

**ANALYSE MULTIRÉSOLUTION ET RECHERCHE
D'IMAGES**

par

Ksantini Riadh

mémoire présenté au Département de mathématiques et d'informatique en
vue de l'obtention du grade de maître ès sciences (M.Sc.)

**FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE**

Sherbrooke, Québec, Canada, avril 2003



National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisisitons et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 0-612-86658-0
Our file *Notre référence*
ISBN: 0-612-86658-0

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

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

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

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

Canada

Le 30/04/03,
Date

le jury a accepté le mémoire de M. Riadh Ksantini dans sa version finale.

Composition du jury

Membre : M. François Dubeau (Direction)
Département de mathématiques et d'informatique

Membre : M. Djemel Ziou
Département de mathématiques et d'informatique

Membre et
président-rapporteur : M. François Deschênes
Département de mathématique et d'informatique

SOMMAIRE

L'objectif de ce mémoire est de développer une méthode rapide et efficace de recherche d'images par contenu. Cette méthode sera utilisée pour des images 1D et pour des images 2D dont les caractéristiques seront représentées par des images 1D.

Une image 1D est un signal ou une fonction définie sur un intervalle. Elle peut être représentée par une fonction constante par morceaux, c'est-à-dire une fonction étagée. L'analyse multirésolution est une décomposition de cette fonction étagée suivant des ensembles de fonctions qui font apparaître les détails de l'image 1D en fonction des niveaux de résolution différents. Cette analyse multirésolution est produite grâce à une transformation en ondelettes de Haar qui décompose l'image 1D en ses composantes à différentes échelles. La compression par approximation s'effectue sur le nombre de coefficients de l'image 1D normalisée et décomposée. Si on néglige certains coefficients représentant les détails les plus fins, il est possible d'avoir une bonne approximation de l'image 1D.

Nous établissons un processus automatisé de recherche d'images par contenu qui s'applique à de grosses bases de données (BD). On choisit pour cela une métrique qui sera calculée d'une façon rapide entre la requête et les images de la BD. Afin de calculer notre métrique, nous associons à cette dernière des poids déterminés par un modèle de régression logistique.

Notre base de données d'images 1D est constituée par des mixtures de gaussiennes. Dans chaque mixture de gaussiennes, le nombre de composantes est identique au nombre de modes apparents. Pour évaluer notre méthode de recherche, nous avons étudié l'effet de plusieurs facteurs tels la translation, l'influence des poids, etc.

Afin de tester l'efficacité et l'utilité de notre méthode de recherche, nous appliquons cette dernière pour une base de données de poissons de Surray et une base de données d'images couleurs.

REMERCIEMENTS

Je voudrais exprimer tout d'abord mes remerciements aux professeurs qui m'ont fait l'honneur de participer au jury de ce mémoire.

Je remercie spécialement le professeur François Dubeau, mon directeur de recherche qui m'a appuyé par ses conseils, son soutien, sa grande disponibilité et la confiance qu'il m'a accordée durant toute la période de la recherche. Ceci a été d'une aide précieuse pour moi afin d'orienter et de bien guider mon travail. Qu'il trouve ici l'expression de ma gratitude pour l'intérêt qu'il porte à ce travail.

Je tiens aussi à exprimer mes remerciements au professeur Djemel Ziou, dont le soutien scientifique et financier m'a permis de mener à bien ce travail. J'ai particulièrement apprécié les nombreuses discussions que nous avons eu ensemble et qui ont guidé mes travaux.

Je remercie aussi le professeur Bernard Colin pour son soutien scientifique qui m'a permis d'utiliser la régression logistique dans mon travail.

La réalisation de ce travail a été rendue possible grâce à la participation financière de Bell Canada (programme de support à la R-D des laboratoires universitaires Bell), de l'Université de Sherbrooke (programme de bourses institutionnelles) et du CRSNG (programme de subvention individuelle et stratégique).

Je remercie enfin ma mère qui m'a soutenu et aidé pour l'aboutissement de mon travail.

TABLE DES MATIÈRES

SOMMAIRE	i
REMERCIEMENTS	iii
INTRODUCTION	1
CHAPITRE 1 — Images 1D et analyse multirésolution	4
1.1 Images 1D	4
1.2 L'analyse multirésolution	6
1.2.1 Ondelettes de Haar	6
1.2.2 Normalisation	9
1.3 Transformée en ondelettes de Haar	10
1.4 Compression par approximation	18
CHAPITRE 2 — Recherche d'images 1D	31
2.1 Recherche d'images 1D	31

2.2	Construction de la métrique	32
2.2.1	Motivation	32
2.2.2	Métrique	33
2.2.3	Tableaux de recherche	35
2.2.4	Organigramme de la recherche	38
2.3	Calcul des poids	39
2.3.1	Calcul des poids	39
2.3.2	Échantillonnage	40
2.3.3	Régression logistique	41
2.4	Méthode de recherche	45
CHAPITRE 3 — Base de données 1D		47
3.1	Base de données de mixtures de gaussiennes 1D	47
3.2	Recherche d'image 1D	48
3.2.1	Exemple	48
3.3	Évaluation	70
3.3.1	Réglage : nombre de coefficients gardés	70
3.3.2	Influence des poids	71
3.3.3	Translation	73
3.3.4	Bruit gaussien dans la requête	75
3.3.5	Multiplication d'intensité	78

3.3.6	Vitesse d'exécution (complexité)	78
CHAPITRE 4	— Applications	80
4.1	Base de données de poissons de Surray	80
4.2	Base de données d'images	90
CONCLUSION		100
Bibliographie		102

LISTE DES TABLEAUX

3.1	Les poids w_0 et w_i ($i = 0,1,..,7$) associés à notre métrique pour différents ordres m de la compression de la requête et des images de la BD.	49
3.2	Les valeurs des ordres m de la compression pour lesquelles on trouve des bons résultats de la recherche, malgré les bruits gaussiens d'écart-type σ_b perturbant la requête.	75

LISTE DES FIGURES

1.1	Exemple d'une image de 4 pixels représentée par l'ensemble des fonctions ϕ_i^J ($J = 2$).	5
1.2	La fonction d'échelle et l'ondelette de Haar.	7
1.3	Orthogonalité entre deux ondelettes de Haar.	8
1.4	Raffinement de fonctions ϕ et ψ normalisées.	10
1.5	Décomposition d'une image de 8 pixels.	14
1.6	Décomposition d'une image de 16 pixels.	15
1.7	Effet de la dimension de l'image sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.	22
1.8	Effet de la dimension de l'image bruitée sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.	23
1.9	Effet du bruit gaussien sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.	24
1.10	Compression 1D: Erreur relative 0.086% et nombre de coefficients gardés 102.	25

1.11	Compression 1D: Erreur relative 0.18% et nombre de coefficients gardés 76 .	26
1.12	Compression 1D: Erreur relative 0.41% et nombre de coefficients gardés 51 .	27
1.13	Compression 1D: Erreur relative 1.22% et nombre de coefficients gardés 25 .	28
1.14	Compression 1D: Erreur relative 4.63% et nombre de coefficients gardés 10 .	29
1.15	Compression 1D: Erreur relative 6.29% et nombre de coefficients gardés 7.	30
2.1	Ajout de la $n^{\text{ième}}$ image aux tableaux Θ_+ et Θ_-	37
2.2	Organigramme de la recherche.	38
2.3	Courbe de la fonction F de distribution de l'erreur logistique u_i	43
2.4	Courbe de la Fonction Logit de distribution des probabilités P_i	44
3.1	Recherche avec compression à 100 coefficients. Les 8 images cibles 1D les plus proches de la requête.	50
3.2	Recherche avec compression à 100 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	51
3.3	Recherche avec compression à 90 coefficients. Les 8 images cibles 1D les plus proches de la requête.	52
3.4	Recherche avec compression à 90 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	53
3.5	Recherche avec compression à 80 coefficients. Les 8 images cibles 1D les plus proches de la requête.	54

3.6	Recherche avec compression à 80 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	55
3.7	Recherche avec compression à 70 coefficients. Les 8 images cibles 1D les plus proches de la requête.	56
3.8	Recherche avec compression à 70 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	57
3.9	Recherche avec compression à 60 coefficients. Les 8 images cibles 1D les plus proches de la requête.	58
3.10	Recherche avec compression à 60 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	59
3.11	Recherche avec compression à 50 coefficients. Les 8 images cibles 1D les plus proches de la requête.	60
3.12	Recherche avec compression à 50 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	61
3.13	Recherche avec compression à 40 coefficients. Les 8 images cibles 1D les plus proches de la requête.	62
3.14	Recherche avec compression à 40 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	63

3.15 Recherche avec compression à 30 coefficients. Les 8 images cibles 1D les plus proches de la requête.	64
3.16 Recherche avec compression à 30 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	65
3.17 Recherche avec compression à 20 coefficients. Les 8 images cibles 1D les plus proches de la requête.	66
3.18 Recherche avec compression à 20 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	67
3.19 Recherche avec compression à 10 coefficients. Les 8 images cibles 1D les plus proches de la requête.	68
3.20 Recherche avec compression à 10 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	69
3.21 Les moyennes des rangs des dernières cibles désirées trouvées parmi les 24 premières images atteintes par les recherches, pour plusieurs ordres de compression de chaque requête et des images de l'échantillon de notre BD.	71
3.22 Comparaison avec $m = 100$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.	72
3.23 Comparaison avec $m = 80$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.	72
3.24 Comparaison avec $m = 60$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.	73

3.25	Recherche : a) Requête originale. b) Requête translatée de 3 unités vers le côté gauche. c) Requête translatée de 5 unités vers le côté gauche.	74
3.26	Recherche : requête perturbée par un bruit gaussien $b \sim N(0, \sigma_b)$ avec $\sigma_b = 3 * 10^{-3}$. a) $m = 100$. b) $m = 60$. c) $m = 40$	76
3.27	Recherche : requête perturbée par un bruit gaussien $b \sim N(0, \sigma_b)$ avec $\sigma_b = 8 * 10^{-3}$. a) $m = 100$. b) $m = 60$. c) $m = 10$	77
4.1	Exemple : a) Un poisson de Surray de la BD. b) L'histogramme du poisson et c) L'image 1D de 2^8 pixels associée au poisson de Surray.	83
4.2	Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	85
4.3	Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	86
4.4	Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	87
4.5	Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	88
4.6	Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	89

4.7	Exemple: a) Une image couleurs de la BD. b) La version convertie de l'image couleurs en une image à niveaux de gris. c) L'histogramme pondéré par le laplacien de l'image.	92
4.8	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	93
4.9	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	94
4.10	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	95
4.11	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	96
4.12	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	97
4.13	Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées: a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.	98
4.14	Comparaison: a) Recherche basée sur les histogrammes pondérés par le laplacien. b) Recherche basée sur les histogrammes à niveaux de gris. . . .	99

INTRODUCTION

La recherche d'information désigne l'opération qui permet, à partir d'une expression des besoins en information d'un utilisateur, de retrouver l'ensemble des documents contenant l'information recherchée. La notion de document est prise ici au sens large et peut représenter un document textuel, une image, un graphique ou une combinaison de ces médias. Au cours des dernières années, nous avons assisté à une croissance énorme de l'information disponible dans des banques (ou bases) de données, comme les banques d'images dans un centre de télédétection ou les banques d'images et de textes concernant des patients dans un centre médical. Le problème avec ces informations est qu'elles ne sont pas stockées d'une manière structurée. Par conséquent, la recherche d'information dans des grandes bases de données, en particulier des bases d'images, est un problème d'une importance cruciale. De ce fait, plusieurs techniques pour manipuler les grandes bases de données d'images commencent à être explorées. D'après [8] et [9], deux approches ont été développées en recherche d'images issues de deux points de vue différents : une approche basée sur les attributs des images et une approche basée sur leur contenu. Dans l'approche basée sur les attributs des images, chaque image de la base de données d'images est représentée par un ensemble d'attributs textuels externes et internes. Ensuite, la recherche se fait sur ces attributs de manière classique et analogue aux systèmes de recherche d'informations traditionnels. Cette approche a été développée à l'origine pour les documents textuels. Cependant, elle présente l'inconvénient de la grande subjectivité dans l'association des

attributs aux images. Aussi, les procédures d'indexation ne sont pas automatiques. Par conséquent, l'utilisation de l'approche basée seulement sur les attributs, dans un système de recherche d'images, est insuffisante et inadaptée. Dans l'approche basée sur le contenu des images, chaque image est représentée par un ensemble de traits physiques ou de caractéristiques géométriques et photométriques telles que la couleur, la texture, etc. Cette approche évite le problème de subjectivité dans l'association d'attributs à l'image. Ensuite les procédures d'indexation peuvent être automatiques. D'après [8], la couleur est la caractéristique visuelle la plus utilisée dans la recherche d'images par contenu. Cette caractéristique est souvent représentée par un histogramme. Par conséquent, la similarité entre deux images est mesurée par une métrique calculée entre leurs histogrammes de couleurs, telle que L_1 ou L_2 (norme 1 et norme 2, respectivement). Aussi, une segmentation des textures des images pourra être utilisée pour la recherche d'images par contenu. Dans ce cas, les images sont décomposées en régions de textures uniformes et la similarité entre deux images est déterminée par une combinaison des similarités entre les différentes régions des deux images.

La recherche d'images par contenu a des applications dans de nombreux domaines : graphisme, architecture, imagerie médicale, sécurité, multimédia, arts, géologie, etc. Le problème rencontré en recherche d'images par contenu est que, souvent, on cherche une image dans une base de données (BD) à partir d'information partielle donnée, par exemple, par une esquisse de l'image cherchée, une photo obtenue à faible résolution, etc. Il faut alors que la méthode de recherche de l'image soit assez robuste pour accepter que la requête (l'image à l'entrée) et la cible (l'image cherchée ou le résultat de la recherche) soient assez différentes tout en ayant les mêmes caractéristiques désirées. Une façon de faire une recherche efficace est d'établir un outil (métrique) qui mesure la "distance" entre deux images sur la base des caractéristiques sélectionnées. Cet outil doit être assez flexible pour tenir compte des distorsions de la requête par rapport à la cible, et aussi assez rapide

d'exécution pour pouvoir être utilisé sur de grandes bases de données.

Il est possible d'utiliser une décomposition en ondelettes de la requête et des images de la BD de façon à exécuter la recherche efficacement en terme de rapidité d'exécution et de qualité des résultats de la recherche. L'analyse multirésolution décrite par [2], [3] et [5], étant une décomposition d'un signal (image) suivant des ensembles de fonctions qui font apparaître les détails du signal en fonction du niveau de résolution, les caractéristiques grossières et les détails fins peuvent être identifiés. On peut utiliser ce fait pour la recherche d'image particulière dans une BD contenant un grand nombre d'images. La recherche se fait en établissant une signature pour chacune des images à partir de leurs décompositions en ondelettes et en calculant une "distance" entre une image requête et les images de la BD. Afin de calculer notre métrique proposée par [2], nous devons déterminer des poids grâce à un modèle de régression logistique spécifique généré par le logiciel "SAS". Pour valider notre méthode de recherche, nous appliquons cette dernière pour une base de données de poissons de Surray et une base de données d'images couleurs.

Notre méthode est basée sur la décomposition d'un signal 1D, appelé ici histogramme. On peut utiliser notre méthode pour des images 2D en utilisant une caractéristique de l'image qui se représente par un signal 1D.

Dans ce mémoire, le chapitre 1 sert à préciser la notion d'image 1D et à présenter l'analyse multirésolution à l'aide des ondelettes de Haar, la transformée en ondelettes et la compression par approximation. La construction de la métrique, la structure de données utilisée et la régression logistique font l'objet du chapitre 2. Le chapitre 3 décrit la construction de notre base de mixtures de gaussiennes et l'évaluation de notre méthode de recherche. Finalement, le chapitre 4 présente l'application de notre méthode de recherche pour une base de données de poissons de Surray et une base de données d'images couleurs.

CHAPITRE 1

Images 1D et analyse multirésolution

1.1 Images 1D

Dans le cas 1D, une image est un signal ou une fonction définie sur un intervalle appelé le support de l'image. Dans ce qui suit, le support sera l'intervalle $[0,1)$. Elle peut être donnée par une suite finie de valeurs représentant l'intensité de l'image sur chaque pixel de l'image. Elle peut aussi être représentée par une fonction constante par morceaux, c'est-à-dire une fonction étagée. Pour établir cette représentation, supposons que l'image est supportée par 2^J pixels ($J \in \mathbb{N}$) et que la suite des coefficients est $\{c_i^J\}_{i=0}^{2^J-1}$. On subdivise l'intervalle $[0,1)$ en 2^J sous intervalles de longueur 2^{-J} de la forme

$$I_i^J = 2^{-J}[i, i+1) \quad \text{pour} \quad i = 0, \dots, 2^J - 1. \quad (1.1)$$

Chaque pixel est représenté par un intervalle I_i^J . Pour définir la fonction étagée, on considère une fonction ϕ , dite fonction d'échelle de Haar, définie par ([3],[7])

$$\phi(x) = \begin{cases} 1 & \text{si } x \in [0,1), \\ 0 & \text{sinon,} \end{cases} \quad (1.2)$$

Les versions translatées et dilatées du facteur 2^J de cette fonction sont notées $\phi_i^J(x) = \phi(2^J x - i)$ et le support des ϕ_i^J est égal à celui de I_i^J . Plus formellement

$$\text{support } \phi_i^J = \{x | \phi_i^J(x) \neq 0\} = I_i^J. \quad (1.3)$$

Cet ensemble de fonctions constitue une base orthogonale de $V^J = \text{Lin}\{\phi_i^J : i = 0, \dots, 2^J - 1\}$ puisque

$$\langle \phi_i^J, \phi_k^J \rangle = 0 \quad \text{pour} \quad i \neq k, \quad (1.4)$$

sous le produit scalaire habituel

$$\langle f, g \rangle = \int_0^1 f(x)g(x)dx. \quad (1.5)$$

En notant le niveau de gris du pixel i de l'image par c_i^J , l'image entière s'écrit

$$I(x) = \sum_{i=0}^{2^J-1} c_i^J \phi_i^J(x). \quad (1.6)$$

La Figure 1.1 montre l'exemple d'une image de quatre pixels représentée sous la forme (1.6). À cette étape, les coefficients c_i^J ne sont que les valeurs associées au niveau de gris des pixels originaux de l'image.

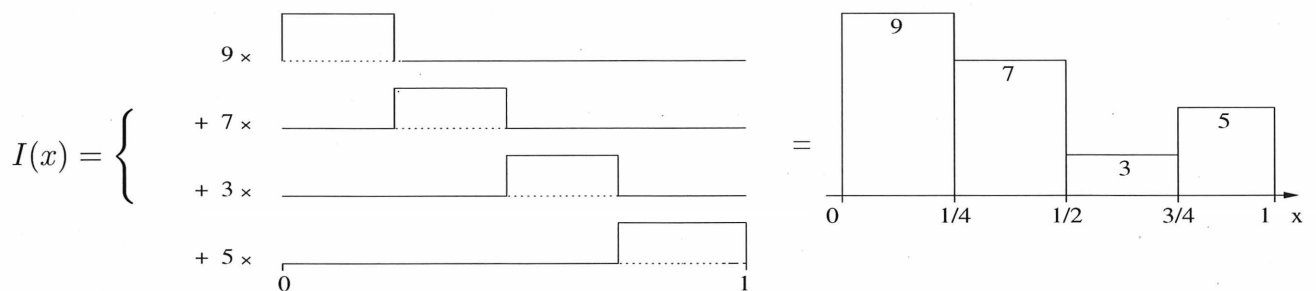


Figure 1.1 – Exemple d'une image de 4 pixels représentée par l'ensemble des fonctions ϕ_i^J ($J = 2$).

1.2 L'analyse multirésolution

1.2.1 Ondelettes de Haar

L'analyse multirésolution (AMR) nous permet de représenter une image à des niveaux de résolution différents, chacun d'eux étant différent du précédent par un facteur de deux. Une résolution de niveau j ($j = 0, \dots, J$), notée V^j , est l'espace vectoriel des combinaisons linéaires des fonctions $\phi_i^j(x) = \phi(2^j x - i)$, c'est-à-dire

$$V^j = \text{Lin}\{\phi_i^j : i = 0, \dots, 2^j - 1\}. \quad (1.7)$$

Un élément de V^j est une image, une fonction étagée, avec 2^j pixels de longueur 2^{-j} . La résolution de niveau J requiert 2^J pixels, soit l'image entière. Elle est la plus fine résolution possible parmi l'ensemble que permet l'AMR produite avec les ondelettes de Haar. Pour changer de résolution, nous regroupons les pixels deux à deux. Nous utilisons une paire de fonctions ϕ^j adjacentes pour former une fonction ϕ^{j-1} grâce à l'équation dite de raffinement

$$\phi_i^{j-1}(x) = \phi_{2i}^j(x) + \phi_{2i+1}^j(x) \quad \text{avec} \quad i = 0, \dots, 2^{j-1} - 1. \quad (1.8)$$

Cette équation sert à exprimer une fonction de base de V^{j-1} en fonction de celles de V^j . Ces changements de résolution introduisent une suite d'espaces vectoriels $\{V^j\}_{j=0}^J$ emboîtés de la façon suivante

$$\overbrace{V^0 \subset \dots \subset V^{j-1} \subset V^j \subset V^{j+1} \subset \dots \subset V^J}^{\substack{\text{Grossier} \quad \longleftrightarrow \quad \text{Détailé}}}. \quad (1.9)$$

Une image de 2^J pixels génère $J + 1$ résolutions différentes puisqu'on se limite au support $[0,1)$. Comme $V^{j-1} \subset V^j$, de l'information est perdue lorsque l'on passe à une résolution plus grossière. Afin de garder les détails de cette information on peut trouver un ensemble W^{j-1} tel que $V^{j-1} \oplus W^{j-1} = V^j$ où \oplus est la somme directe des deux espaces. Pour définir

W^{j-1} , on introduit l'ondelette mère

$$\psi(x) = \begin{cases} 1 & \text{si } x \in [0, \frac{1}{2}), \\ -1 & \text{si } x \in [\frac{1}{2}, 1), \\ 0 & \text{sinon,} \end{cases} \quad (1.10)$$

et on note les versions translatées et dilatées de cette fonction par $\psi_i^j(x) = \psi(2^j x - i)$. L'espace vectoriel engendré par les fonctions ψ_i^j sera

$$W^j = \text{Lin}\{\psi_i^j : i = 0, \dots, 2^j - 1\}. \quad (1.11)$$

Ces fonctions sont les ondelettes de Haar. Elles sont orthogonales aux fonctions ϕ comme nous montre la Figure 1.2 puisque $\langle \phi, \psi \rangle = 0$. Ainsi, on a

$$W^j \perp V^j. \quad (1.12)$$

De plus

$$\psi_i^{j-1}(x) = \phi_{2i}^j(x) - \phi_{2i+1}^j(x) \quad \text{avec} \quad i = 0, 1, \dots, 2^{j-1} - 1, \quad (1.13)$$

ainsi $W^{j-1} \subset V^j$. À l'aide de (1.8) et (1.13) nous obtenons

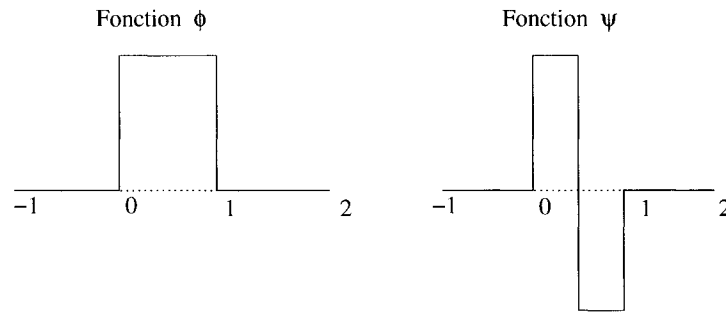


Figure 1.2 – La fonction d'échelle et l'ondelette de Haar.

