A. Chambolle. An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, Volume 20, Issue 1-2, January 2004, Pages 89-97.
- [12] D.L. Donoho et M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. Journal of the American Statistical Association, 1995.
- [13] Y. Meyer. Oscillating patterns in image processing and in some non linear evolution equations. The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures, 2001.
- [14] T. Chan et S. Osher J.F. Aujol, G. Gilboa. Structure-texture image decomposition - modeling, algorithms, and parameter selection. International Journal of Computer Vision, Volume 67, Issue 1, April 2006, Pages 111-136.
- [15] J. Gilles. Décomposition et détection de structures géométriques en imagerie. Thèse de Doctorat, 2006.
- [16] COCQUEREZ Jean-Pierre et PHILIPP Sylvie. Analyse d'images : filtrage et segmentation. Livre, 1997.

- [17] E. Candès et D. Donoho. A surprisingly effective nonadaptive representation of objects with edges. *Curves and Surfaces*, 1999.
- [18] A. Cohen et B. Matei. Nonlinear subdivisions schemes : applications to image processing. *Tutorials on Multiresolution in Geometric Modelling*, 2002.
- [19] M. N. Do et M. Vetterli R. Shukla, P. L. Dragotti. Rate-distortion optimized tree structured compression algorithm for piecewise smooth images. *IEEE Transactions Image Processing*, 2004.
- [20] E. Lepennec et S. Mallat. *Bandelettes et représentation géométrique des images*. Grets, 2003.
- [21] M. N. Do et M. Vetterli. The finite ridgelet transform for image representation. *IEEE transaction on image processing*.
- [22] HACHAMA Mohamed et BOHOUA-NASSE Franck-Olivier. Une segmentation grossière et rapide des images en couleurs. DEA, 2003-2004.
- [23] Raghu Krishnapuram et Nikhil R. Pal James C. Bezdek, James Keller. *Fuzzy models and algorithms for pattern recognition and image processing*. Kluwer Academic Publishers, TA 1650.F89 1999.
- [24] C. CAVARO-MENARD et P. JALLET. A coding system based on a contour-texture model for medical image compression. *Innovation et technologie en biologie et médecine*, 1996.
- [25] L. KHODJA. Contribution à la classification floue non supervisée. Thèse.
- [26] R. ZITOUNI. Segmentation contextuelle des images irm cérébrales par le fcm semi supervisé, première étape a l'aide au diagnostic. *Journées Francophones d'Informatique Médicale*, 2005.
- [27] G. PEYRE. Géométrie multi-échelles pour les images et les textures. Thèse, 2005.
- [28] F. HACHOUF et N. MEZHOUD. Segmentation neuro-floue, application aux images à résonance magnétique. CISTEMA, 2003.
- [29] A. RICK. Représentation de la variabilité dans le traitement d'images flou. Thèse, 1999.
- [30] M.N. MURTY et P.J. FLYNN A.K. JAIN. Data clustering. Review, 1999.
- [31] O. PEREZ-ORAMAS et V. BOMBARDIER. Le paradigme du traitement flou des images. Internet : [http ://www.angelfire.com/fl/fuzzymage/fripf.html](http://www.angelfire.com/fl/fuzzymage/fripf.html).

Annexe A

Codage et décodage des contours de l'image classifiée

A.1 Codage de l'image classifiée

Un codage des contours est réalisé après extraction des contours de l'image classifiée. Le schéma général d'extraction des contours est donné figure A.1. La figure A.2 illustre la méthode employée avec l'exemple de l'image géométrique étudiée.