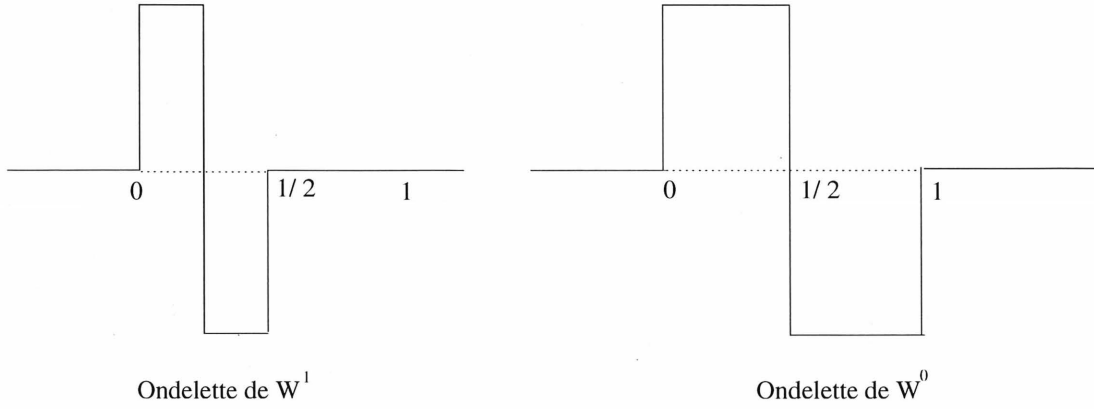


Figure 1.3 – Orthogonalité entre deux ondelettes de Haar.

$$\phi_{2i}^j(x) = \frac{1}{2}[\phi_i^{j-1}(x) + \psi_i^{j-1}(x)] \quad (1.14)$$

$$\phi_{2i+1}^j(x) = \frac{1}{2}[\phi_i^{j-1}(x) - \psi_i^{j-1}(x)]$$

pour $i = 0, 1, \dots, 2^{j-1} - 1$. On en déduit

$$V^{j-1} \oplus W^{j-1} = V^j. \quad (1.15)$$

pour $j = 1, \dots, J$. Notons qu'on obtient $W^{j-1} \perp W^j$ de (1.15), ce qu'on peut également montrer directement à l'aide de la définition des ψ_i^j (voir la Figure 1.3).

L'image peut donc se réécrire

$$I(x) = \underbrace{\sum_{i=0}^{2^{J-1}-1} c_i^{J-1} \phi_i^{J-1}(x)}_{I_g(x)} + \underbrace{\sum_{i=0}^{2^{J-1}-1} d_i^{J-1} \psi_i^{J-1}(x)}_{I_d(x)} \quad (1.16)$$

où $I_g(x)$ est une image 1D, moins fine que $I(x)$, ayant des pixels deux fois plus larges mais un nombre de pixels deux fois moins grand, tandis que $I_d(x)$ représente les détails gardés de l'information perdue lors du passage d'un niveau de résolution J à un niveau de résolution $J - 1$.

1.2.2 Normalisation

Afin de simplifier et d'accélérer les calculs, nous utilisons les bases suivantes de fonctions normalisées

$$\Phi^j = \{\tilde{\phi}_i^j(x) | i = 0, \dots, 2^j - 1\} \subset V^j \quad (1.17)$$

et

$$\Psi^j = \{\tilde{\psi}_i^j(x) | i = 0, \dots, 2^j - 1\} \subset W^j, \quad (1.18)$$

où $\tilde{\phi}_i^j$ et $\tilde{\psi}_i^j$ sont maintenant des fonctions de base normalisées. Les conditions de normalisation, $\langle \tilde{\phi}_i^j, \tilde{\phi}_i^j \rangle = 1$ et $\langle \tilde{\psi}_i^j, \tilde{\psi}_i^j \rangle = 1$, nous donnent $\tilde{\phi}_i^j(x) = \sqrt{2^j} \phi(2^j x - i)$ et $\tilde{\psi}_i^j(x) = \sqrt{2^j} \psi(2^j x - i)$. On a donc

$$\tilde{\phi}_i^j(x) = \begin{cases} 2^{\frac{j}{2}} & \text{si } x \in 2^{-j} [i, i + 1), \\ 0 & \text{sinon,} \end{cases} \quad (1.19)$$

et

$$\tilde{\psi}_i^j(x) = \begin{cases} 2^{\frac{j}{2}} & \text{si } x \in 2^{-j} [i, i + \frac{1}{2}) \\ -2^{\frac{j}{2}} & \text{si } x \in 2^{-j} [i + \frac{1}{2}, i + 1) \\ 0 & \text{sinon.} \end{cases} = \begin{cases} 2^{-(j+1)} [2i, 2i + 1), \\ 2^{-(j+1)} [2i + 1, 2i + 2), \end{cases} \quad (1.20)$$

La Figure 1.4 illustre les équations de raffinement suivantes

$$\tilde{\phi}_i^{j-1}(x) = \frac{1}{\sqrt{2}} (\tilde{\phi}_{2i}^j(x) + \tilde{\phi}_{2i+1}^j(x)) \quad (1.21)$$

et

$$\tilde{\psi}_i^{j-1}(x) = \frac{1}{\sqrt{2}} (\tilde{\phi}_{2i}^j(x) - \tilde{\phi}_{2i+1}^j(x)) \quad (1.22)$$

pour $j = 1$.

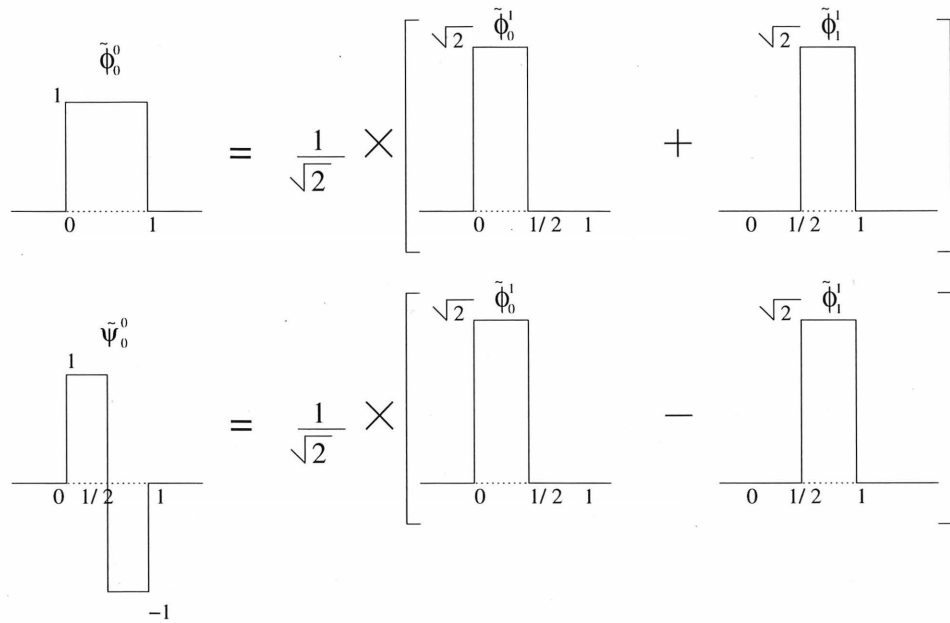


Figure 1.4 – *Raffinement de fonctions ϕ et ψ normalisées.*

La combinaison linéaire des fonctions normalisées de la base Φ^{J-1} représente une image 1D ayant la moitié des pixels de celle donnée par une combinaison linéaire des éléments de Φ^J , tandis que la combinaison linéaire des fonctions normalisées de la base Ψ^{J-1} représente les détails perdus lors du passage du niveau de résolution J au niveau de résolution $J - 1$.

1.3 Transformée en ondelettes de Haar

La transformation en ondelettes décompose l'image en ses composantes à différentes échelles. Elle consiste à passer successivement de l'espace V^J à l'espace V^0 , tout en générant les différents W^j , de telle sorte que

$$V^J = V^0 \oplus \bigoplus_{k=0}^{J-1} W^k. \quad (1.23)$$

Soit $I(x)$ une image 1D appartenant à V^J ,

$$I(x) = \sum_{i=0}^{2^J-1} \tilde{c}_i^J \tilde{\phi}_i^J(x). \quad (1.24)$$

Comme on l'a montré, on peut réécrire cette image comme

$$I(x) = \sum_{i=0}^{2^{J-1}-1} c_i^{J-1} \tilde{\phi}_i^{J-1}(x) + \sum_{i=0}^{2^{J-1}-1} d_i^{J-1} \tilde{\psi}_i^{J-1}(x). \quad (1.25)$$

La deuxième sommation tient compte des ondelettes de W^{J-1} . Les coefficients de cette dernière restent inchangés jusqu'à la fin de la décomposition. En décomposant successivement les sommations qui portent sur V^j , l'image décomposée s'écrit

$$I(x) = \underbrace{c_0^0 \tilde{\phi}_0^0(x)}_{\in V^0} + \sum_{k=0}^{J-1} \underbrace{\sum_{i=0}^{2^k-1} d_i^k \tilde{\psi}_i^k(x)}_{\in W^k} \quad (1.26)$$

Le coefficient de $\tilde{\phi}_0^0$ correspond à la moyenne des niveaux de gris (l'intensité) de l'image qu'on nomme facteur d'échelle. Maintenant qu'on comprend comment les fonctions se combinent pour générer une AMR, montrons comment deux pixels sont simplifiés en un seul grâce à un changement de résolution. Examinons la relation entre les coefficients de V^j et ceux de V^{j-1} sur l'intervalle I_k^{j-1} . Pour ce faire, utilisons les deux représentations (1.24) et (1.25) de l'image et isolons les coefficients qui agissent sur I_k^{j-1} . On utilise

d'abord $\tilde{\phi}_{2k}^j$

$$\langle \tilde{\phi}_{2k}^j, I \rangle = \langle \tilde{\phi}_{2k}^j, \sum_{i=0}^{2^j-1} c_i^j \tilde{\phi}_i^j \rangle \quad (1.27)$$

$$= \sum_{i=0}^{2^j-1} c_i^j \langle \tilde{\phi}_{2k}^j, \tilde{\phi}_i^j \rangle \quad (1.28)$$

$$= c_{2k}^j \quad (1.29)$$

$$\langle \tilde{\phi}_{2k}^j, I \rangle = \langle \tilde{\phi}_{2k}^j, \sum_{i=0}^{2^{j-1}-1} c_i^{j-1} \tilde{\phi}_i^{j-1} + \sum_{i=0}^{2^{j-1}-1} d_i^{j-1} \tilde{\psi}_i^{j-1} \rangle \quad (1.30)$$

$$= \langle \tilde{\phi}_{2k}^j, \sum_{i=0}^{2^{j-1}-1} c_i^{j-1} \tilde{\phi}_i^{j-1} \rangle + \langle \tilde{\phi}_{2k}^j, \sum_{i=0}^{2^{j-1}-1} d_i^{j-1} \tilde{\psi}_i^{j-1} \rangle \quad (1.31)$$

$$= \sum_{i=0}^{2^{j-1}-1} c_i^{j-1} \langle \tilde{\phi}_{2k}^j, \tilde{\phi}_i^{j-1} \rangle + \sum_{i=0}^{2^{j-1}-1} d_i^{j-1} \langle \tilde{\phi}_{2k}^j, \tilde{\psi}_i^{j-1} \rangle \quad (1.32)$$

$$= c_k^{j-1} \langle \tilde{\phi}_{2k}^j, \tilde{\phi}_k^{j-1} \rangle + d_k^{j-1} \langle \tilde{\phi}_{2k}^j, \tilde{\psi}_k^{j-1} \rangle \quad (1.33)$$

$$= \frac{1}{\sqrt{2}} [c_k^{j-1} + d_k^{j-1}], \quad (1.34)$$

car on a

$$\begin{aligned} \tilde{\phi}_{2k}^j(x) \neq 0 &\iff x \in I_{2k}^j, \\ \tilde{\phi}_{2k+1}^j(x) \neq 0 &\iff x \in I_{2k+1}^j, \\ \tilde{\phi}_i^{j-1}(x) \neq 0 &\iff x \in I_i^{j-1}, \\ \tilde{\psi}_i^{j-1}(x) \neq 0 &\iff x \in I_i^{j-1}, \end{aligned}$$

et

$$\langle \tilde{\phi}_{2k}^j, \tilde{\phi}_k^{j-1} \rangle = \frac{1}{\sqrt{2}} = \langle \tilde{\phi}_{2k+1}^j, \tilde{\phi}_k^{j-1} \rangle \quad (1.35)$$

$$\langle \tilde{\phi}_{2k}^j, \tilde{\psi}_k^{j-1} \rangle = \frac{1}{\sqrt{2}} = -\langle \tilde{\phi}_{2k+1}^j, \tilde{\psi}_k^{j-1} \rangle. \quad (1.36)$$

De même, en multipliant les versions (1.24) et (1.25) de l'image par $\tilde{\phi}_{2k+1}^j$, on obtient

$$\langle \tilde{\phi}_{2k+1}^j, I \rangle = c_{2k+1}^j \quad (1.37)$$

$$\langle \tilde{\phi}_{2k+1}^j, I \rangle = \frac{1}{\sqrt{2}}[c_k^{j-1} - d_k^{j-1}]. \quad (1.38)$$

En écrivant le tout sous forme matricielle, on obtient les égalités matricielles suivantes

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} c_k^{j-1} \\ d_k^{j-1} \end{pmatrix} = \begin{pmatrix} c_{2k}^j \\ c_{2k+1}^j \end{pmatrix} \quad (1.39)$$

représentant la reconstruction de l'image 1D et

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} c_{2k}^j \\ c_{2k+1}^j \end{pmatrix} = \begin{pmatrix} c_k^{j-1} \\ d_k^{j-1} \end{pmatrix} \quad (1.40)$$

représentant la décomposition de notre image en ondelettes de Haar. La relation de changement de résolution pour les fonctions est donnée par

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \tilde{\phi}_{2k}^j \\ \tilde{\phi}_{2k+1}^j \end{pmatrix} = \begin{pmatrix} \tilde{\phi}_k^{j-1} \\ \tilde{\psi}_k^{j-1} \end{pmatrix} \quad (1.41)$$

et la reconstruction est donnée par

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \tilde{\phi}_k^{j-1} \\ \tilde{\psi}_k^{j-1} \end{pmatrix} = \begin{pmatrix} \tilde{\phi}_{2k}^j \\ \tilde{\phi}_{2k+1}^j \end{pmatrix}. \quad (1.42)$$

La Figure 1.5 représente un exemple de la décomposition des coefficients d'une image de 8 pixels. La Figure 1.6 nous montre la décomposition d'une image de 16 pixels d'un point de vue graphique.

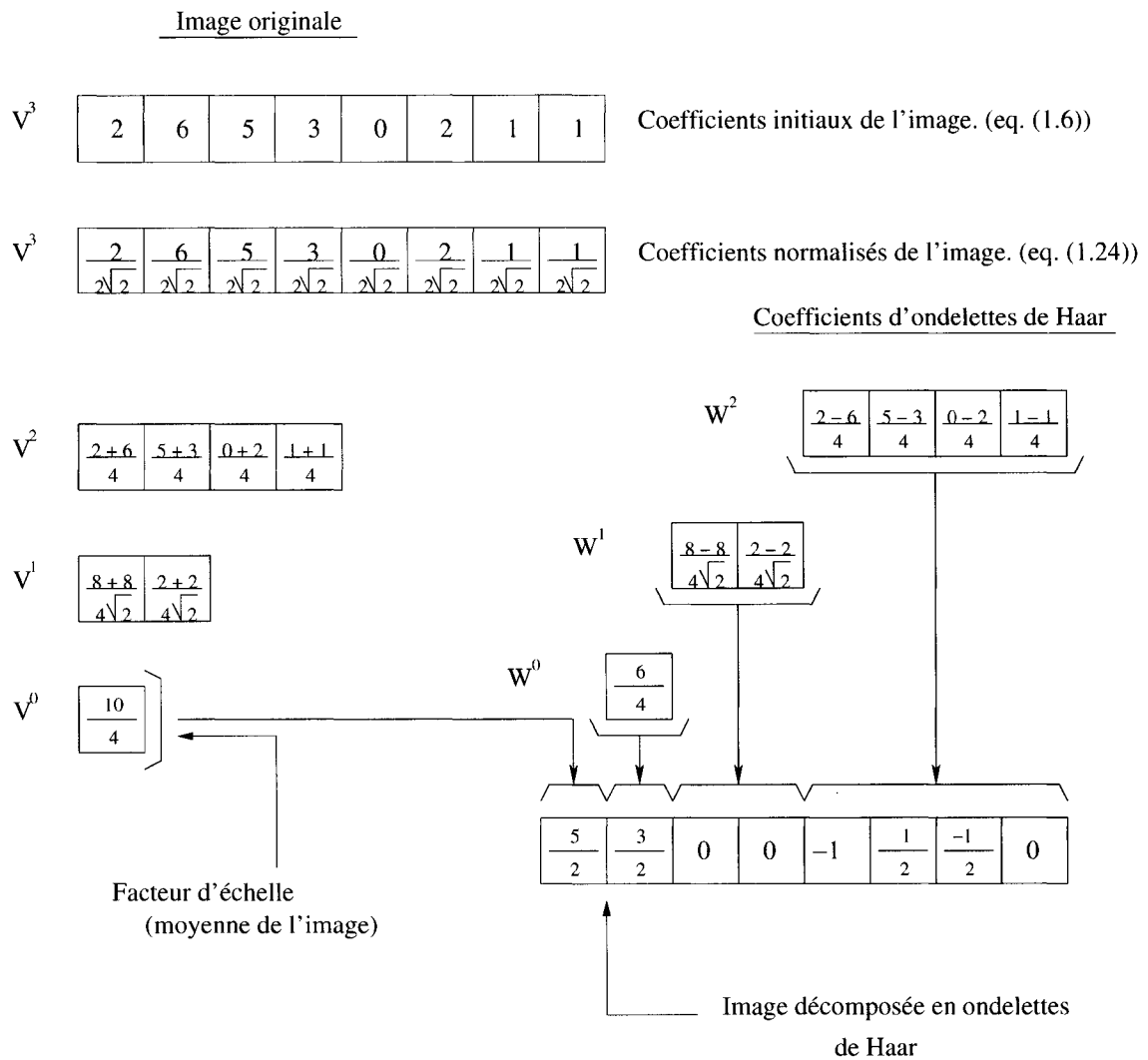


Figure 1.5 – Décomposition d'une image de 8 pixels.

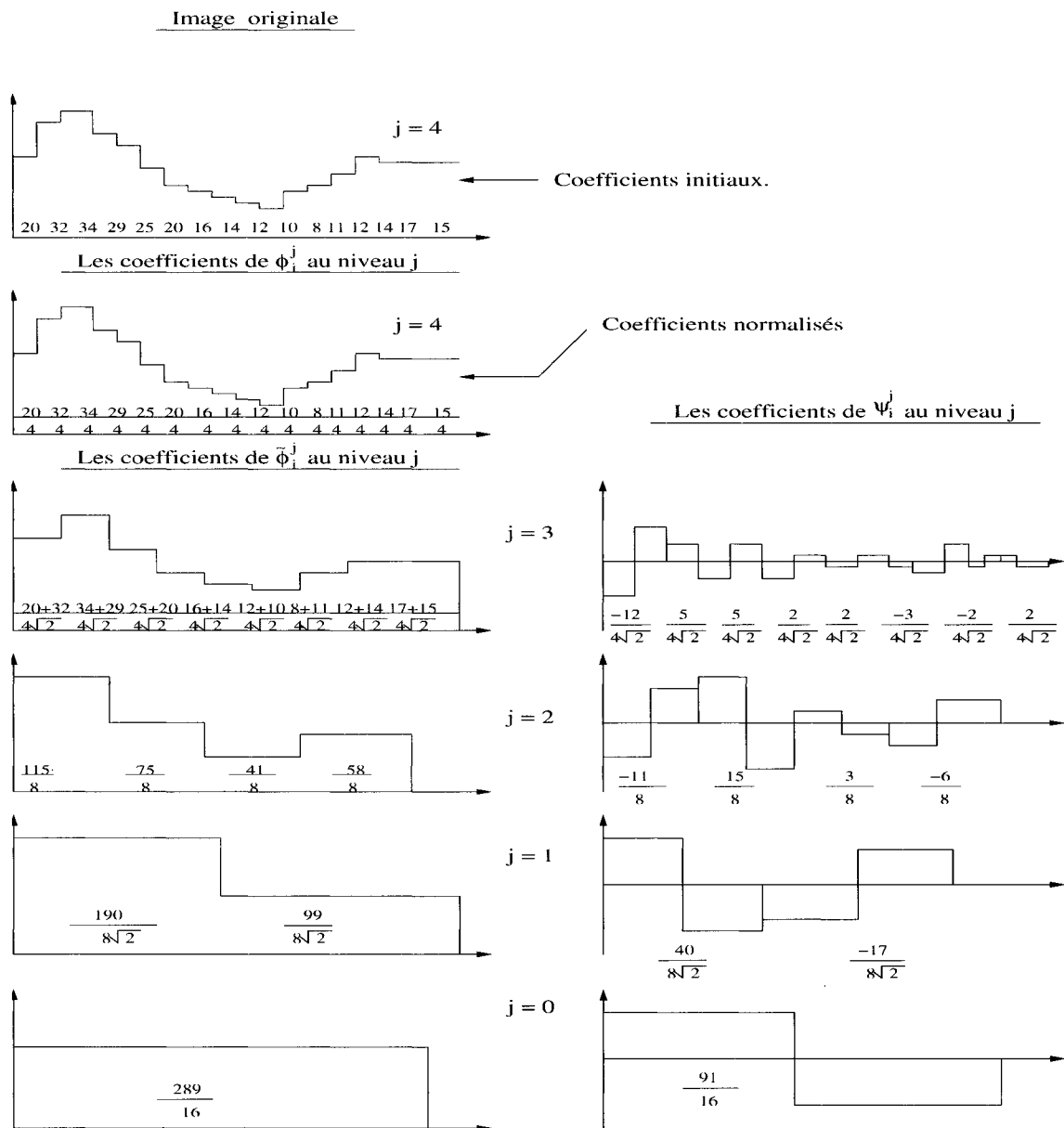


Figure 1.6 – Décomposition d'une image de 16 pixels.

Voici les procédures de décomposition et de reconstruction en ondelettes de Haar d'une image 1D ([3],[6]):

Procédure Décomposition(c: tableau [1..2^j] de réels)

$c \leftarrow c/\sqrt{2^j}$ (normalisation initiale)

$g \leftarrow 2^j$

Tant que $g \geq 2$ **faire**

 ÉtapeDécomposition(c[1..g])

$g \leftarrow g/2$

Fin tant que

Fin procédure

Procédure ÉtapeDécomposition(c: tableau[1..2^j] de réels)

Pour $i \leftarrow 1$ à 2^{j-1} **faire**

$c'[i] \leftarrow (c[2i-1] + c[2i])/\sqrt{2}$

$c'[2^{j-1} + i] \leftarrow (c[2i-1] - c[2i])/\sqrt{2}$

Fin pour

$c \leftarrow c'$

Fin procédure

Procédure Reconstruction(c : tableau $[1..2^j]$ de réels)

$g \leftarrow 2$

Tant que $g \leq 2^j$ **faire**

 ÉtapeReconstruction($c[1..g]$)

$g \leftarrow 2g$

Fin tant que

$c \leftarrow c\sqrt{2^j}$ (dénormalisation finale)

Fin procédure

Procédure ÉtapeReconstruction(c : tableau $[1..2^j]$ de réels)

Pour $i \leftarrow 1$ à 2^{j-1} **faire**

$c'[2i - 1] \leftarrow (c[i] + c[2^{j-1} + i])/\sqrt{2}$

$c'[2i] \leftarrow (c[i] - c[2^{j-1} + i])/\sqrt{2}$

Fin pour

$c \leftarrow c'$

Fin procédure

1.4 Compression par approximation

La compression s'effectue sur le nombre de coefficients retenus pour représenter l'image. Les coefficients d'ondelettes représentent des variations locales d'intensité dans l'image. L'amplitude des coefficients d'ondelettes représente directement l'importance d'une variation, soit l'écart entre les coefficients de deux fonctions ϕ^j adjacentes. Le signe du coefficient, quant à lui, exprime le type de la variation : soit montante ou descendante pour le cas des images 1D. Si on laisse tomber un certain nombre de coefficients, il nous est possible d'obtenir une bonne approximation de l'image dans la mesure où les variations rejetées représentent de fins détails par rapport aux variations gardées. Seule la valeur absolue d'un coefficient d'ondelette doit être prise en compte lors de la compression.

La méthode de compression consiste à réécrire l'image décomposée (1.26) sous la forme

$$I(x) = c_0^0 \phi_0^0 + \sum_{i=1}^{2^J-1} \alpha_i u_i, \quad (1.43)$$

où les u_i sont les différentes fonctions d'ondelettes et les α_i sont leurs coefficients associés. On réécrit (1.43) en sommant par ordre décroissant de l'amplitude des coefficients. En représentant une permutation de coefficients par $\sigma(i)$, on obtient

$$I(x) = c_0^0 \phi_0^0 + \sum_{i=1}^{2^J-1} \alpha_{\sigma(i)} u_{\sigma(i)}, \quad (1.44)$$

où

$$|\alpha_{\sigma(i)}| \geq |\alpha_{\sigma(j)}| \quad \text{pour} \quad 0 < i < j. \quad (1.45)$$

En ne gardant que les m plus grands coefficients, on obtient une approximation $\hat{I}(x)$ de l'image $I(x)$

$$\hat{I}(x) = c_0^0 \tilde{\phi}_0^0 + \sum_{i=1}^{m-1} \alpha_{\sigma(i)} u_{\sigma(i)}(x) \quad (1.46)$$

Cette approximation introduit une erreur (dite absolue en moyenne quadratique) par rapport à l'image discrétisée originale que l'on calcule avec $\| u \|_2 = \langle u, u \rangle^{\frac{1}{2}}$, soit

$$\text{erreur} = \| I(x) - \hat{I}(x) \|_2 = \left[\sum_{i=m}^{2^J-1} (\alpha_{\sigma(i)})^2 \right]^{\frac{1}{2}}. \quad (1.47)$$

Puisque les images ont toutes une énergie différente, l'erreur absolue est difficile à interpréter car elle n'est pas un pourcentage de cette énergie. En passant à l'erreur relative définie par

$$\text{erreur relative} = \frac{\text{erreur}}{\| I(x) \|_2}, \quad (1.48)$$

on définit un taux d'erreur qui nous permet de comparer les erreurs en moyenne quadratique des différentes images compressées. En pratique, on veut compresser des images en tolérant un certain taux d'erreur. En multipliant ce taux par $\| I(x) \|_2$, on détermine une valeur d'erreur ε que nous approcherons sans la dépasser avec la formule suivante

$$\sum_{i=m}^{2^J-1} (\alpha_{\sigma(i)})^2 \leq \varepsilon^2 < \sum_{i=m-1}^{2^J-1} (\alpha_{\sigma(i)})^2. \quad (1.49)$$

Les Figures 1.7 et 1.8 nous montrent l'effet de la dimension de l'image sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative. La Figure 1.7 est obtenue grâce à des variations successives de la dimension d'une image 1D obtenues par différentes dilatations de cette dernière. Pour obtenir la Figure 1.8 nous avons fait les mêmes traitements sur la même image 1D perturbée par un bruit gaussien $b \sim N(\mu_b, \sigma_b)$ de moyenne $\mu_b = 0$ et d'écart-type $\sigma_b = 2 * 10^{-3}$. La Figure 1.9 représente l'effet du bruit gaussien sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative. Dans le cas des images 1D perturbées par un bruit gaussien, on peut utiliser la compression pour lisser l'image. Puisqu'un bruit gaussien consiste en de petites variations qui sont rejetées (mises à zéro) avec un taux de compression assez élevé, la courbe bruitée est approchée par une courbe plus lisse. Les Figures 1.10 à 1.15 représentent des boîtes de dialogue générées par Visual C++ 6.0, donnant des exemples de compressions d'une image 1D de 2^8 pixels (coefficients) perturbée par un bruit blanc gaussien

$b \sim N(\mu_b, \sigma_b)$ de moyenne $\mu_b = 0$ et d'écart-type $\sigma_b = 3 * 10^{-3}$, avec différentes erreurs relatives et les nombres de coefficients gardés. Chaque boîte de dialogue est sous forme d'une interface contenant quatre histogrammes représentant respectivement l'image 1D originale bruitée qui est un mixage de gaussiennes [1], l'image 1D décomposée en ondelettes de Haar, l'image 1D décomposée et compressée avec le nombre de coefficients gardés et l'erreur relative associée et, finalement, l'image originale bruitée approchée par une image reconstruite plus lisse.

Voici la procédure de compression d'une image 1D avec m le nombre de coefficients gardés et ϵ l'erreur absolue en moyenne quadratique [3].

Procédure Compression(c : tableau $[1..m]$ de réels, ϵ : réel)

$\tau_{min} \leftarrow \min\{|c[i]|\}$

$\tau_{max} \leftarrow \max\{|c[i]|\}$

faire

$\tau \leftarrow (\tau_{min} + \tau_{max})/2$

$s \leftarrow 0$

pour $i \leftarrow 0$ **jusqu'à** m **faire**

si $|c[i]| < \tau$ **alors** $s \leftarrow s + |c[i]|^2$

fin pour

si $s < \epsilon^2$ **alors** $\tau_{min} \leftarrow \tau$ **sinon** $\tau_{max} \leftarrow \tau$

jusqu'à $\tau_{min} \approx \tau_{max}$

pour $i \leftarrow 1$ **jusqu'à** m **faire**

si $|c[i]| < \tau$ **alors** $c[i] \leftarrow 0$

fin pour

fin Procédure

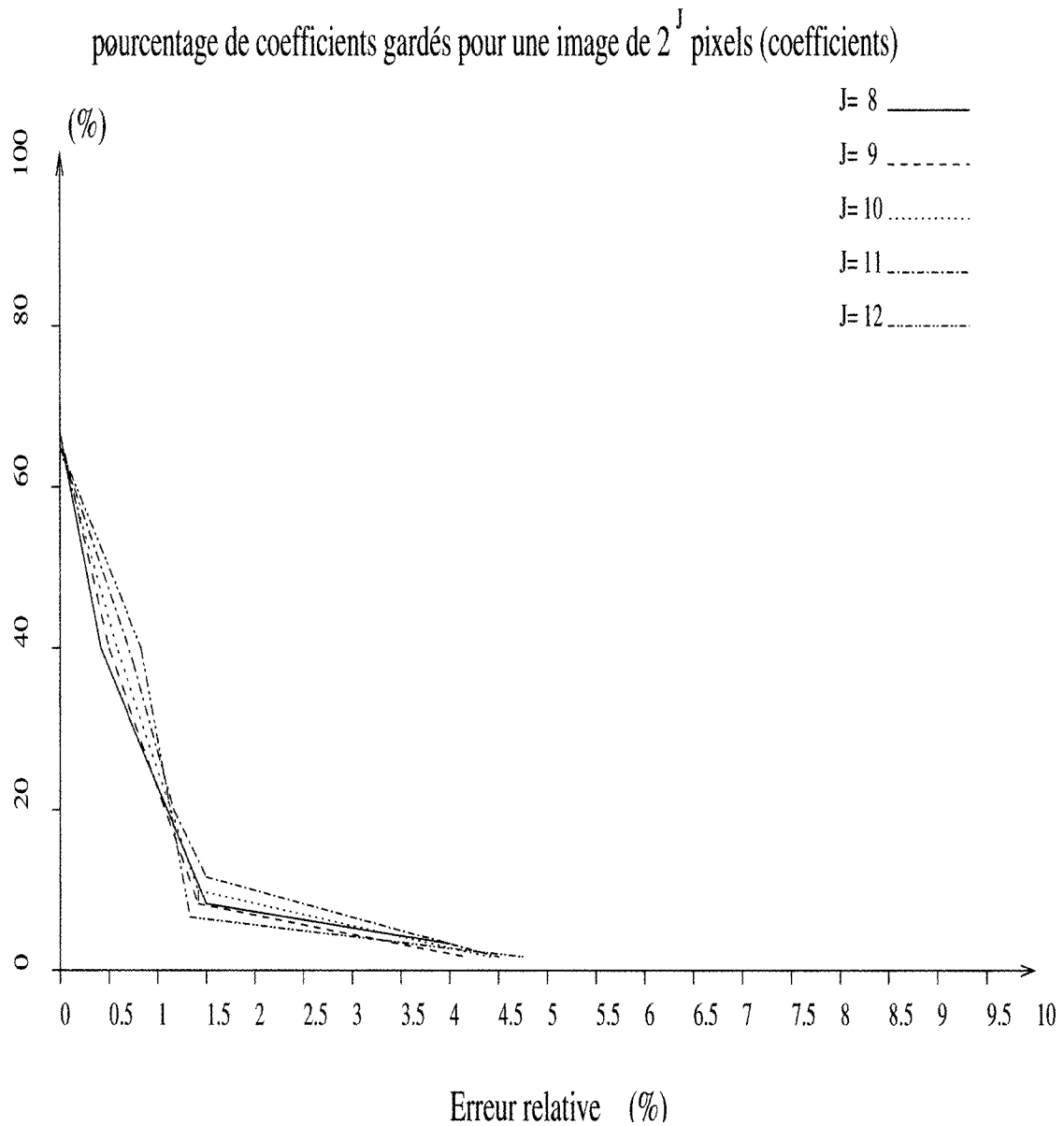


Figure 1.7 – Effet de la dimension de l'image sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.

pourcentage de coefficients gardés pour une image bruitée de 2^J pixels (coefficients)

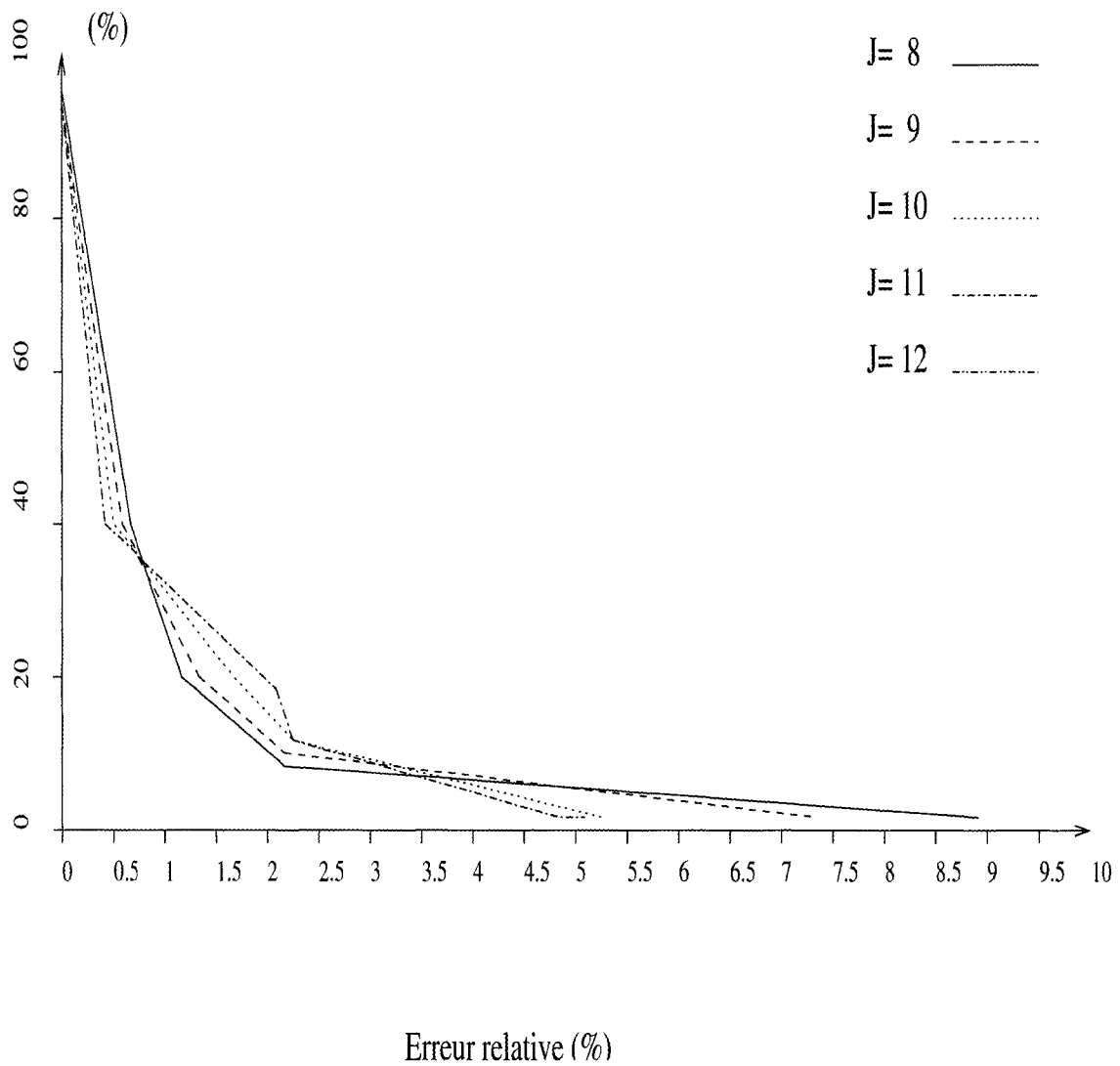


Figure 1.8 – Effet de la dimension de l'image bruitée sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.

Pourcentage de coefficients gardés pour une image de 2^J pixels (coefficients)
 perturbée par un bruit gaussien $N(0, \sigma_b)$

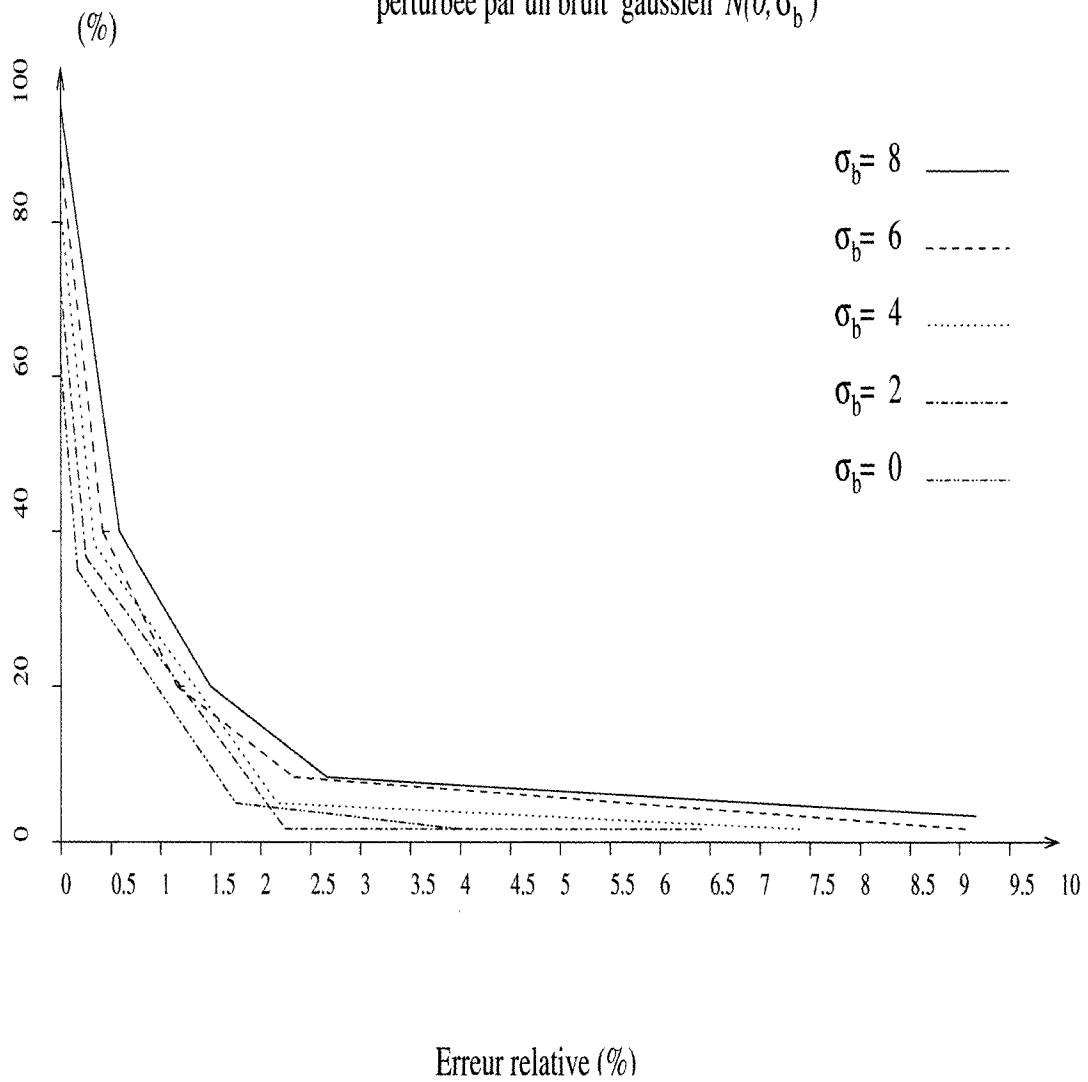


Figure 1.9 – Effet du bruit gaussien sur le nombre de coefficients gardés pour une compression permettant $x\%$ d'erreur relative.

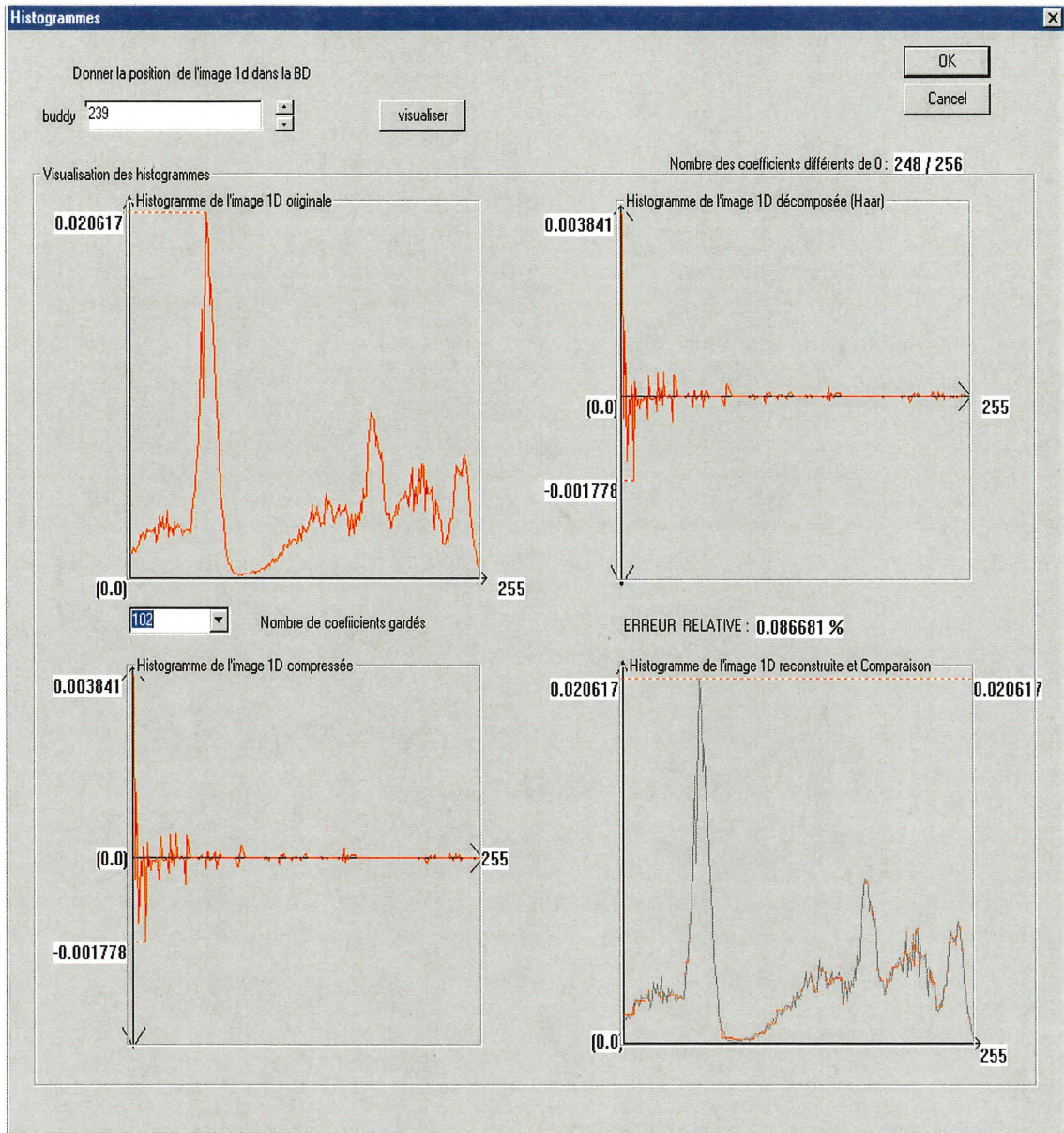


Figure 1.10 – Compression 1D: Erreur relative 0.086% et nombre de coefficients gardés 102.