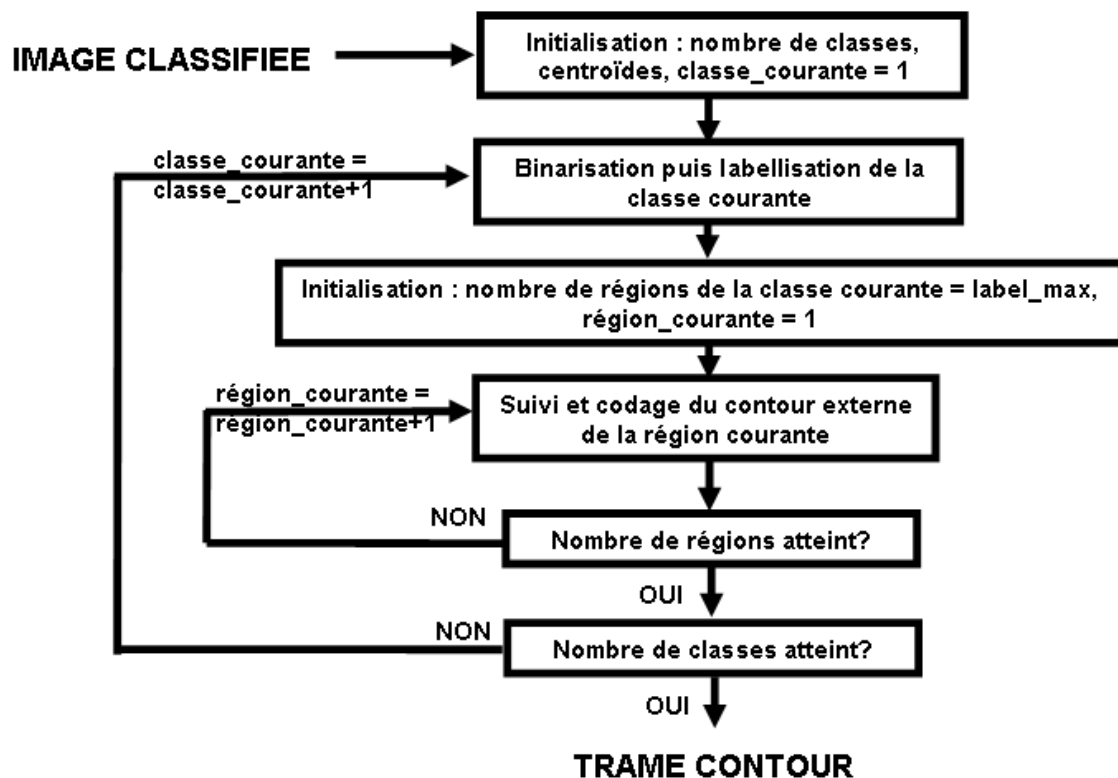


FIG. A.1 – Extraction et codage des contours.

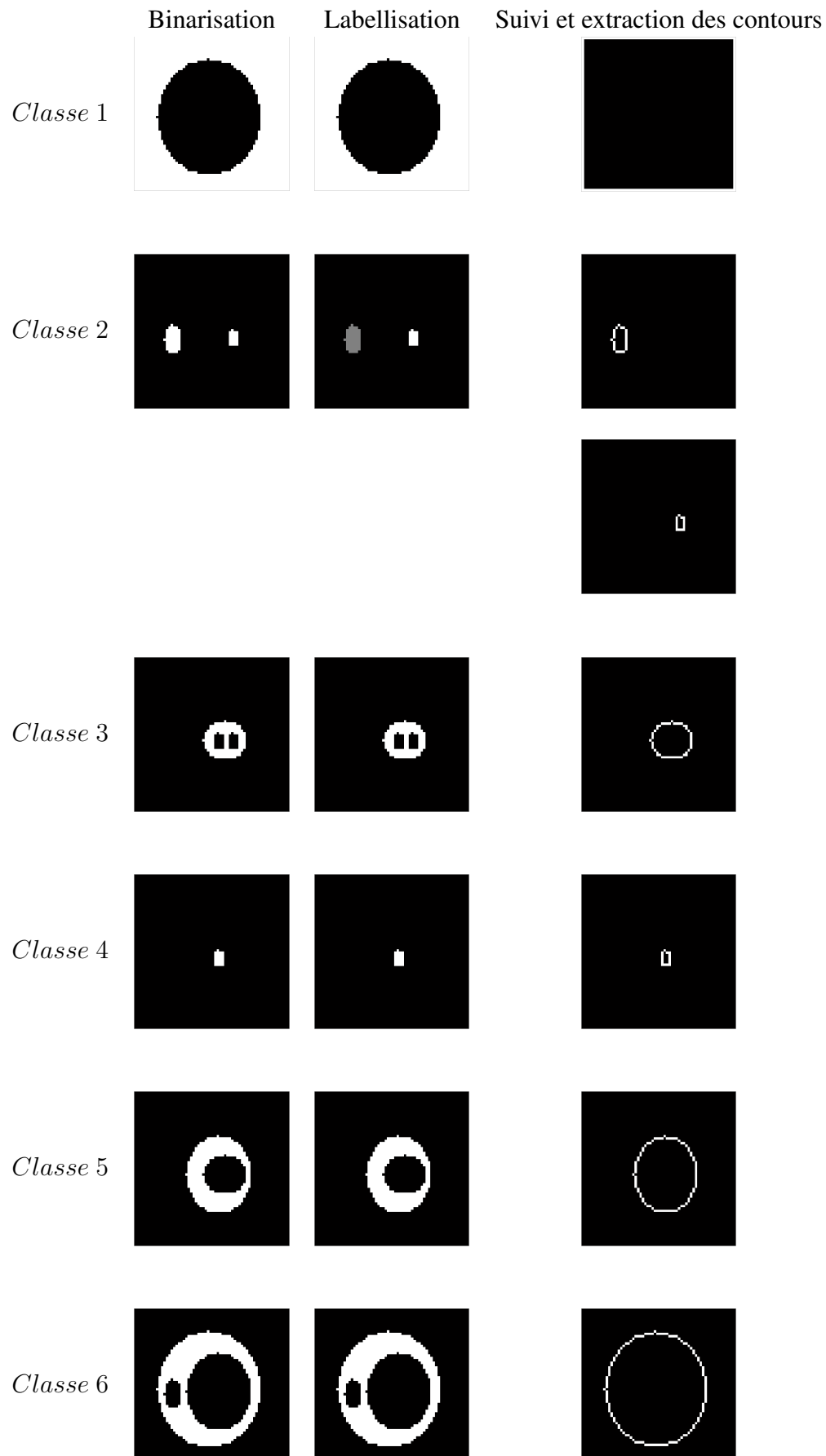


FIG. A.2 – Suivi et extraction des contours sur l'image géométrique.

Chaque classe est binarisée (figure A.2), c'est à dire que les pixels de la matrice sont mis à 1 si ils appartiennent à la classe étudiée et à 0 dans le cas contraire.

Cette matrice binaire est ensuite labelisée : la classe est alors divisée si nécessaire en plusieurs régions (composantes non connexes). Une valeur est attribuée à chaque région et nous pouvons en déduire le nombre de régions par classe. Chaque région est ensuite binarisée et son contour externe est suivi à partir du premier point en haut à gauche selon un protocole que nous avons défini. Les changements de direction du contour sont codés selon Freeman.

Nous obtenons finalement une trame comme le montre la figure A.3.

Nombre de classes	Centroïdes de chaque classe	Nombre de régions par classe	Contours externes de la région 1	Contours externes de la région 2	

FIG. A.3 – Trame contenant les frontières de chaque région.

Cette trame est un vecteur contenant :

- le nombre de classes, 6 dans notre exemple ;
- un vecteur de dimension le nombre de classes contenant les centroïdes de chaque classe : dans notre exemple ce vecteur contiendra les valeurs 0, 44, 84, 128, 192, 255 ;
- un vecteur de dimension le nombre de classes contenant le nombre de régions par classe : dans notre exemple ce vecteur contiendra les valeurs 1, 2, 1, 1, 1, 1 ;
- et ensuite la description des contours externes de chaque région après codage de Freeman (chapitre 1.1.4.2) ;

Protocole de suivi des points contours d'une région

Partant du premier point contour de la région le plus en haut à gauche, les points contours sont explorés dans le sens trigonométrique. Certaines régions contiennent des trous et la description des contours de ces trous n'est pas nécessaire pour que le système reste réversible (car elle est réalisée lors de l'étude de la région constituant le trou).

Sur la figure A.4, les flèches rouges représentent les différentes directions pour passer d'un point contour à un autre.

Pour repérer le premier point contour voisin du point initial de la région (premier point en haut à gauche), on étudie le voisinage du point situé au dessous du point initial (ligne du dessous, même colonne), en partant de la direction codée 3 selon Freeman (point situé en haut à gauche du point étudié). L'exploration des voisins se fait dans le sens trigonométrique, le premier point de la région (non déjà qualifié de point contour) est alors le point contour recherché.