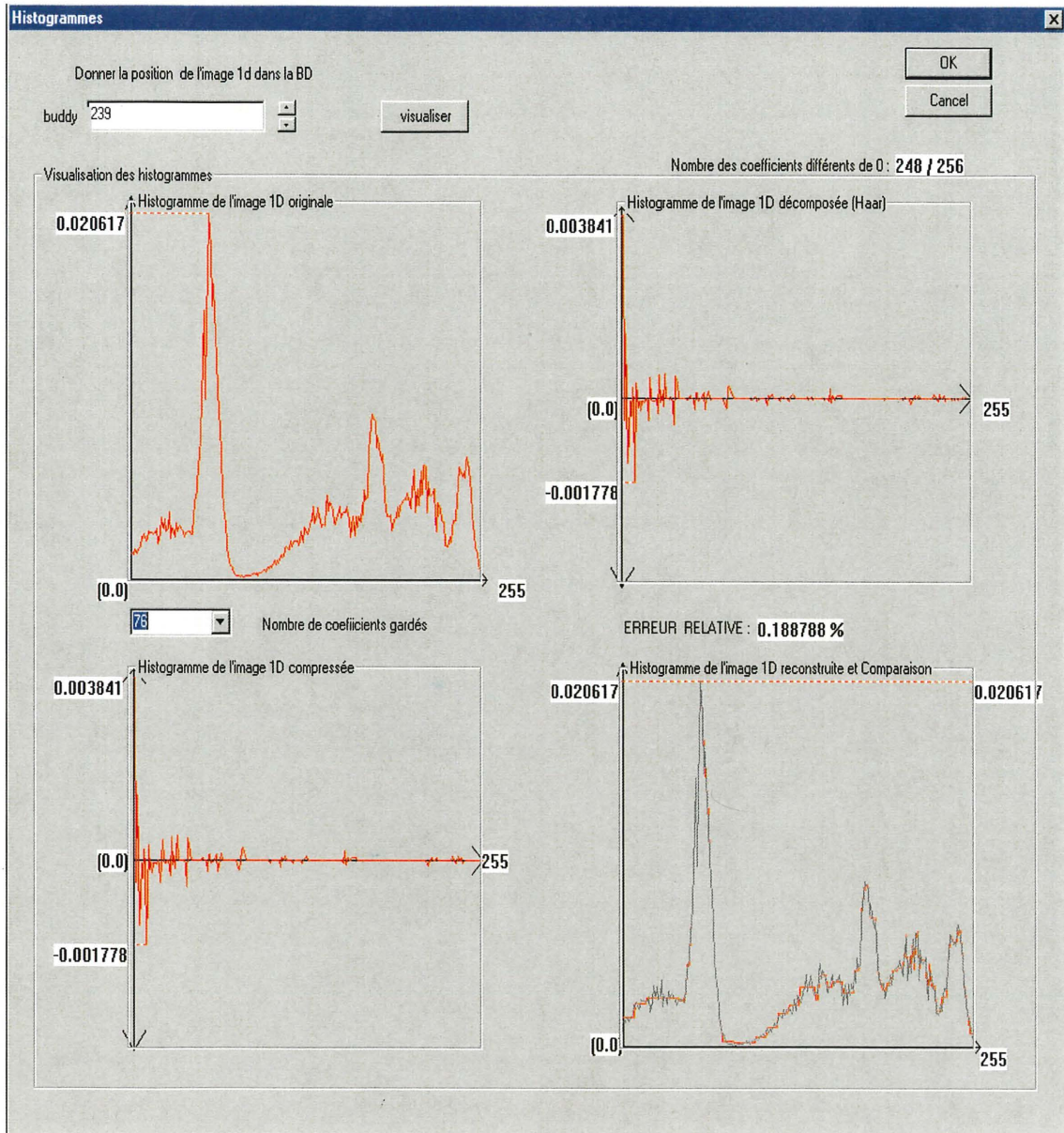


Figure 1.11 – Compression 1D: Erreur relative 0.18% et nombre de coefficients gardés 76

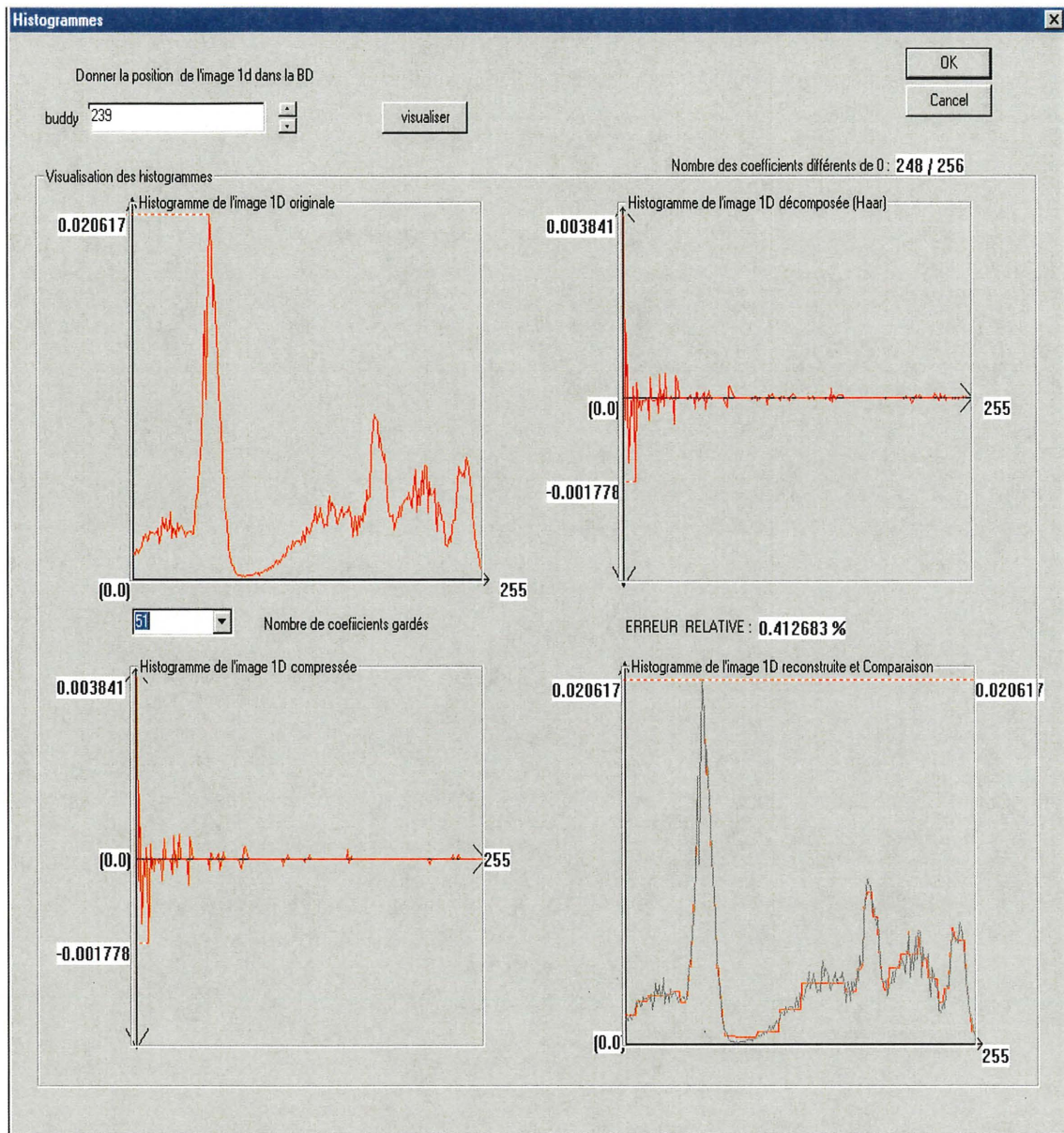


Figure 1.12 – Compression 1D: Erreur relative 0.41% et nombre de coefficients gardés 51

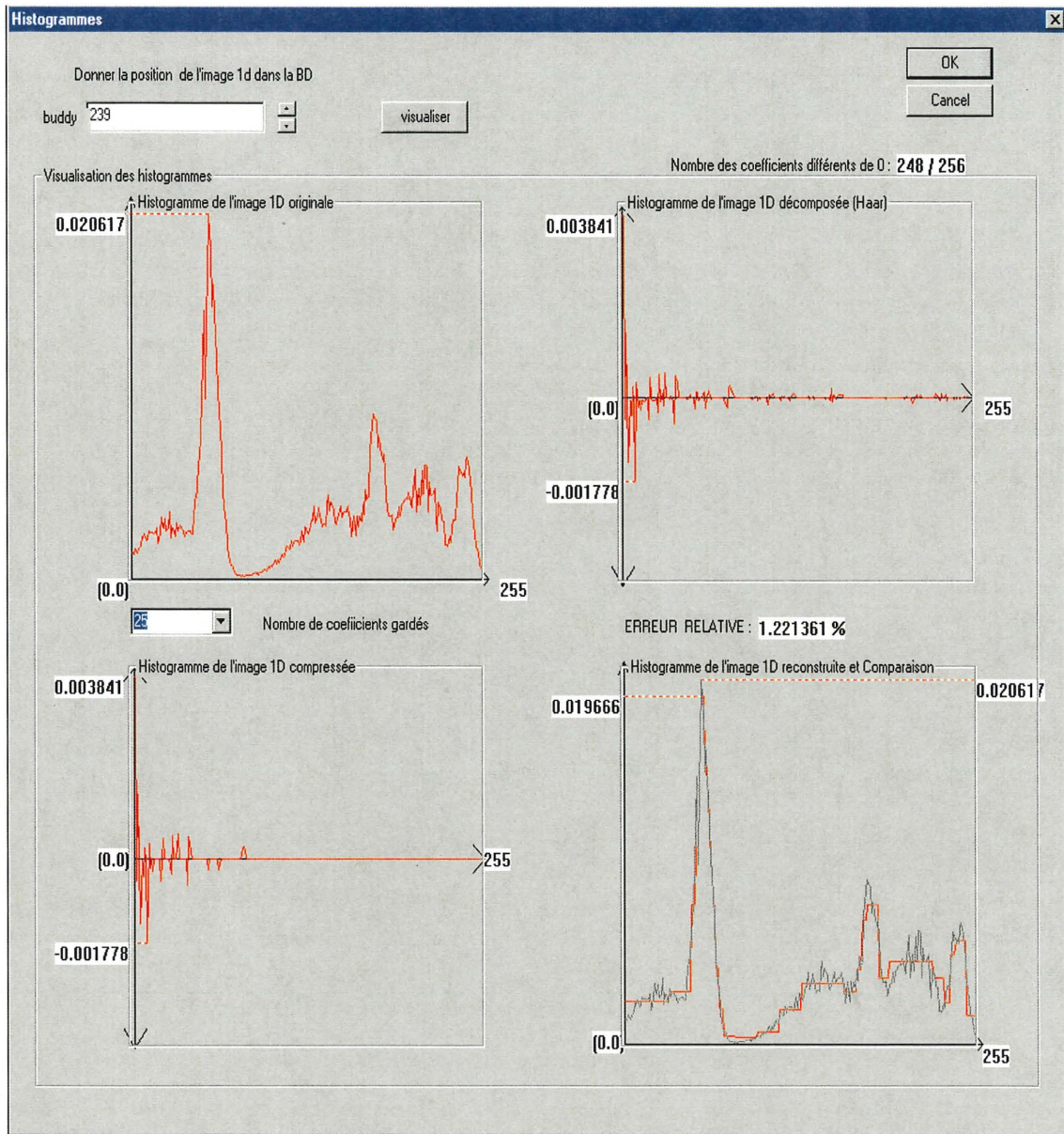


Figure 1.13 – Compression 1D: Erreur relative 1.22% et nombre de coefficients gardés 25

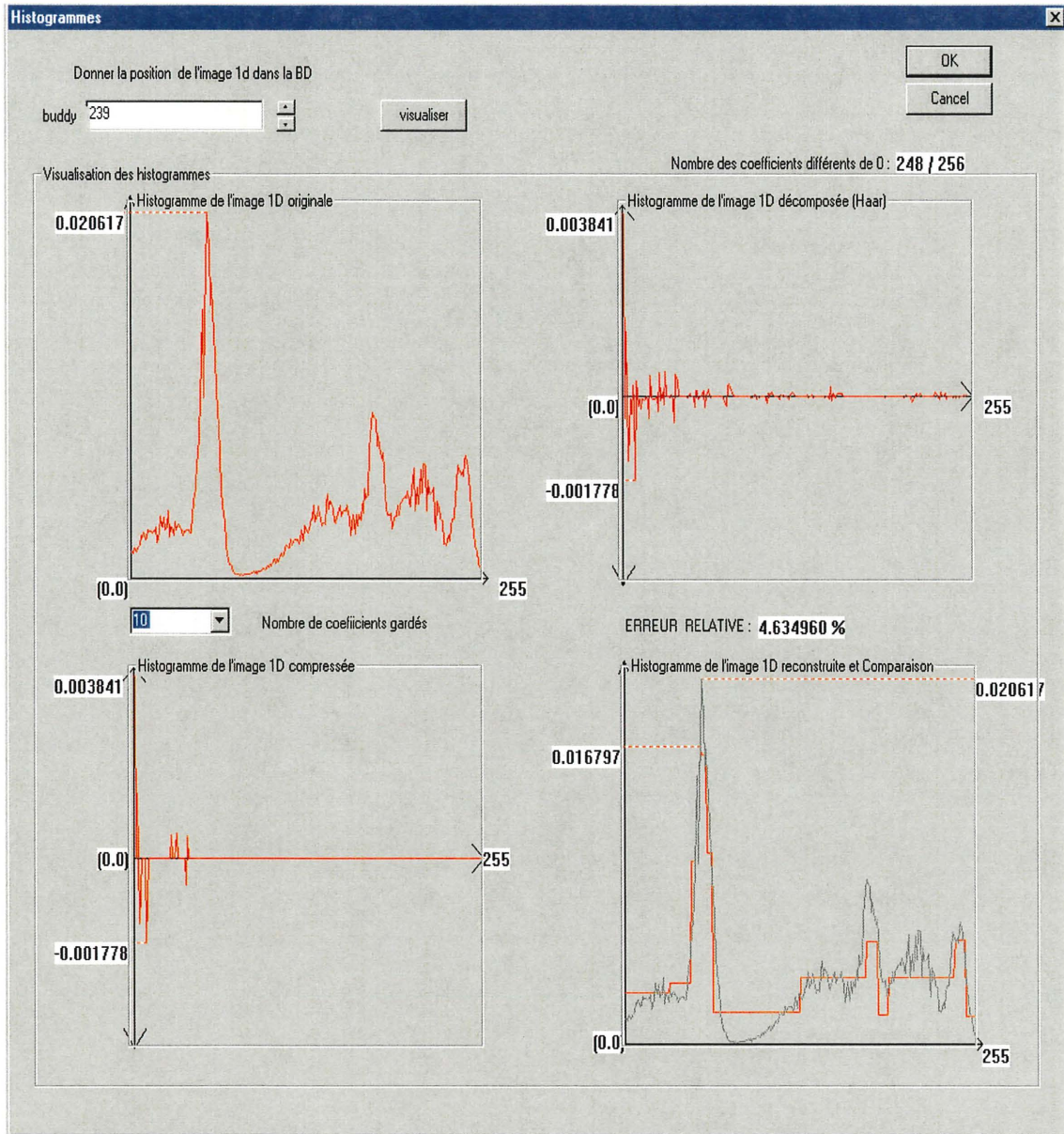


Figure 1.14 – Compression 1D: Erreur relative 4.63% et nombre de coefficients gardés 10

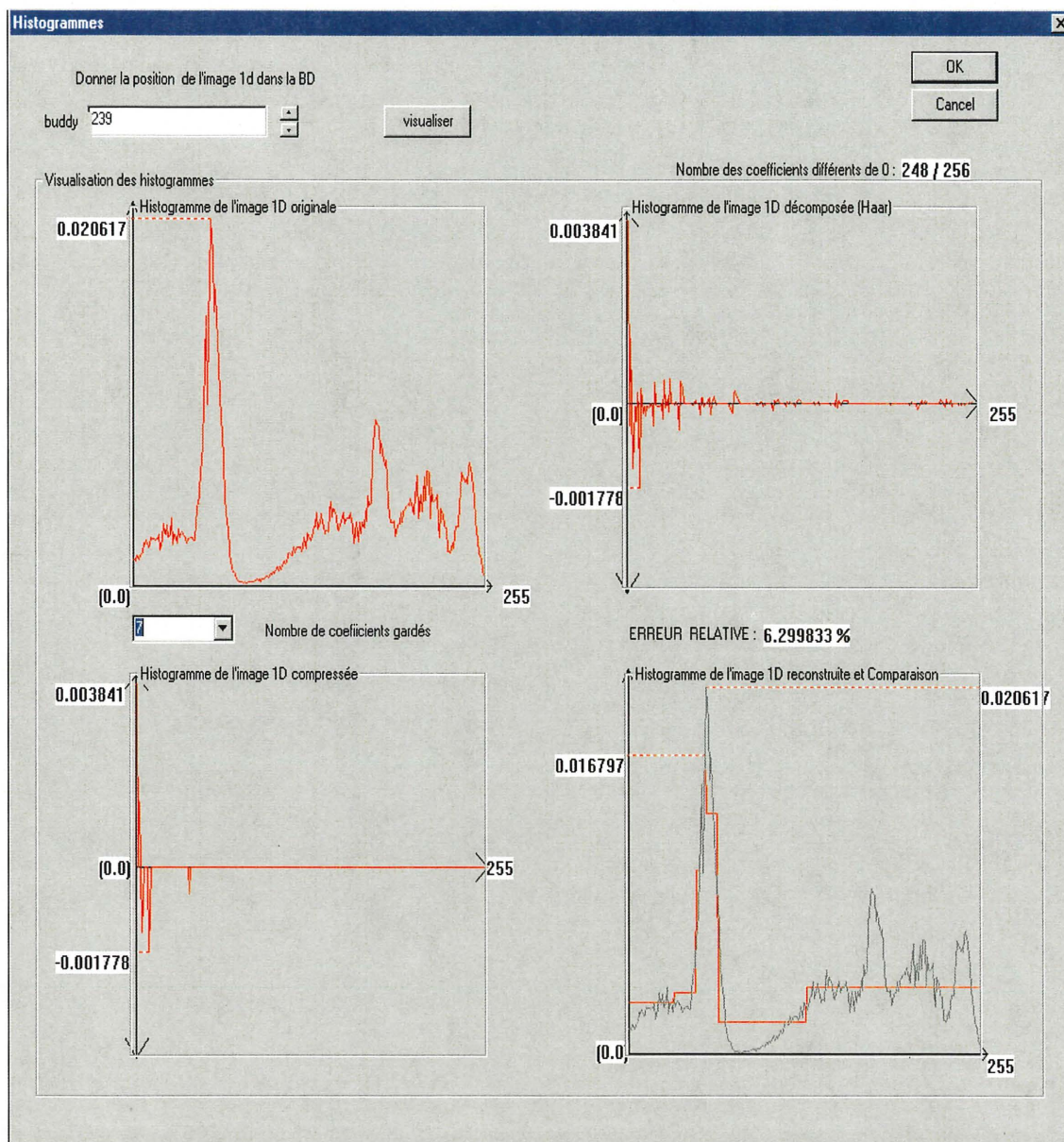


Figure 1.15 – Compression 1D: Erreur relative 6.29% et nombre de coefficients gardés 7.

CHAPITRE 2

Recherche d'images 1D

2.1 Recherche d'images 1D

Dans ce chapitre, nous voulons établir un processus automatisé de recherche d'images par contenu qui s'applique à de grosses BD. Nous choisissons pour cela une "distance" suggérée par [2], qui sera calculée d'une façon rapide entre la requête Q et les images T de la BD. Pour chaque image de la BD, le calcul de cette "distance" représentera sa proximité perceptuelle à la requête. Nous déduisons cette proximité de la ressemblance des allures des histogrammes représentant respectivement la requête et la cible. Afin d'obtenir le résultat optimal de la recherche, la "distance" nous permettra d'ordonner les cibles de BD selon leurs ressemblances par rapport à la requête.

2.2 Construction de la métrique

2.2.1 Motivation

Les métriques d'images 1D les plus souvent utilisées sont celles induites par les normes L^1 et L^2 définies par

$$\|Q, T\|_1 = \sum_i |Q[i] - T[i]|, \quad (2.1)$$

et

$$\|Q, T\|_2 = \left(\sum_i (Q[i] - T[i])^2 \right)^{\frac{1}{2}}, \quad (2.2)$$

où

$$Q(x) = \sum_{i=0}^{2^J-1} Q[i] \phi_i^J(x), \quad (2.3)$$

et

$$T(x) = \sum_{i=0}^{2^J-1} T[i] \phi_i^J(x). \quad (2.4)$$

Cependant, ces métriques se calculent sur la totalité des pixels de l'image requête Q et l'image cible T , ce qui peut rendre la recherche coûteuse en temps de calcul surtout lorsque l'on s'attaque à de grandes BD. Pour remédier à cela, nous utilisons une métrique suggérée par [2], qui calcule une "distance" sur les signatures des images de la BD.

La décomposition en ondelettes de Haar représente l'image 1D par la moyenne de ses niveaux de gris (facteur d'échelle) et les coefficients associés aux fonctions d'ondelettes ψ_i^j , où $j \in \{0, 1, \dots, J\}$ et $i \in \{0, 1, \dots, 2^j - 1\}$. La section du chapitre 1 sur la compression nous a montré que nous pouvions obtenir de bonnes approximations de l'image décomposée avec peu de coefficients. La compression des images (à m coefficients) met en relief

les variations caractéristiques de l'image 1D en conservant les valeurs et les positions des coefficients importants. En gardant en plus le facteur d'échelle, nous tenons compte de l'aspect de luminosité globale dans l'image originale.