Pour déterminer le prochain point contour à partir d'un repère, on balaye toutes les directions possibles (flèches noirs) jusqu'à trouver un nouveau point contour. Ce balayage s'effectue selon un certain sens dépendant du repère (sens trigonométrique comme nous le voyons avec la flèche blanche).

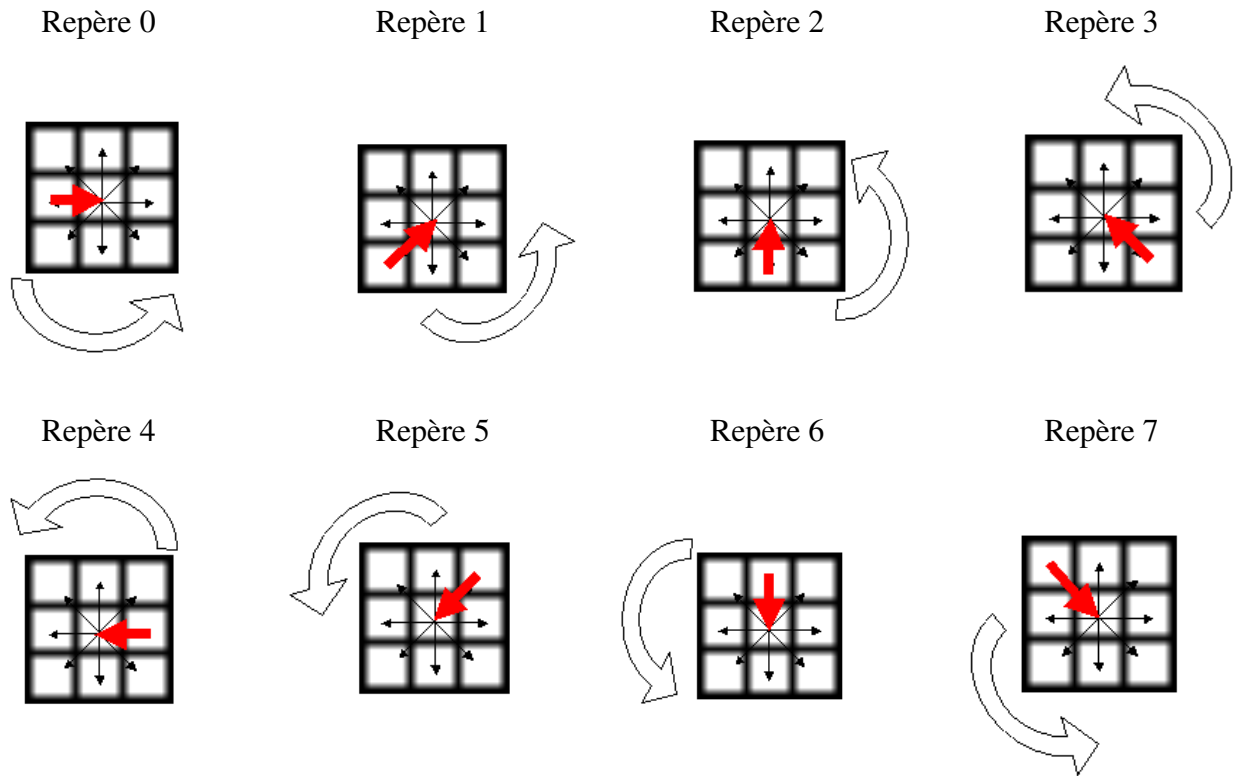


FIG. A.4 – Détermination des points contours.

A.2 Décodage de l'image classifiée

La construction de l'image classifiée est réalisée suivant 3 étapes.

A.2.1 Construction d'une matrice 3D contenant toutes les régions

Nous contruisons d'abord une matrice 3D contenant suivant la troisième dimension chacune des régions comme nous le voyons sur la figure A.5.

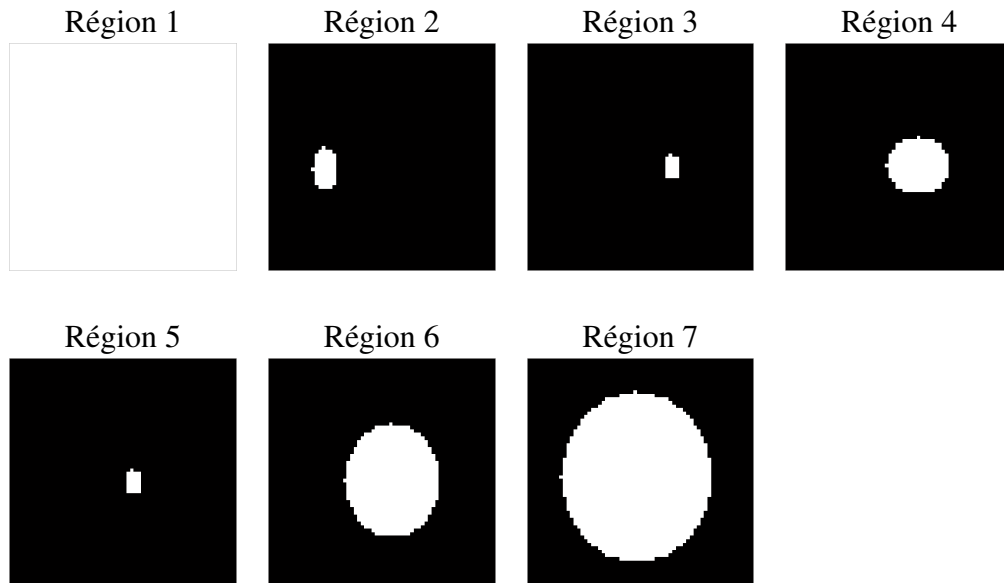


FIG. A.5 – Formation de la matrice 3D.

Nous effectuons pour cela un décodage de Freeman des contours codés (donc externes) de chaque région, puis un remplissage des régions.

Dans notre exemple, nous avons 7 régions. La classe 2 contient 2 régions (la région 2 et la région 3) et toutes les autres classes en contiennent chacune une seule. Or les régions codées dans le processus de codage pouvaient contenir des trous dont les pixels n'ont pas été codés (cas de la région 4). Il a donc été nécessaire de rechercher les intersections des régions.

A.2.2 Recherche des intersections

Cette étape permet de trouver les différentes intersections entre les régions et de définir un ordre de remplissage des régions (donné dans le tableau A.1).

Trois cas se présentent :

- tout d'abord le cas où il n'y a pas d'intersection entre la région X et la région Y (figure A.6) ;

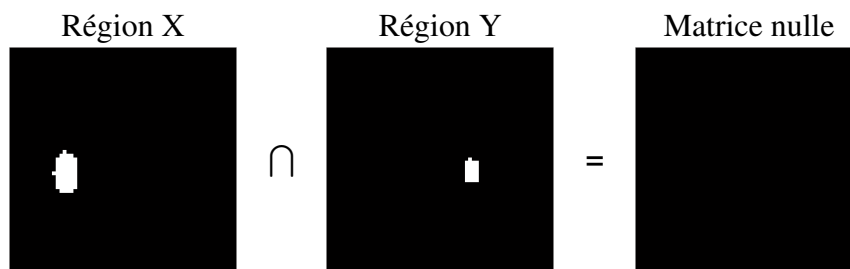


FIG. A.6 – Pas d'intersection.

- ensuite le cas où il y a une intersection entre la région X et la région Y, et que cette intersection est égale à la région Y : cela signifie que la région X contient la région Y (figure A.7) ;

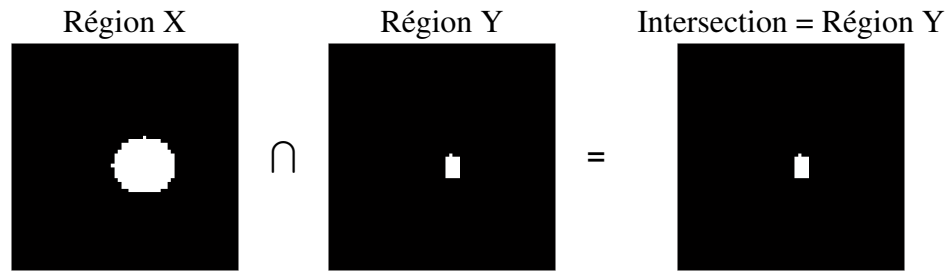


FIG. A.7 – La région X contient la région Y.