Les m coefficients qui ont été gardés après la compression, sont ceux qui ont les plus grandes amplitudes dans l'image décomposée. D'après [2], en prenant en compte seulement des signes de ces coefficients dans la recherche, nous augmenterons la rapidité de l'algorithme de la comparaison. Néanmoins, le stockage de ces derniers nécessite un grand espace. Par conséquent, nous effectuons une quantification de nos images qui a l'avantage de diminuer l'espace nécessaire pour stocker ces dernières au détriment de la capacité discriminatoire de la métrique. La quantification consiste à ne garder que le signe des m plus grands coefficients. Ce signe représente le type de variation (ou contour) de la fonction ψ . Chaque coefficient significativement non nul est quantifié seulement à deux niveaux, $+1$ représente les plus larges coefficients positifs et -1 représente les plus larges coefficients négatifs.

2.2.2 Métrique

La métrique proposée par [2] et que nous utilisons ici est obtenue à partir de l'expression suivante

$$\| Q, T \| = w_0 |Q[0] - T[0]| + \sum_{i=1}^{2^j-1} w_i |\tilde{Q}[i] - \tilde{T}[i]|, \quad (2.5)$$

où les w_i sont des poids non-négatifs, $Q[0]$ et $T[0]$ sont respectivement les facteurs d'échelle des images Q et T et $\tilde{Q}[i]$ et $\tilde{T}[i]$ sont les coefficients décomposés, compressés et quantifiés de ces dernières. Une fois les images quantifiées, le terme

$$|\tilde{Q}[i] - \tilde{T}[i]| \in \{0, 1, 2\} \quad (2.6)$$

puisque la valeur possible d'un coefficient après la quantification est -1, 0 ou 1. Dans le cas où

$$|\tilde{Q}[i] - \tilde{T}[i]| = 2. \quad (2.7)$$

Q et T ont conservé leurs coefficients à la position i après la compression, mais les signes des deux coefficients (le type de contour) ne correspondent pas. Ce sont des variations de signes contraires. Du point de vue de la distance entre deux images Q et T , deux variations de signes contraires ayant les mêmes positions ne constituent pas plus une proximité perceptuelle que lorsque l'une d'elles est mise à zéro après la compression. Seule la distance entre une requête et des cibles où il y a au moins une correspondance parfaite pour un coefficient donné sera calculée. Nous rejetons en fait l'hypothèse qu'une image T soit la cible désirée si aucun de ses coefficients gardés ne correspond à ceux de la requête. La distance entre les images de la BD et la requête ne prend en considération que les coefficients non nuls de cette dernière, soient les m coefficients gardés. Par conséquent, notre métrique devient asymétrique (rigoureusement elle ne pourrait garder le titre de métrique, cependant, nous continuerons tout de même à l'appeler ainsi pour des raisons pratiques). Nous préférons cette approche car il y a beaucoup plus de non-correspondances que de correspondances pour une recherche donnée. C'est pour cela que nous réécrivons notre métrique comme suit

$$\|Q, T\| = w_0|Q[0] - T[0]| + \sum_{i:\tilde{Q}(i)\neq 0} w_i \left(\tilde{Q}[i] \neq \tilde{T}[i] \right), \quad (2.8)$$

avec

$$\left(\tilde{Q}[i] \neq \tilde{T}[i] \right) = \begin{cases} 1 & \text{si } \tilde{Q}[i] \neq \tilde{T}[i], \\ 0 & \text{sinon.} \end{cases} \quad (2.9)$$

Nous pouvons simplifier le calcul de notre métrique en prenant en considération les correspondances plutôt que les non-correspondances. Pour cette raison, on réécrit

$$\sum_{i:\tilde{Q}(i)\neq 0} w_i \left(\tilde{Q}[i] \neq \tilde{T}[i] \right) = \sum_{i:\tilde{Q}(i)\neq 0} w_i - \sum_{i:\tilde{Q}(i)\neq 0} w_i \left(\tilde{Q}[i] = \tilde{T}[i] \right) \quad (2.10)$$

avec

$$\left(\tilde{Q}[i] = \tilde{T}[i] \right) = \begin{cases} 1 & \text{si } \tilde{Q}[i] = \tilde{T}[i], \\ 0 & \text{sinon.} \end{cases} \quad (2.11)$$

Notre métrique devient alors

$$\| Q, T \| = w_0 |Q[0] - T[0]| + \sum_{i:\tilde{Q}(i) \neq 0} w_i - \sum_{i:\tilde{Q}(i) \neq 0} w_i \left(\tilde{Q}[i] = \tilde{T}[i] \right). \quad (2.12)$$

Puisque le terme qui somme seulement les w_i est constant pour toutes les images de la BD pour une requête donnée, nous pouvons le négliger car tout ce qui nous intéresse est d'ordonner les images de la BD selon leurs distances calculées par rapport à la requête. De plus, puisqu'il ne peut y avoir correspondance qu'au niveau des m coefficients gardés de la requête, nous ne calculerons la distance que pour les positions de ces derniers. Ainsi, nous obtenons

$$\| Q, T \| = w_0 |Q[0] - T[0]| - \sum_{i:\tilde{Q}(i) \neq 0} w_i \left(\tilde{Q}[i] = \tilde{T}[i] \right). \quad (2.13)$$

Le poids w_i donne une importance relative à la position i . Pour simplifier, on suppose que les poids associés à des coefficients appartenant à un même niveau de résolution sont identiques. Ainsi, on regroupe les w_i par niveaux de résolution en utilisant la fonction $bin()$ qui représente le plus grand entier inférieur ou égal à $\log_2(i)$ pour $i \in \{1, \dots, 2^J - 1\}$

$$bin(i) = \lfloor \log_2(i) \rfloor \quad \text{avec} \quad i = 1, \dots, 2^J - 1, \quad (2.14)$$

où J est le nombre maximum de niveaux de résolution puisque nos images ont 2^J pixels. Ainsi, on obtient

$$\| Q, T \| = \tilde{w}_0 |Q[0] - T[0]| - \sum_{i:\tilde{Q}(i) \neq 0} w_{bin(i)} \left(\tilde{Q}[i] = \tilde{T}[i] \right). \quad (2.15)$$

2.2.3 Tableaux de recherche

Afin d'optimiser le processus de calcul de la métrique, on considère deux tableaux de listes d'occurrence proposés par [2], soient Θ_+ pour les coefficients quantifiés à +1 et Θ_-

pour ceux quantifiés à -1 . Chacun de ces deux tableaux a $2^J - 1$ cellules et chacune de leurs cellules contient une liste. La liste de la cellule i du tableau Θ_- pointe sur toutes les images de la BD qui ont un coefficient -1 gardé dans la position i après la compression et la quantification. De même la liste de la cellule i du tableau Θ_+ pointe sur toutes les images de la BD qui ont un coefficient $+1$ gardé à la position i après la compression et la quantification. La Figure 2.1 nous donne un exemple de l'ajout d'une image n aux tableaux Θ_- et Θ_+ . Ainsi, lors d'une recherche, nous n'avons besoin de parcourir que les m listes associées aux m coefficients de la requête \tilde{Q} au lieu des $2^J - 1$ coefficients. Cela a pour effet d'accélérer la recherche.

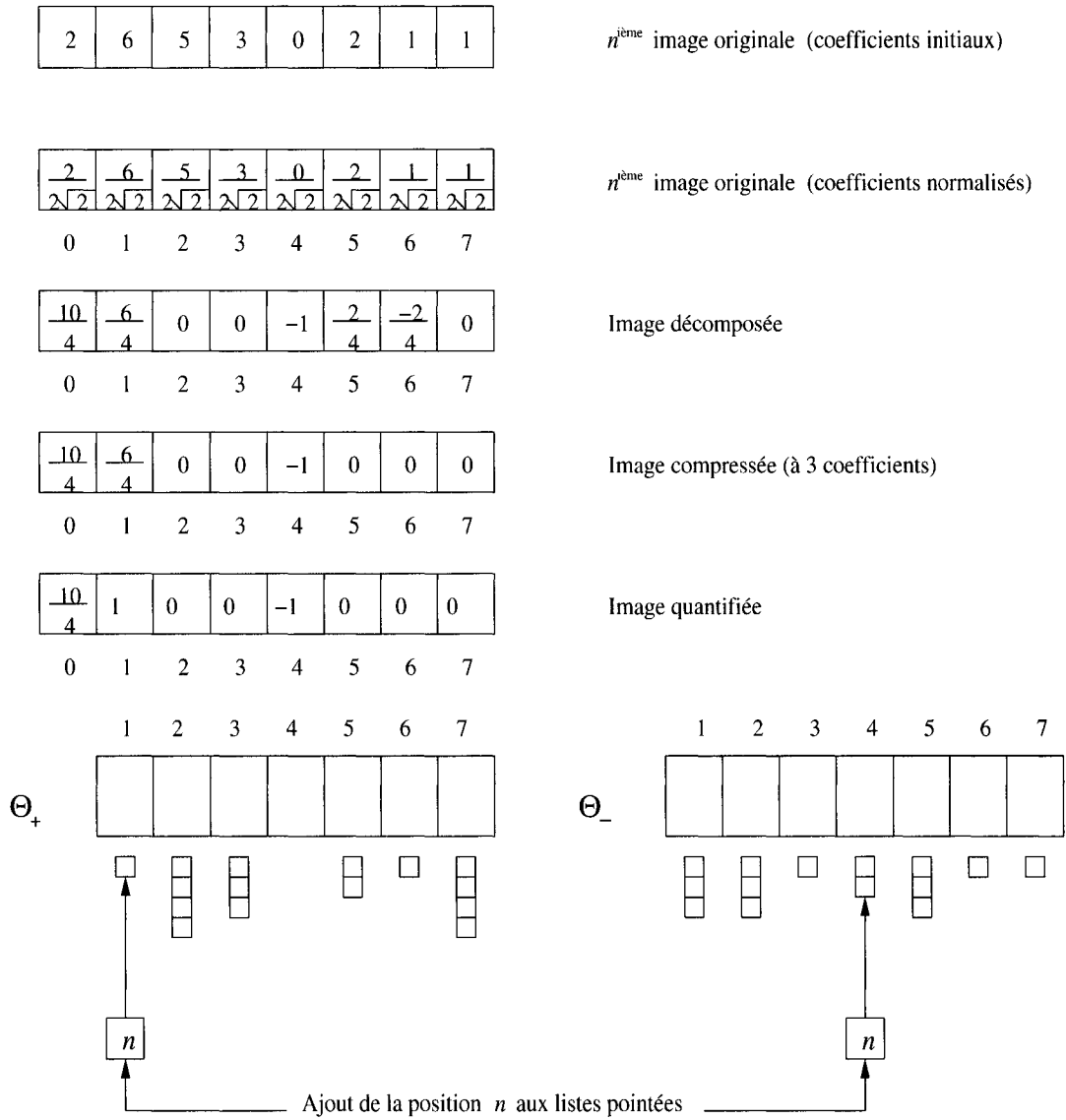


Figure 2.1 – Ajout de la $n^{\text{ème}}$ image aux tableaux Θ_+ et Θ_- .

2.2.4 Organigramme de la recherche

La Figure 2.2 montre l'organigramme de la recherche d'une image requête Q dans une BD d'images T .

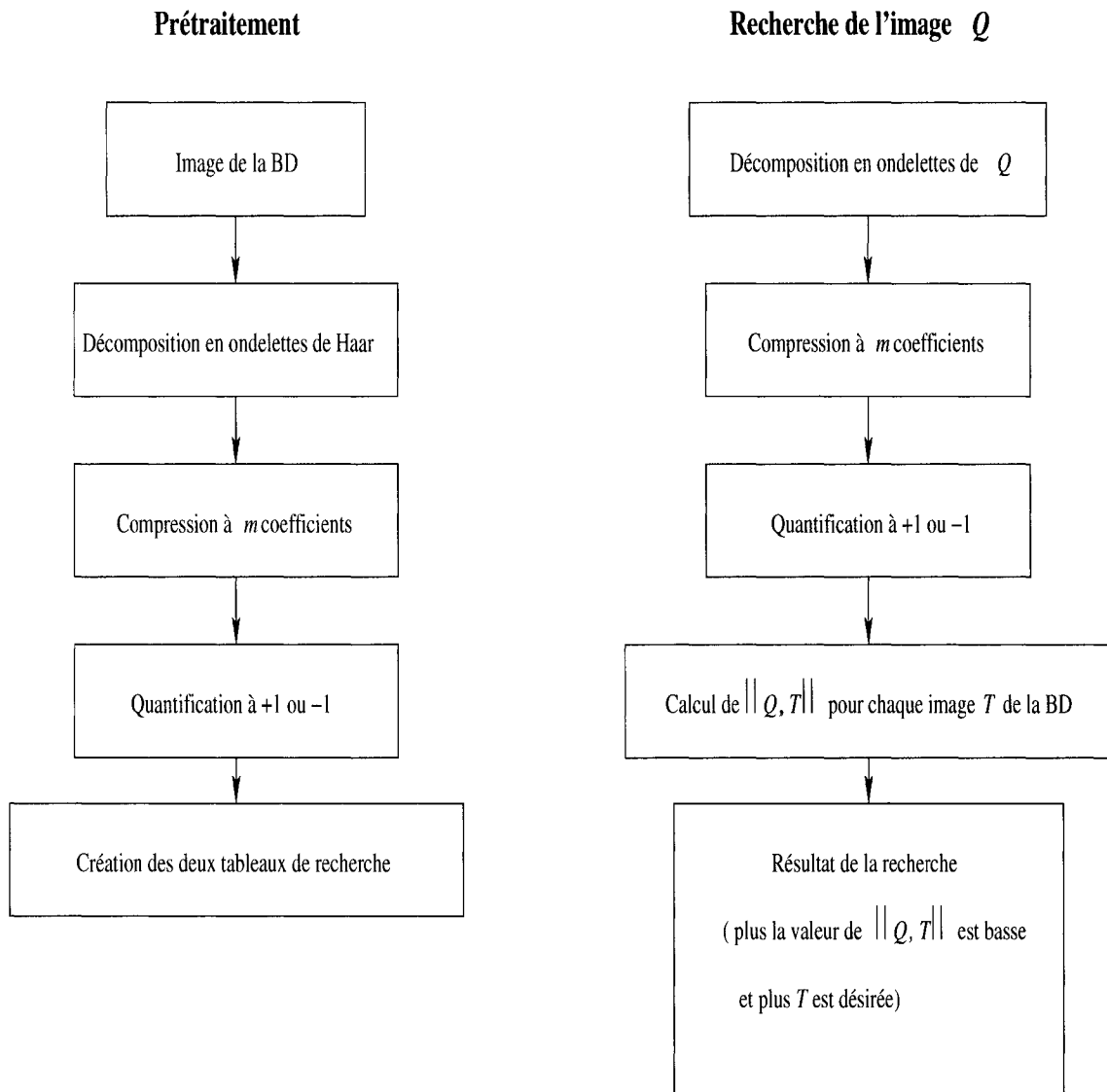


Figure 2.2 – Organigramme de la recherche.

2.3 Calcul des poids

2.3.1 Calcul des poids

Afin de calculer notre métrique, nous devons déterminer les poids w_k qui représentent le degré de correspondance qu'a un coefficient appartenant au niveau de résolution k , gardé après la compression à m coefficients et la quantification, à un autre élément de la requête appartenant au même niveau de résolution, ayant la même position et le même signe (même type de contour). La "distance" calculée à l'aide de ces poids nous permet d'ordonner et de mieux discriminer les cibles entre elles, puisqu'elle représente la proximité perceptuelle à la requête.

L'information que nous désirons obtenir, une fois que nos distances sont calculées pour chacune des cibles de la BD, c'est de déterminer si une cible T est l'image recherchée ou non. Comme toutes ces distances seront relatives, nous pourrons les ordonner par ordre décroissant et déterminer l'image désirée.

D'après [11], en assumant un modèle de régression logistique et en effectuant un ajustement de moindres carrés aux données de notre base, nous pouvons arriver à déterminer un ensemble de poids w_k qui optimiseront la probabilité qu'une image T soit la cible désirée. En [2], il est proposé d'utiliser le modèle de régression logistique suivant

$$v_l^* = v + \tilde{w}_0 \tilde{t}_{0,0} + \sum_{k=0}^{J-1} w_k t_{k,l} + u_l \quad (2.16)$$

où les différentes variables sont

v_l^* : proximité perceptuelle de la cible par rapport à la requête,

v : constante indépendante du couple requête-cible,

- w_k : poids des niveaux de résolutions k ,
- \tilde{w}_0 : poids multiplié par la valeur absolue de la différence entre les facteurs d'échelle du couple requête-cible l ,
- $t_{k,l}$: nombre de coefficients appartenant au niveau de résolution k , où il n'y a pas de correspondance entre l'image requête et l'image cible l ,
- $\tilde{t}_{0,0}$: différence en valeur absolue entre les facteurs d'échelle du couple requête-cible l ,
- u_l : terme d'erreur.

Puisque v_l^* est une variable continue qu'on ne peut pas interpréter, nous introduisons une nouvelle variable de sortie binaire v_l . Pour pouvoir construire les échantillons nécessaires au calcul des poids, nous observerons alternativement la variable binaire v_l qui représente la correspondance ou la non-correspondance entre deux images comparées.

2.3.2 Échantillonnage

Nous construisons deux échantillons, un pour les correspondances entre l'image cible et la requête ($v_l = 0$) et un pour celles qui ne correspondent pas ($v_l = 1$). Le premier est constitué de 85 images tirées aléatoirement de notre BD. Pour chacune d'elles, nous générons une version bruitée de l'image (bruit blanc gaussien d'écart-type $2 * 10^{-3}$). Nous voulons que les poids soient fixés de telle sorte qu'une recherche de l'image bruitée trouvera l'image originale et c'est ce qui impose la valeur 0 à la variable v_l^* . Nous calculons les valeurs de $t_{k,l}$ pour cette paire d'images. nous faisons de même pour les 84 autres images du premier échantillon. Nous pigeons ensuite 8500 paires d'images de la BD aléatoirement et nous fixons pour ce deuxième échantillon la valeur de v_l^* à 1. Comme pour le premier échantillon, nous calculons pour chaque paire d'images les valeurs de $t_{k,l}$ et $\tilde{t}_{0,0}$ [2]. Ces échantillons d'images nous fournissent des données (les valeurs de $t_{k,l}$ et $\tilde{t}_{0,0}$) qui seront des variables indépendantes de notre modèle de régression logistique. De

ces données, ainsi que des valeurs de v_l posées, nous pourrons obtenir nos poids.

2.3.3 Régression logistique

Ce type de régression est effectué par de nombreux logiciels. Nous avons utilisé la fonction Logit de SAS 6.2 pour effectuer la nôtre.

La régression logistique permet de définir les poids w_k et \tilde{w}_0 afin qu'ils maximisent la probabilité qu'un calcul de v_l^* , sachant $t_{k,l}$ et $\tilde{t}_{0,0}$, produise la sortie v_l ajoutée à l'échantillon destiné à SAS. Une régression logistique s'applique dans les cas où la variable de sortie est binaire. Cependant, la variable latente v_l^* est difficile à observer et à mesurer directement, car elle représente une variable continue qui définit la proximité perceptuelle d'une cible l par rapport à la requête. Par conséquent, nous la remplaçons par une variable temporaire définie par

$$v_l = \begin{cases} 1 & \text{si } v_l^* > 0 \text{ (non-correspondance),} \\ 0 & \text{sinon (correspondance parfaite),} \end{cases} \quad (2.17)$$

pour la construction de nos échantillons.

La procédure "LOGISTIC" de SAS 6.2 affecte l'évènement d'intérêt à la variable $Y = 0$ par défaut. En construisant nos échantillons de telle sorte que les échantillons pour $v_l = 1$ apparaissent avant ceux pour $v_l = 0$, et en utilisant l'option $ORDER = DATA$ dans le code fourni à SAS, la fonction "LOGISTIC" modèle l'évènement d'intérêt pour $v_l = 1$. Nous modelons ainsi la probabilité que la cible l ne soit pas la cible désirée. Lorsque SAS modèle l'évènement d'intérêt, les poids sont plus faciles à interpréter. De plus, il est plus simple de générer les échantillons dans le cas où $v_l = 1$, car nous tirons au hasard 2 images de la BD et l'évènement d'intérêt doit avoir beaucoup plus d'échantillons. De plus, dans une base de données de 1000 images, chaque image requête aura 999 non-

correspondances et une seule correspondance. L'option *OUTEST = BETAS* stocke les poids fournis dans une variable *BETAS*.

Voici la procédure fournie à SAS 6.2 pour effectuer la régression logistique.

```

TITLE 'poids des résolutions';
DATA ingots;

      INPUT V  $\tilde{T}0$  T0 T1 T2 T3 T4 T5 T6 T7;

CARDS;
1 SF(1) ... Somme des non-correspondances pour les  $k$  autres niveaux
0 SF(1) ... Somme des non-correspondances pour les  $k$  autres niveaux
;
PROC LOGISTIC DATA = ingots OUTEST = BETAS ORDER = DATA;
MODEL V =  $\tilde{T}0$  T0 T1 T2 T3 T4 T5 T6 T7;
run;

```

Les valeurs de $t_{k,l}$ et $\tilde{t}_{0,0}$ sont les proportions des non-correspondances entre la requête et la cible pour un niveau k et pour les facteurs d'échelle du couple requête-cible. Puisque la totalité de nos poids w_k et \tilde{w}_0 sont positifs, plus les valeurs de $t_{k,l}$ et $\tilde{t}_{0,0}$ seront élevées et plus la probabilité que la cible ne soit pas celle désirée augmentera.

Ainsi codé, SAS suppose que

$$v_l = \begin{cases} 0 & \text{si } -\infty < v_l^* \leq 0, \\ 1 & \text{si } 0 < v_l^* < +\infty. \end{cases} \quad (2.18)$$

Le modèle *Logit* génère les poids w_k et \tilde{w}_0 qui pourront être utilisés pour le calcul de la probabilité qu'une valeur de v_l^* donnée puisse produire une sortie positive v_l . En effet, en

prenant en considération les hypothèses du modèle *logit*, la probabilité P_l qu'une cible l corresponde parfaitement avec la requête, est donnée par

$$P_l = Prob(v_l = 1 | t_l) = F(v + \tilde{w}_0 \tilde{t}_{0,0} + \sum_{k=0}^{J-1} w_k t_{k,l}) \quad (2.19)$$

et

$$\bar{P}_l = 1 - P_l = Prob(v_l = 0 | t_l) = F(-v - \tilde{w}_0 \tilde{t}_{0,0} - \sum_{k=0}^{J-1} w_k t_{k,l}) \quad (2.20)$$

où

$$F(x) = \frac{e^x}{1 + e^x}. \quad (2.21)$$

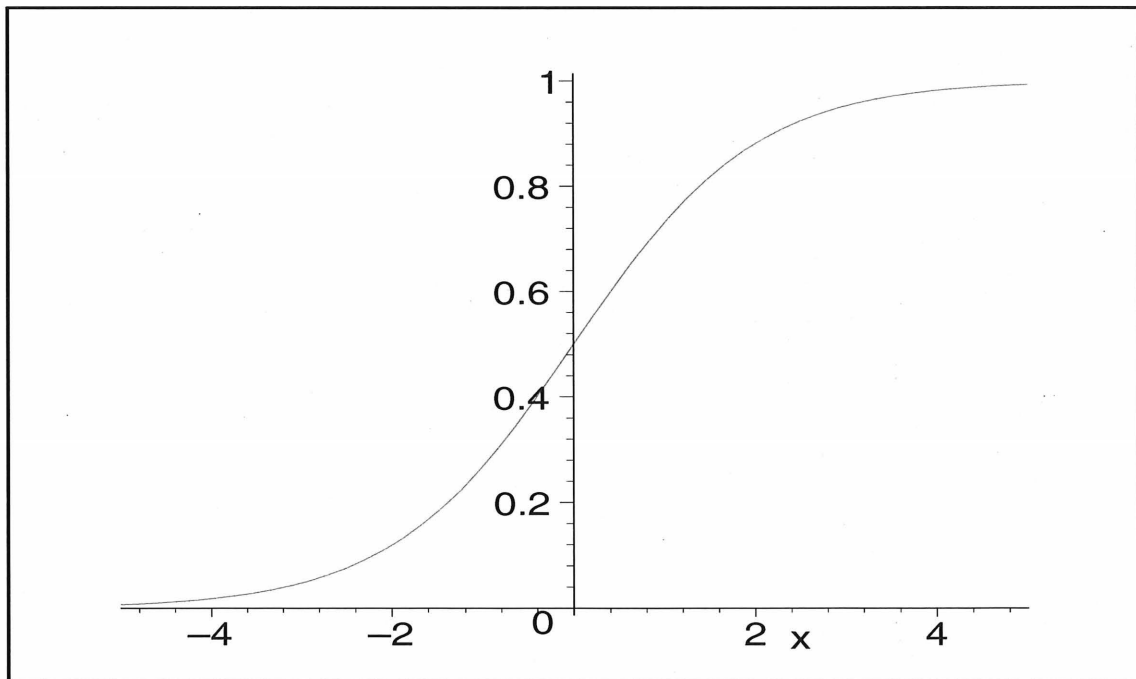


Figure 2.3 – Courbe de la fonction F de distribution de l'erreur logistique u_l .

La fonction F découle de la définition même de la fonction $Logit(P_l)$ donnée par

$$Logit(P_l) = \log\left(\frac{P_l}{1 - P_l}\right) = v + \tilde{w}_0 \tilde{t}_{0,0} + \sum_{k=0}^{J-1} w_k t_{k,l} \quad (2.22)$$

avec

$$\text{Logit}(x) = \log\left(\frac{x}{1-x}\right). \quad (2.23)$$

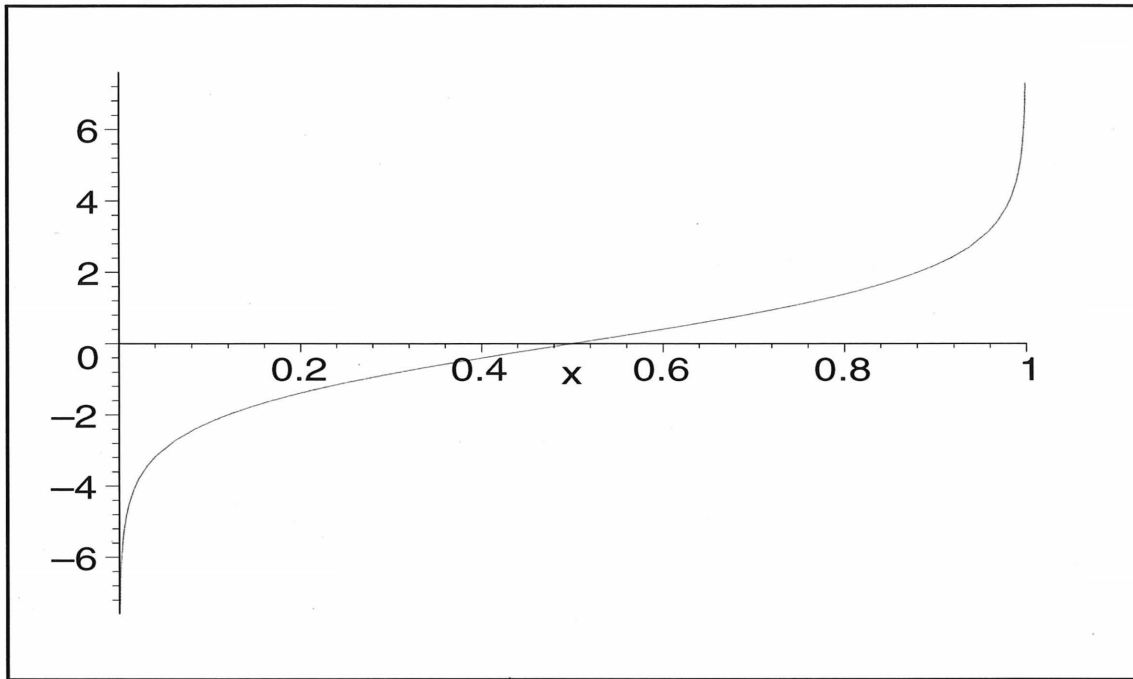


Figure 2.4 – Courbe de la Fonction Logit de distribution des probabilités P_l .

Puisque F est une fonction monotone (croissante) et v est constante pour tout couple requête-cible, on pourra calculer le terme

$$\tilde{w}_0 \tilde{t}_{0,0} + \sum_k w_k t_{k,l}$$

et ordonner les probabilités de similitude des images de la BD par rapport à la requête par ordre décroissant. On peut ainsi mesurer la "distance" de n'importe quelle requête par rapport aux images de la BD.

2.4 Méthode de recherche

Voici la procédure de la recherche d'une image requête Q dans une BD lorsque les poids w_k sont fournis.

Procédure Recherche(Q : tableau $[1..2^J]$ de réels, m : entier)

$Q^c \leftarrow \text{StandardDécomposition}(Q)$

Initialise Score[i] = 0, pour tout i

Pour chaque image T de la BD **faire**

Score[position de T dans BD] = $\tilde{w}_0 * \| Q^c[0] - T^c[0] \|$

Fin pour

$\tilde{Q}^c \leftarrow \text{ComprimeCoefficients}(Q^c)$

$\tilde{Q}_q^c \leftarrow \text{Quantifie}(\tilde{Q}^c)$

Pour chaque $\tilde{Q}_q^c \neq 0$ **faire**

si $\tilde{Q}_q^c > 0$ **alors**

Liste $\leftarrow \Theta_+[i]$

Sinon

Liste $\leftarrow \Theta_-[i]$

Fin si

pour chaque l de Liste faire

Score[position de l dans BD] = Score[position de l
dans BD] - $w_{bin(i)}$

Fin pour

Fin pour

Retourne Score

Fin procédure

Notre algorithme examine la liste des scores qui peuvent être positifs ou négatifs. Les images cibles ayant les plus bas scores (les plus négatifs), sont considérées comme étant les images les plus proches de l'image requête.

CHAPITRE 3

Base de données 1D

3.1 Base de données de mixtures de gaussiennes 1D

Dans la recherche d'images par contenu, [12] propose de décrire la distribution des couleurs de chaque image d'une base de données d'images couleurs, par une mixture de gaussiennes. Par conséquent, en appliquant notre méthode de recherche sur des mixtures de gaussiennes, il est possible aussi de l'appliquer sur une base de données d'images couleurs. Dans cette optique, nous nous attardons d'abord à une BD 1D constituée de mixtures de gaussiennes. Une fonction de densité gaussienne est donnée par

$$G(x, \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (x \in \mathbb{R}) \quad (3.1)$$

où $\theta = (\mu, \sigma)$ sont les paramètres (moyenne, écart-type) de la densité. Une gaussienne pondérée par un poids κ sera notée

$$\Gamma(x, \gamma) = \kappa G(x, \theta) \quad (x \in \mathbb{R}) \quad (3.2)$$

où $\gamma = (\theta, \kappa)$.

Soient n un entier naturel strictement positif représentant le nombre de composantes de notre mixture, $\{\theta_i = (\mu_i, \sigma_i)\}_{i=1}^n$ une suite de paramètres de gaussiennes avec la suite des moyennes $\{\mu_i\}_{i=1}^n$ strictement croissantes, $\mu_1 < \dots < \mu_i < \dots < \mu_n$, et $\{\kappa_i\}_{i=1}^n$ une suite de poids strictement positifs telle que $\sum_{i=1}^n \kappa_i = 1$.

La mixture de gaussiennes, qui sera notée image 1D, est alors

$$I(x) = \sum_{i=1}^n \Gamma(x, \gamma_i) = \sum_{i=1}^n \kappa_i G(x, \theta_i) \quad (x \in \mathbb{R}) \quad (3.3)$$

où $\gamma_i = (\theta_i, \kappa_i)$ pour $i = 1, \dots, n$.

Chaque image de notre base de données sera une mixture définie sur l'intervalle $[0, 2^J)$. Elle sera échantillonnée en 2^J points x_k avec $(x_k = 0, 1, \dots, 2^J - 1)$.

Finalement, nous désirons que dans chaque mixture de gaussiennes le nombre de composantes soit identique au nombre de modes apparents, autrement dit, qu'il n'y ait pas de recouvrement de modes. Un algorithme décrit dans [1], nous permet de générer systématiquement des mixtures à n composantes avec n modes apparents. Notre base de données contient 6×150 mixtures de gaussiennes de 2^8 coefficients, chaque ensemble de 150 mixtures possède le même nombre de composantes n avec n modes apparents tel que $(n = 1, \dots, 6)$.

3.2 Recherche d'image 1D

3.2.1 Exemple

À cause du terme $\tilde{w}_0 |Q[0] - T[0]|$ de notre métrique, cette dernière discrimine les mixtures de gaussiennes de notre BD, limitant des surfaces différentes à celle limitée par la mixture de gaussiennes requête. Par conséquent, pour que notre recherche ne tienne pas

compte de ces surfaces, nous fixons \tilde{w}_0 à 0 avant l'échantillonnage.

Les Figures 3.1, 3.3, 3.5, 3.7, 3.9, 3.11, 3.13, 3.15, 3.17 et 3.19 représentent des boîtes de dialogue générées par Visual C++ 6.0, donnant des exemples de résultats de la recherche d'une même requête de 2^8 pixels (coefficients) dans la base de données décrite dans la section précédente, avec différents nombres de coefficients gardés pour la requête et les images de la BD après la compression. Chaque boîte de dialogue est sous forme d'une interface contenant neuf histogrammes représentant respectivement l'image requête et les 8 images cibles les plus proches de cette dernière. Les Figures 3.2, 3.4, 3.6, 3.8, 3.10, 3.12, 3.14, 3.16, 3.18 et 3.20 montrent les 24 images 1D cibles les plus proches de la requête, trouvées après les différentes recherches. Dans les différentes recherches décrites par ces figures, les poids associés à notre métrique et le nombre $(\tilde{w}_0, w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7)$ des coefficients gardés m à chaque compression sont présentés dans le tableau suivant

m	\tilde{w}_0	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7
100	0	11.81	11.28	8.40	8.51	0.74	13.18	8.03	4.48
90	0	12.94	11.82	9.40	10.78	9.28	8.84	0.11	8.91
80	0	13.07	12.31	11.17	11.35	9.78	10.21	0.38	9.72
70	0	13.28	12.29	11.35	10.67	7.70	10.69	0.30	10.14
60	0	13.40	11.97	11.28	10.74	10.42	9.63	0.93	10.11
50	0	13.67	12.11	11.70	11.06	10.86	10.84	0.45	10.55
40	0	14.07	12.49	12.27	11.70	11.57	10.83	10.58	11.37
30	0	14.51	12.99	12.76	12.26	12.23	12.30	0.78	12.68
20	0	15.14	12.88	12.65	12.44	12.33	12.69	12.54	12.82
10	0	14.35	14.21	13.64	13.71	13.60	13.69	13.80	13.83

Tableau 3.1 – Les poids \tilde{w}_0 et w_i ($i = 0, 1, \dots, 7$) associés à notre métrique pour différents ordres m de la compression de la requête et des images de la BD.

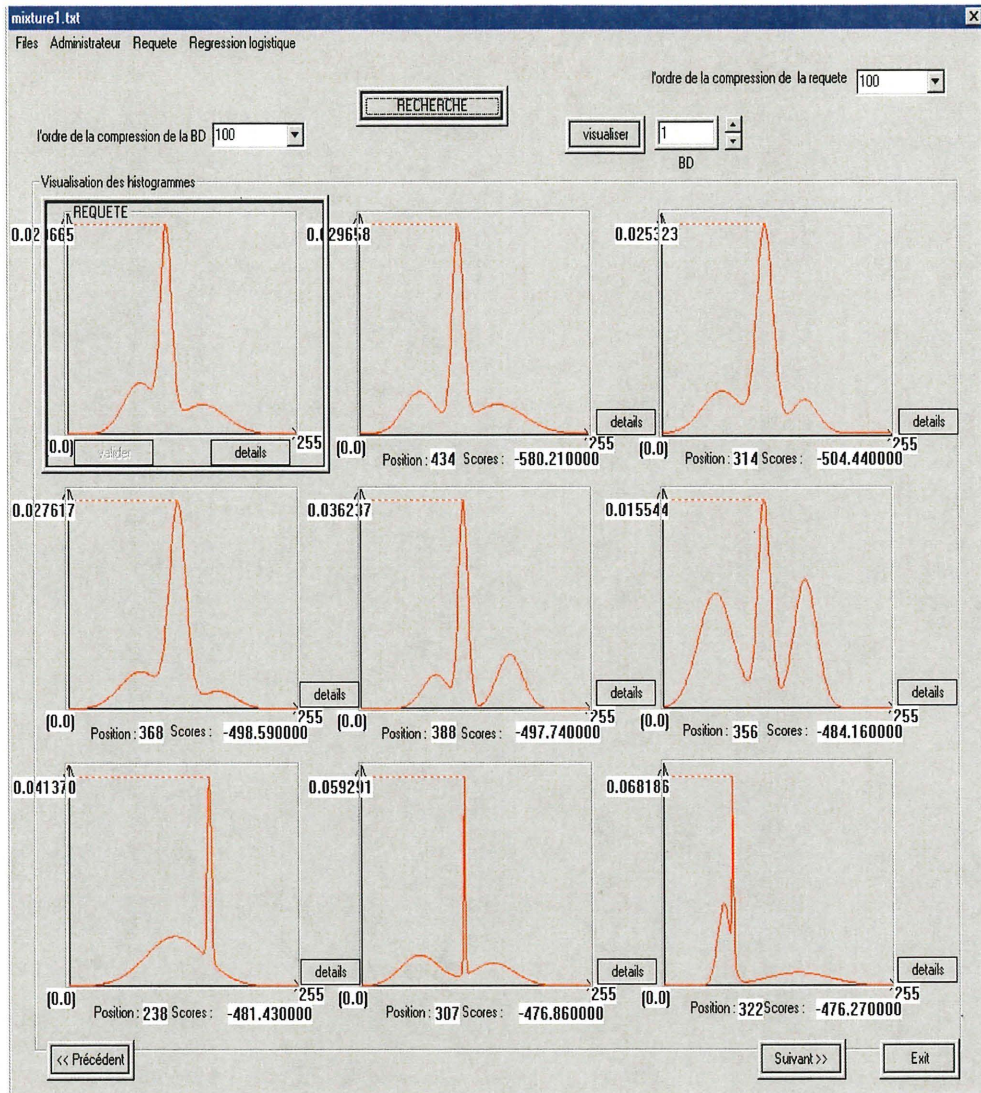
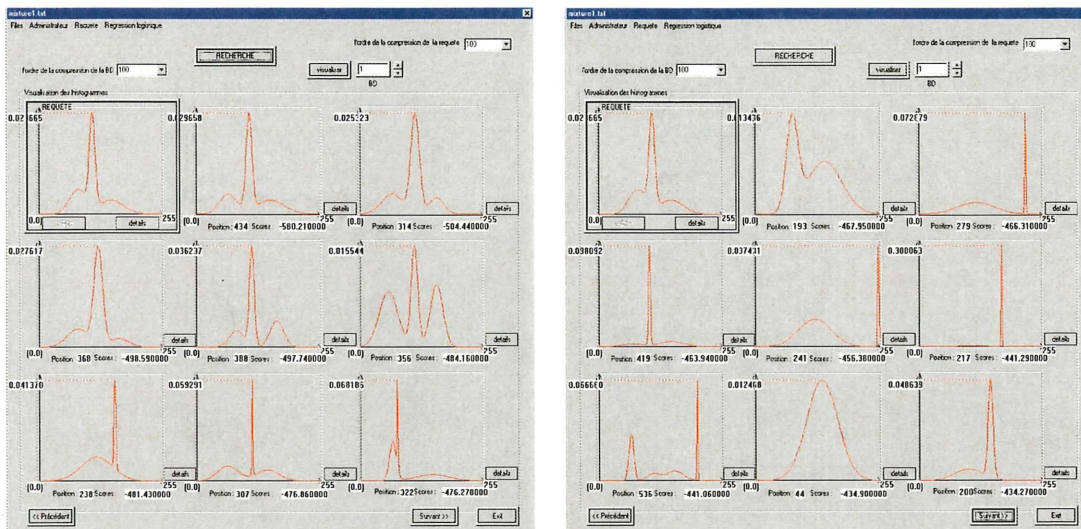
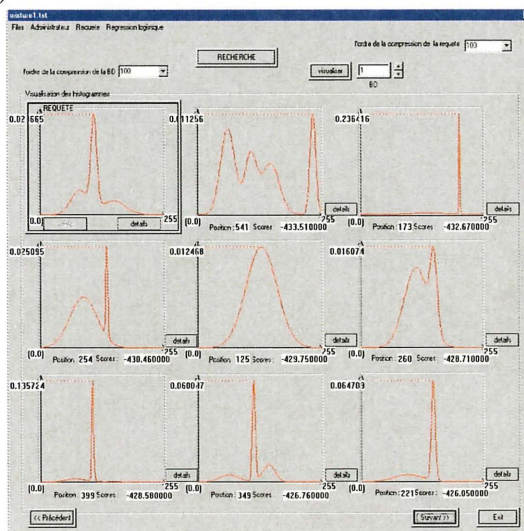


Figure 3.1 – Recherche avec compression à 100 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.2 – Recherche avec compression à 100 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

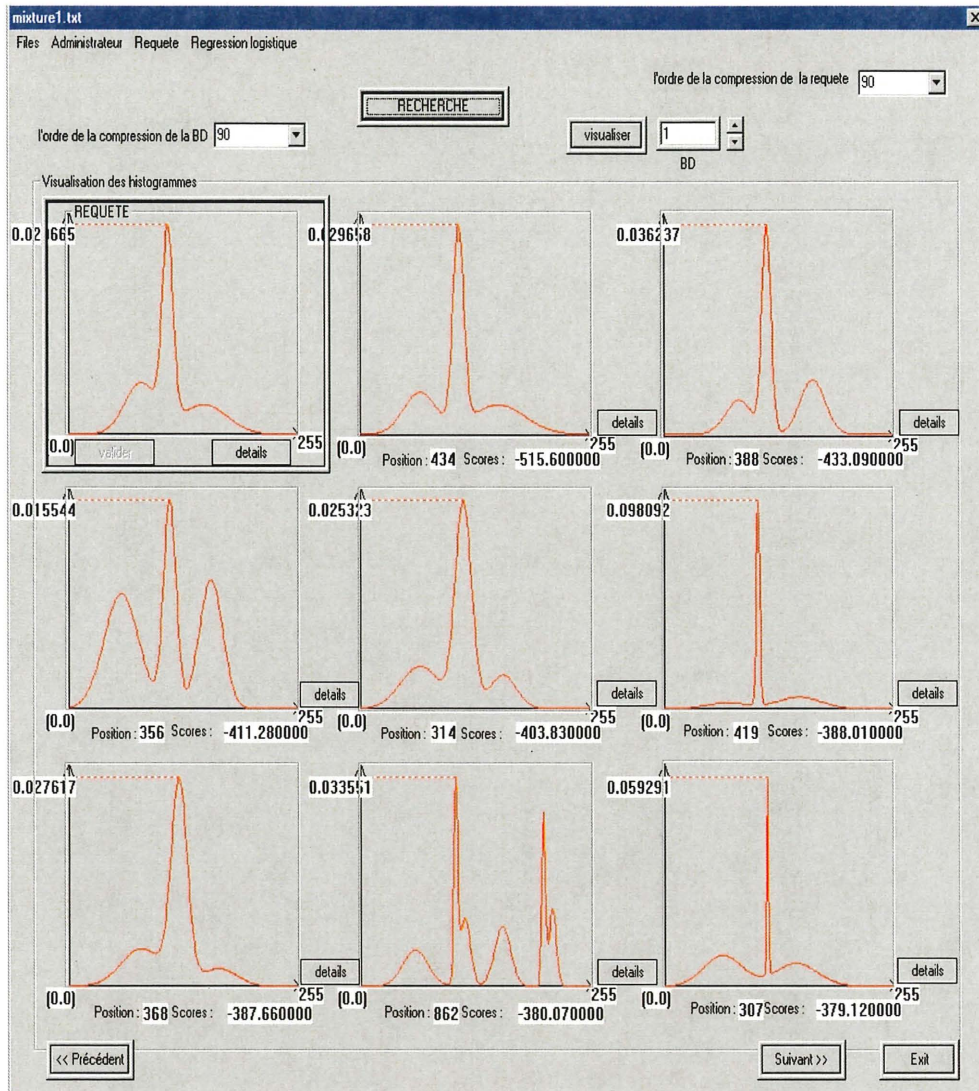
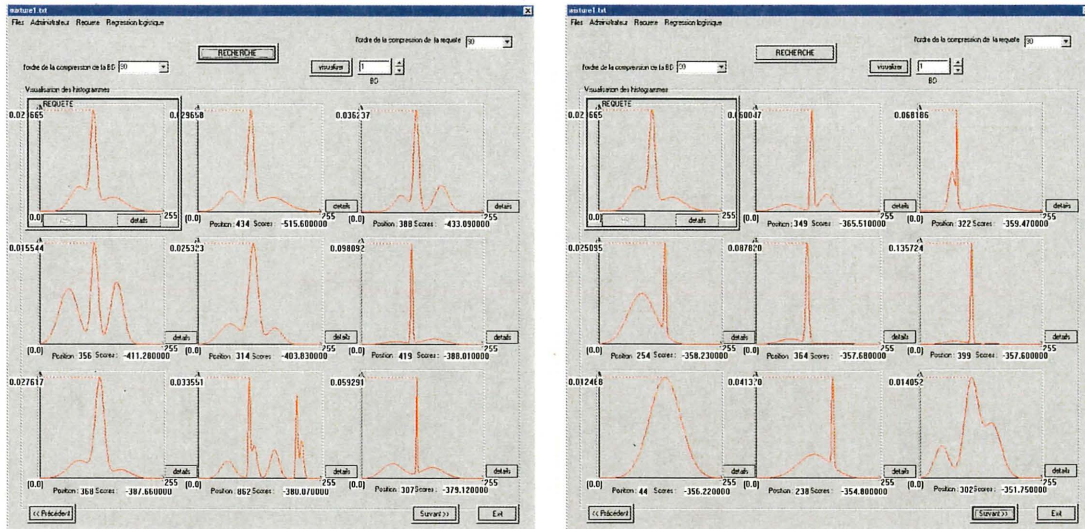
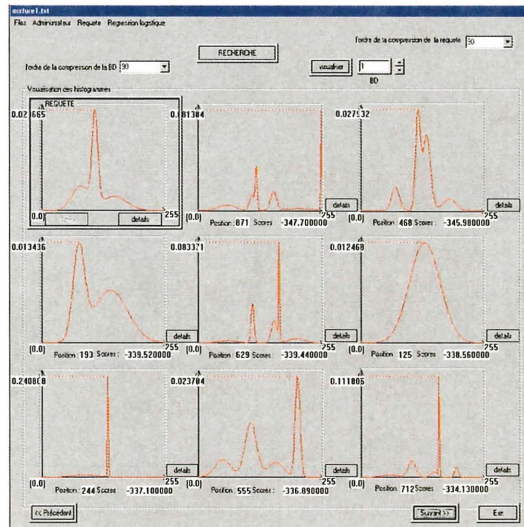