- enfin le cas où il y a une intersection entre la région X et la région Y, et que cette intersection est différente de la région Y : cela veut dire que la région Y contient la région X (figure A.8).

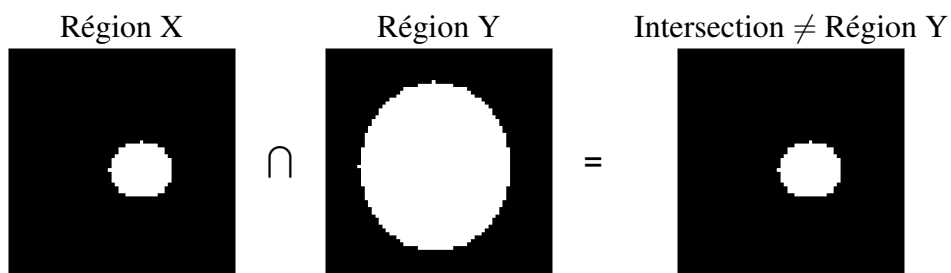


FIG. A.8 – La région Y contient la région X.

A.2.3 Construction de l'image classifiée

Une fois la phase de recherche d'intersections entre les régions réalisées, nous pouvons définir un ordre de remplissage des régions comme indiqué dans le tableau A.1.

	Région 1	Région 2	Région 3	Région 4	Région 5	Région 6	Région 7
Nombre de régions contenues dans la région :	6	0	0	2	0	3	5
Ordre de remplissage :	1	5	6	4	7	3	2

TAB. A.1 – Formation de l'image classifiée

Les régions contenant le plus de régions seront traitées en premier pour la reconstruction de l'image classifiée. L'ordre de traitement est fonction du nombre de régions contenues dans la région.

Dans notre exemple, la région 1 contient toutes les autres régions ; la région 6 contient les régions 4, 3 et 5 ; la région 4 contient les régions 3 et 5 ; et les régions 2, 3 et 5 n'en contiennent aucune.

L'ordre d'étude pour la construction de l'image classifiée sera donc (figure A.9) : région 1, puis région 7, puis région 6, puis région 4, puis région 2, puis région 3 et région 5.

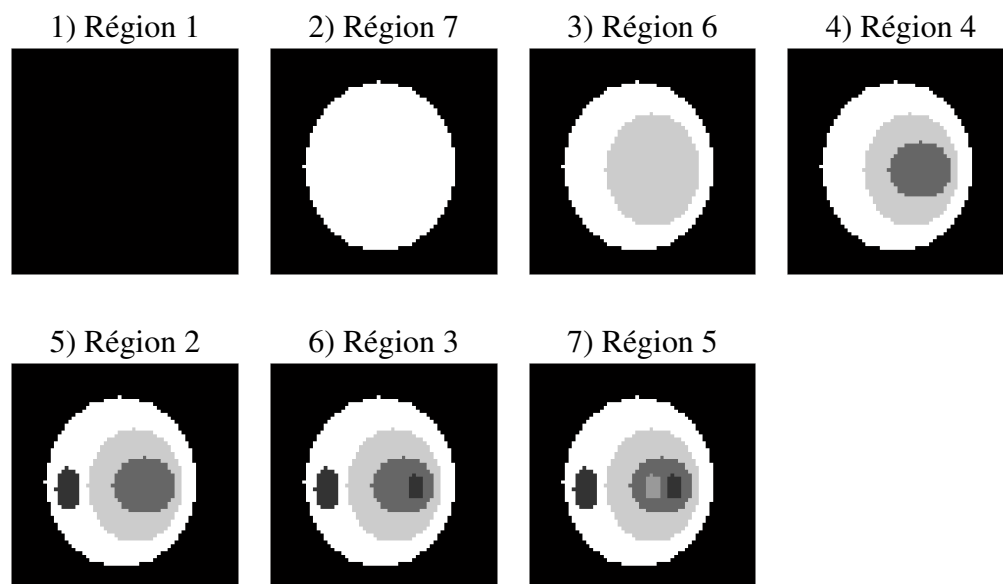


FIG. A.9 – Formation de l'image classifiée.

Nous obtenons alors l'image classifiée originale (voir figure 2.6 à la page 32).