Figure 3.3 – Recherche avec compression à 90 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.4 – Recherche avec compression à 90 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

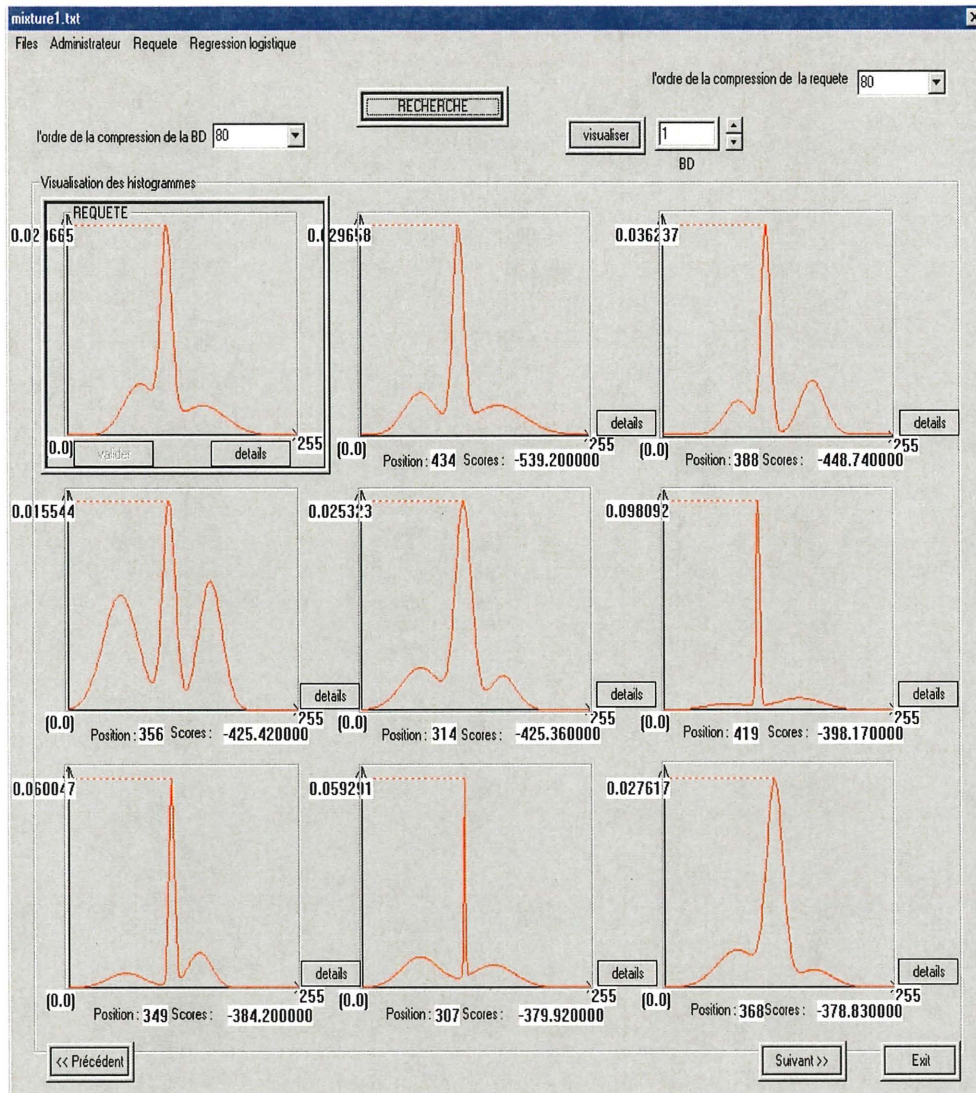
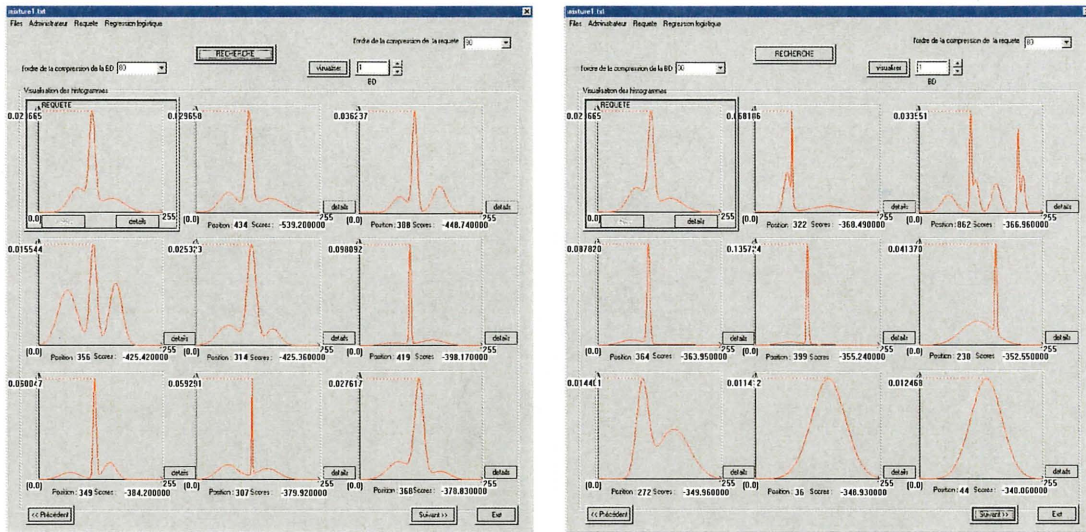
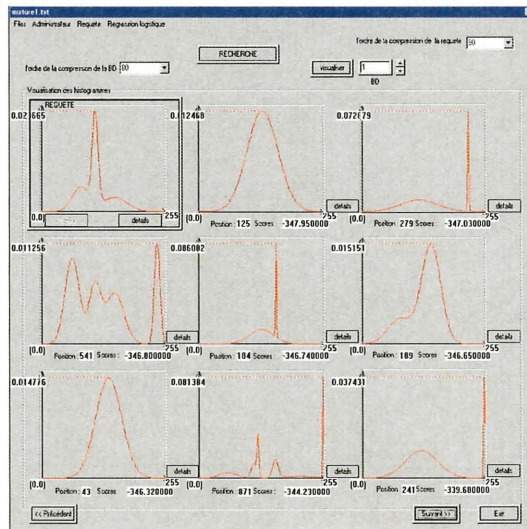


Figure 3.5 – Recherche avec compression à 80 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.6 – Recherche avec compression à 80 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

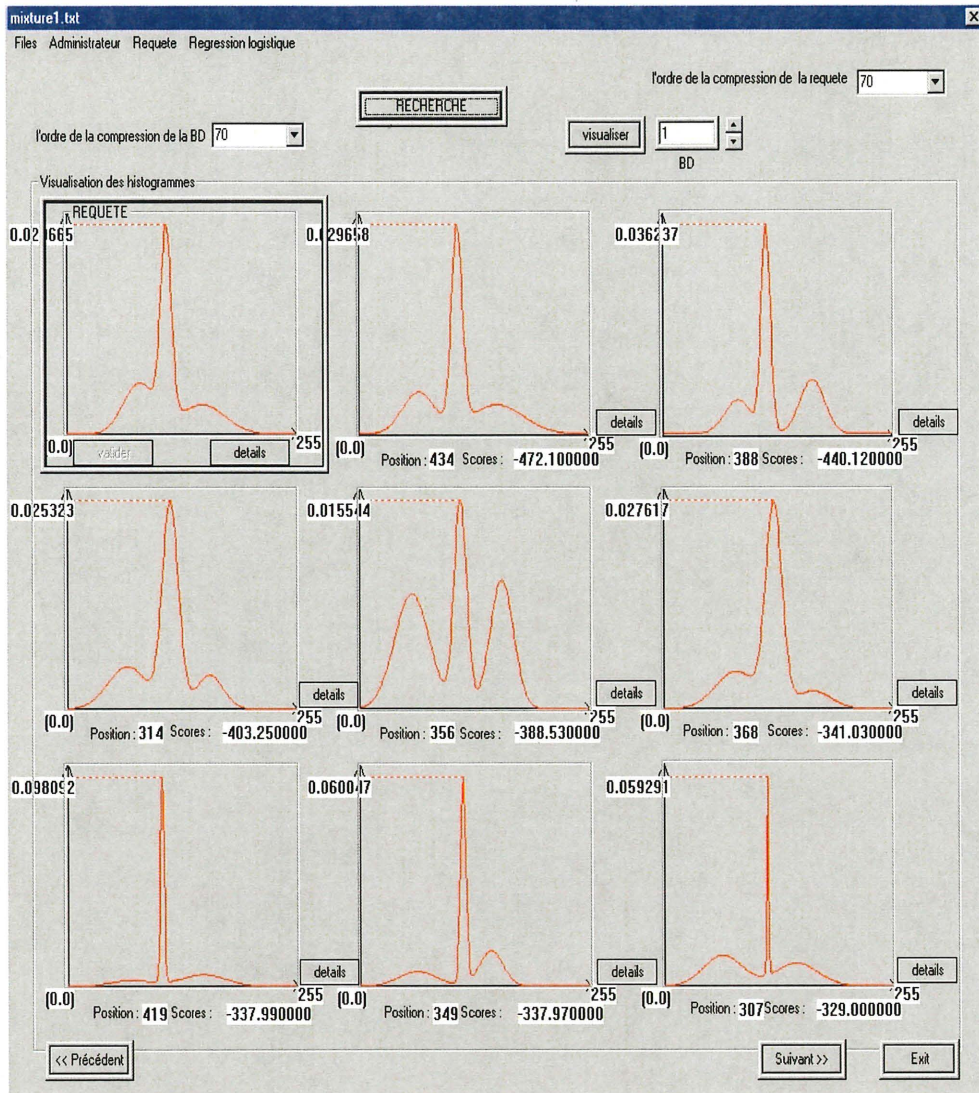
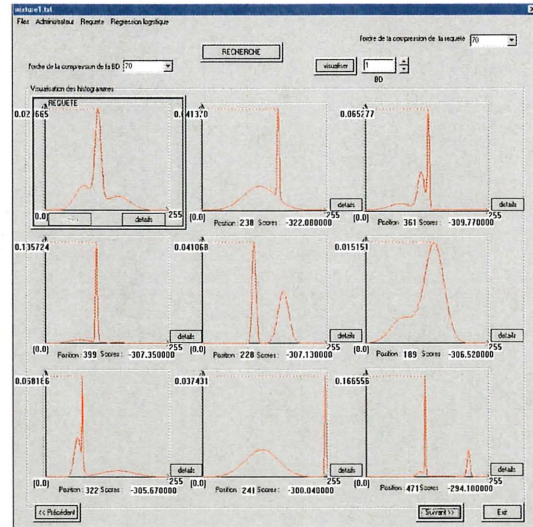
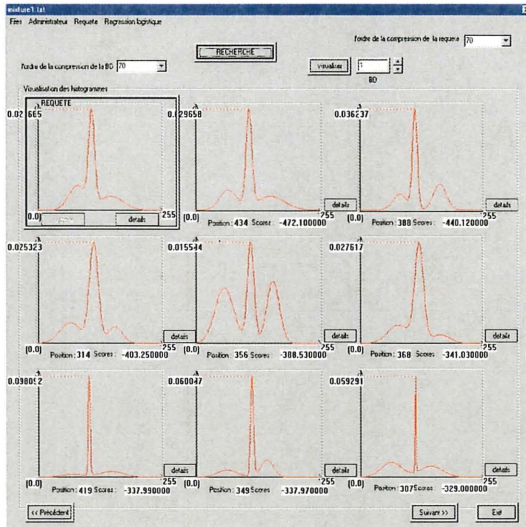
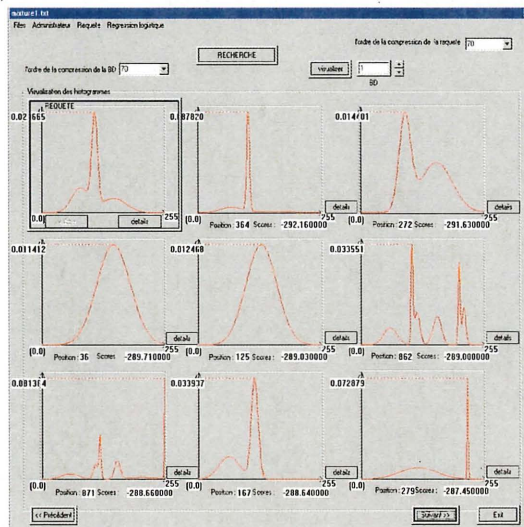


Figure 3.7 – Recherche avec compression à 70 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.8 – Recherche avec compression à 70 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

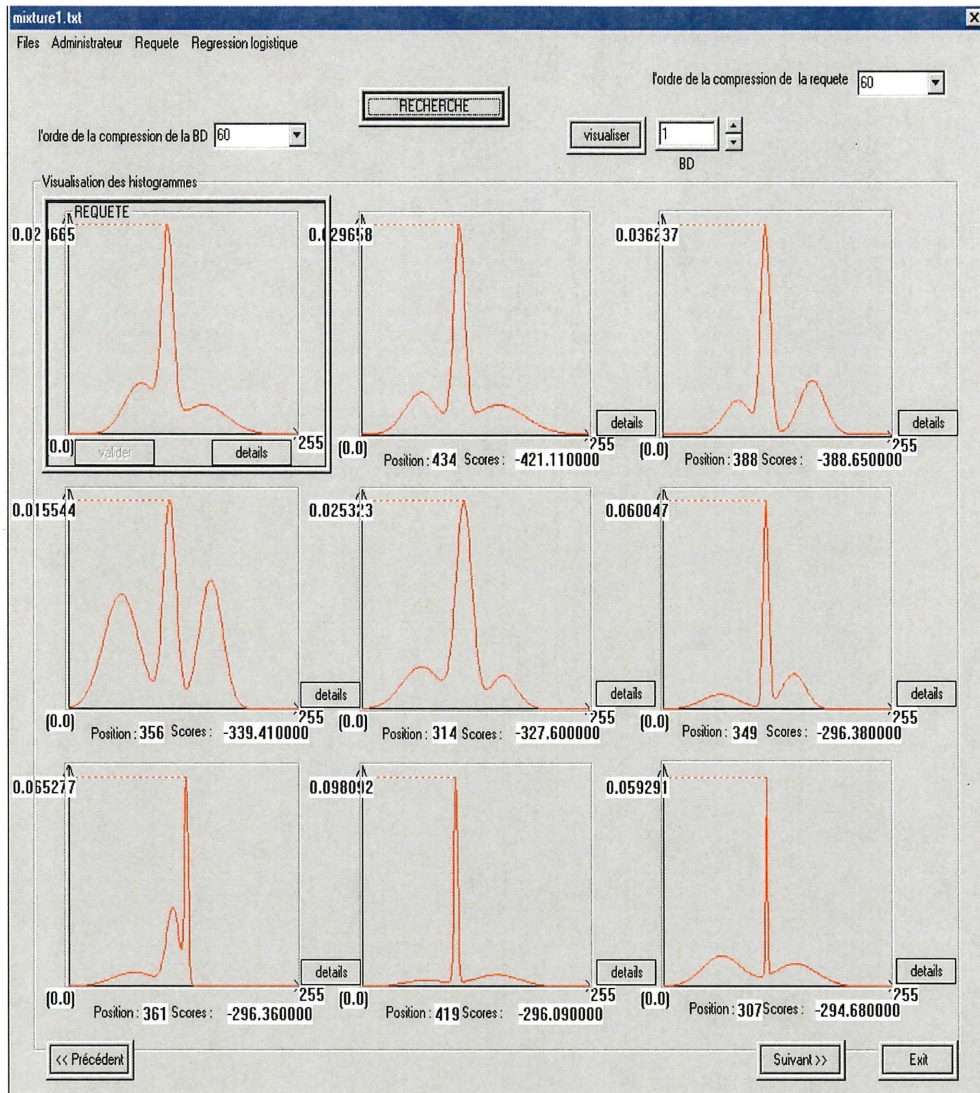
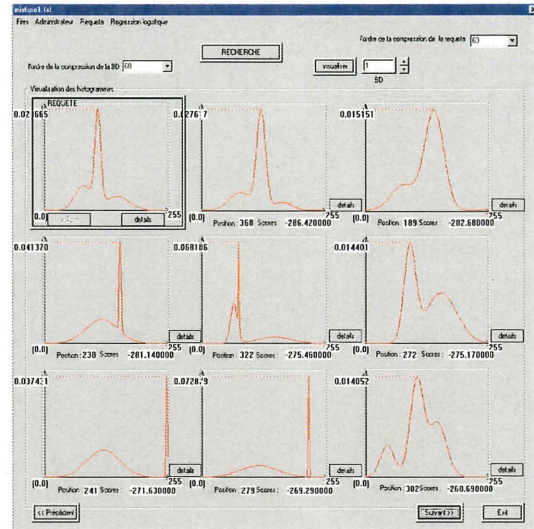
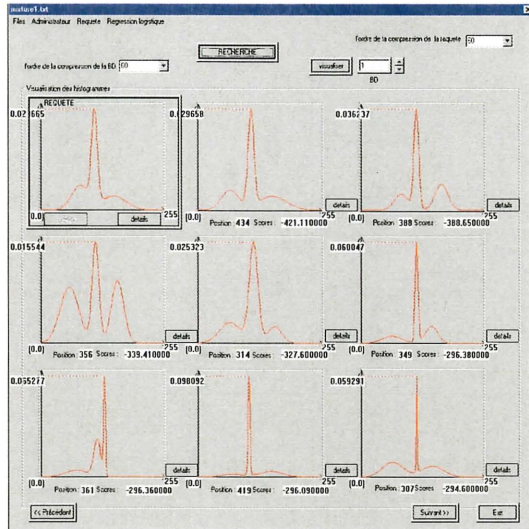
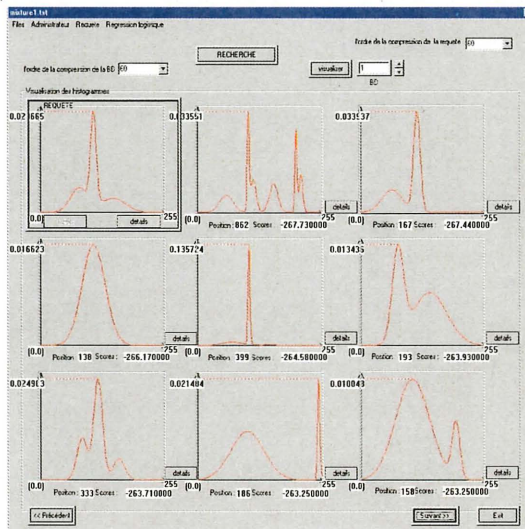


Figure 3.9 – Recherche avec compression à 60 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.10 – Recherche avec compression à 60 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

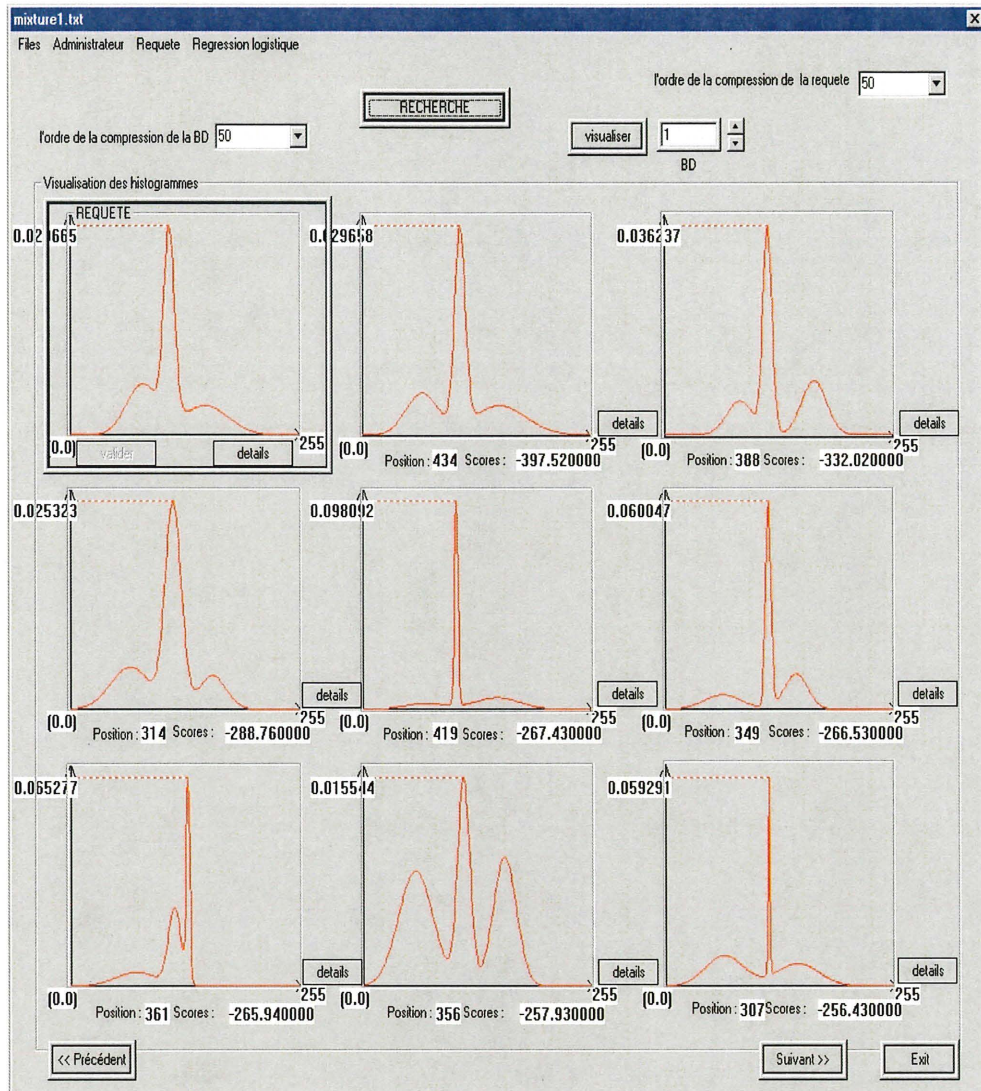
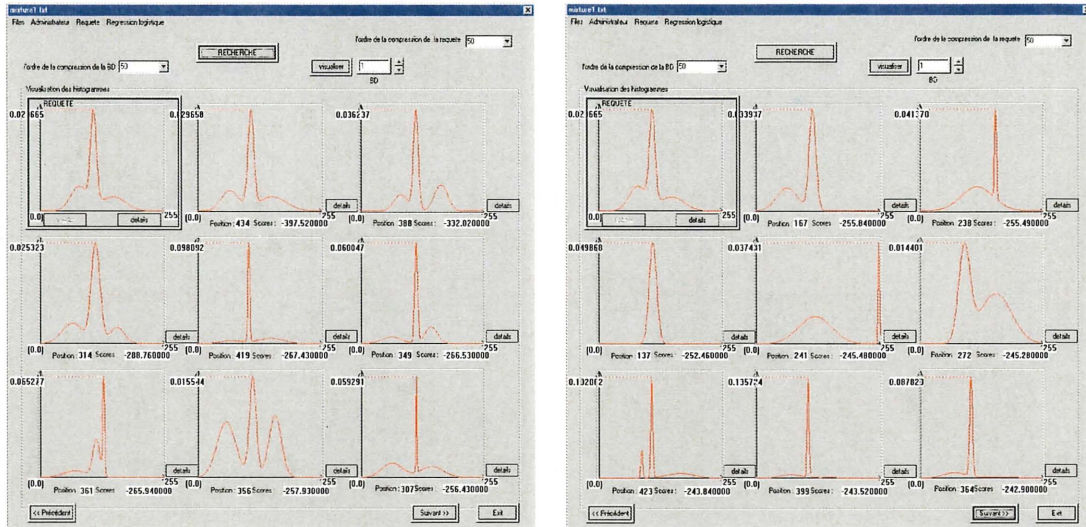
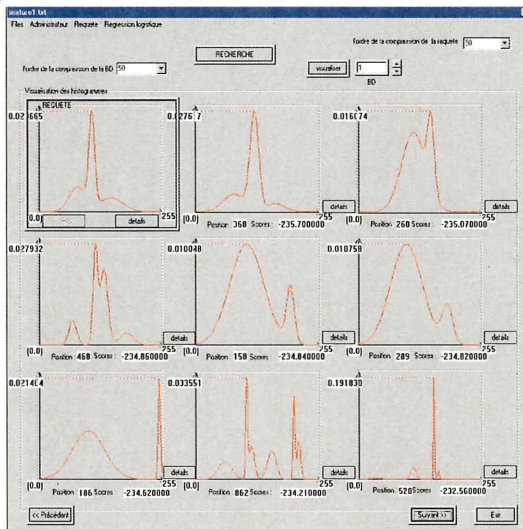


Figure 3.11 – Recherche avec compression à 50 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.12 – Recherche avec compression à 50 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

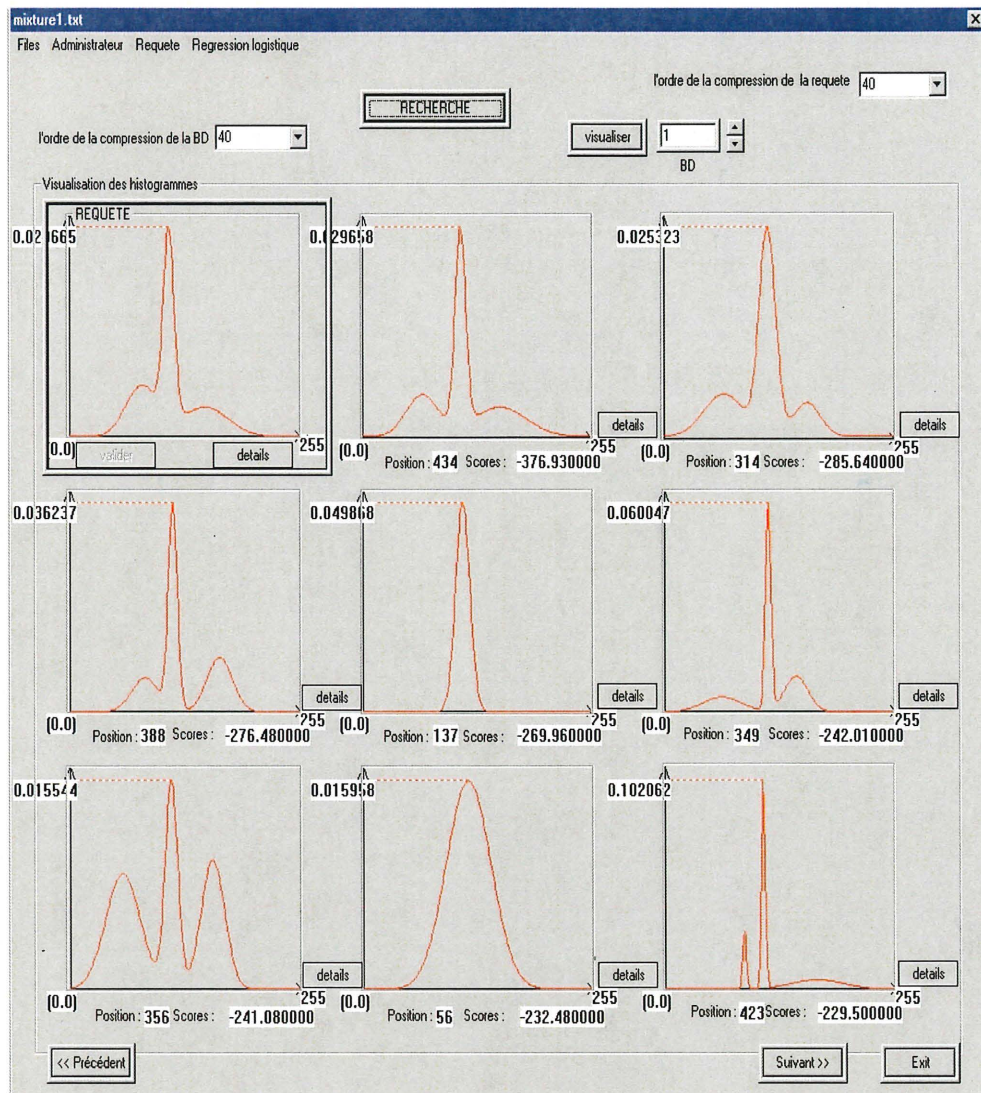
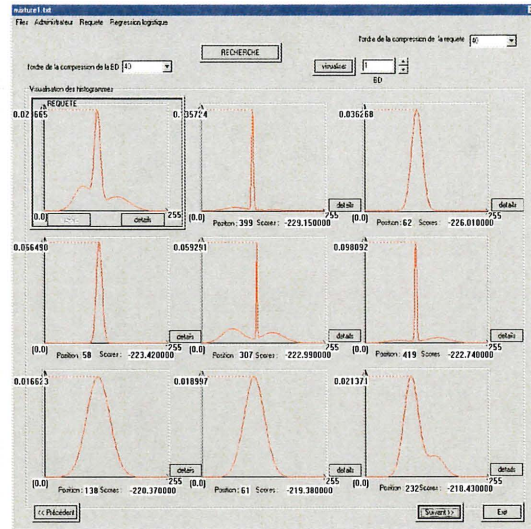
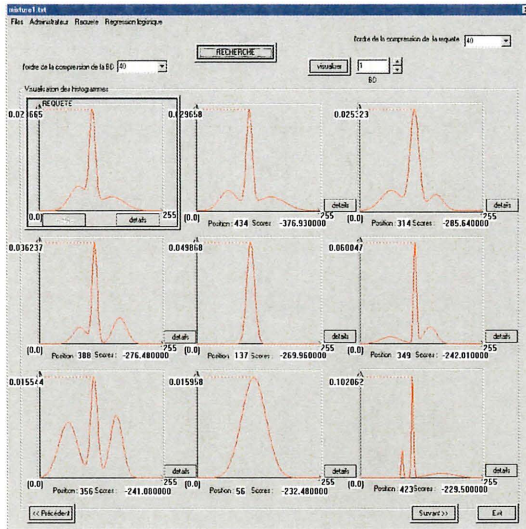
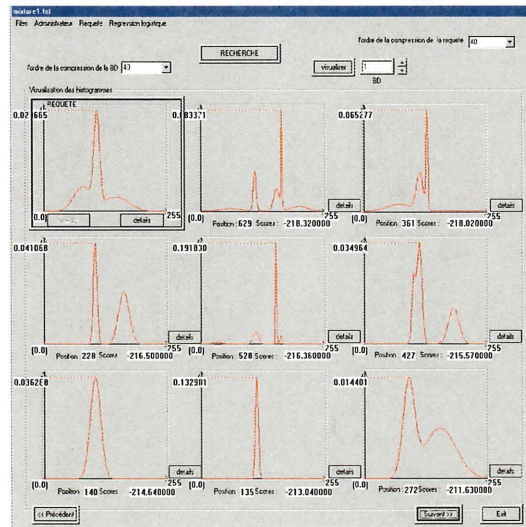


Figure 3.13 – Recherche avec compression à 40 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.14 – Recherche avec compression à 40 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

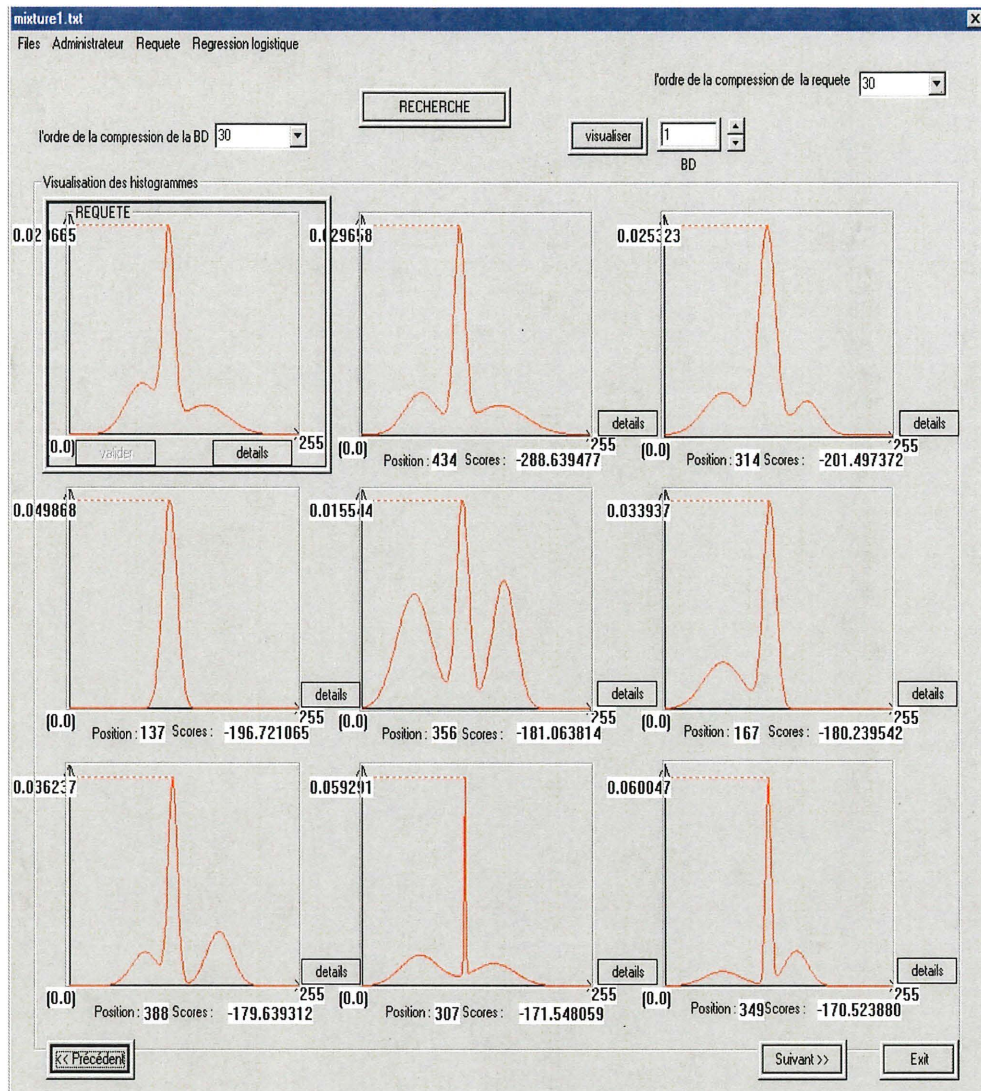
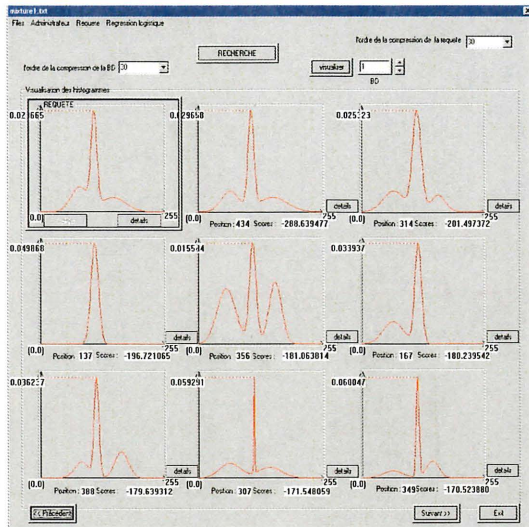
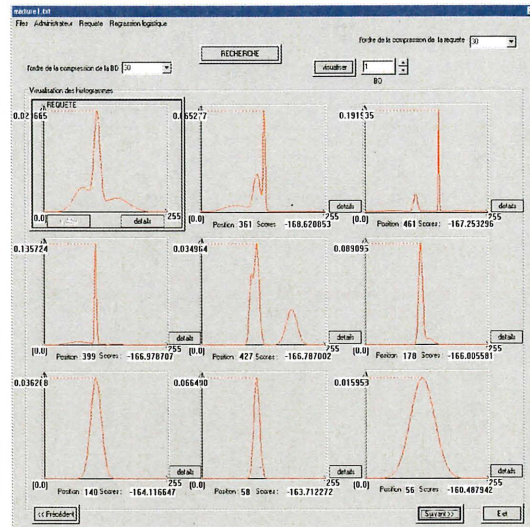


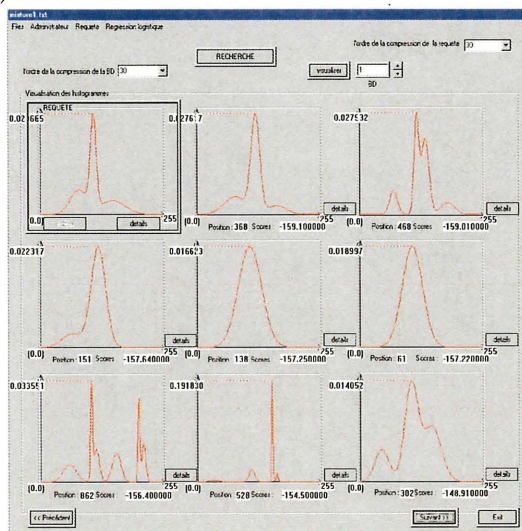
Figure 3.15 – Recherche avec compression à 30 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)



(b)



(c)

Figure 3.16 – Recherche avec compression à 30 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

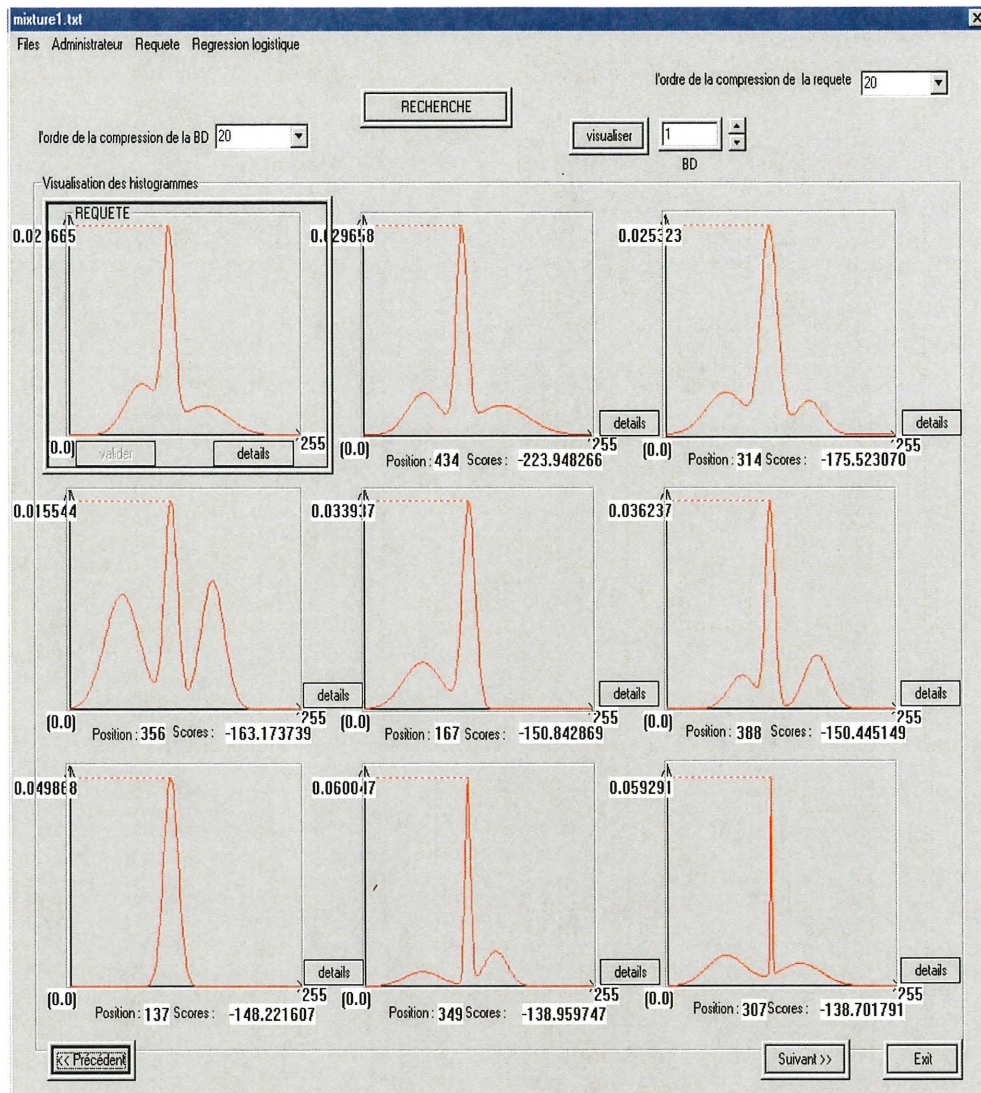
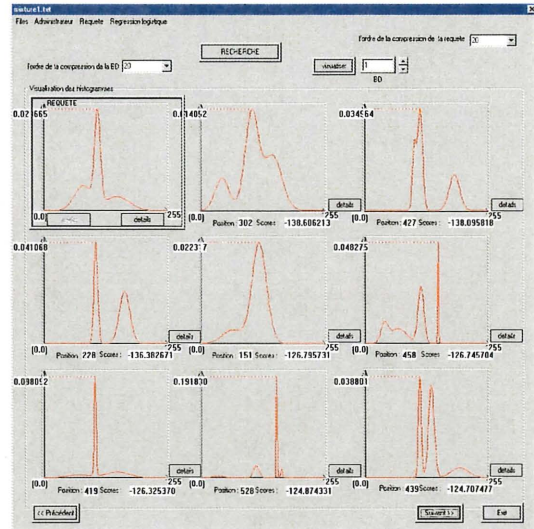
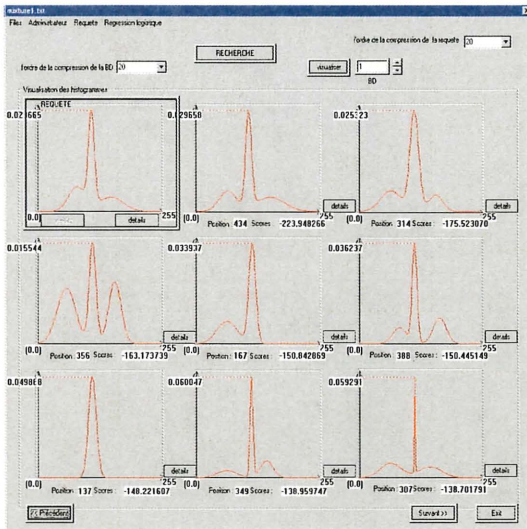
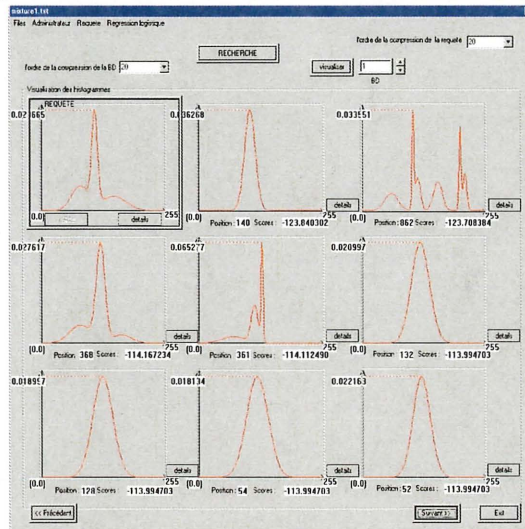


Figure 3.17 – Recherche avec compression à 20 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.18 – Recherche avec compression à 20 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

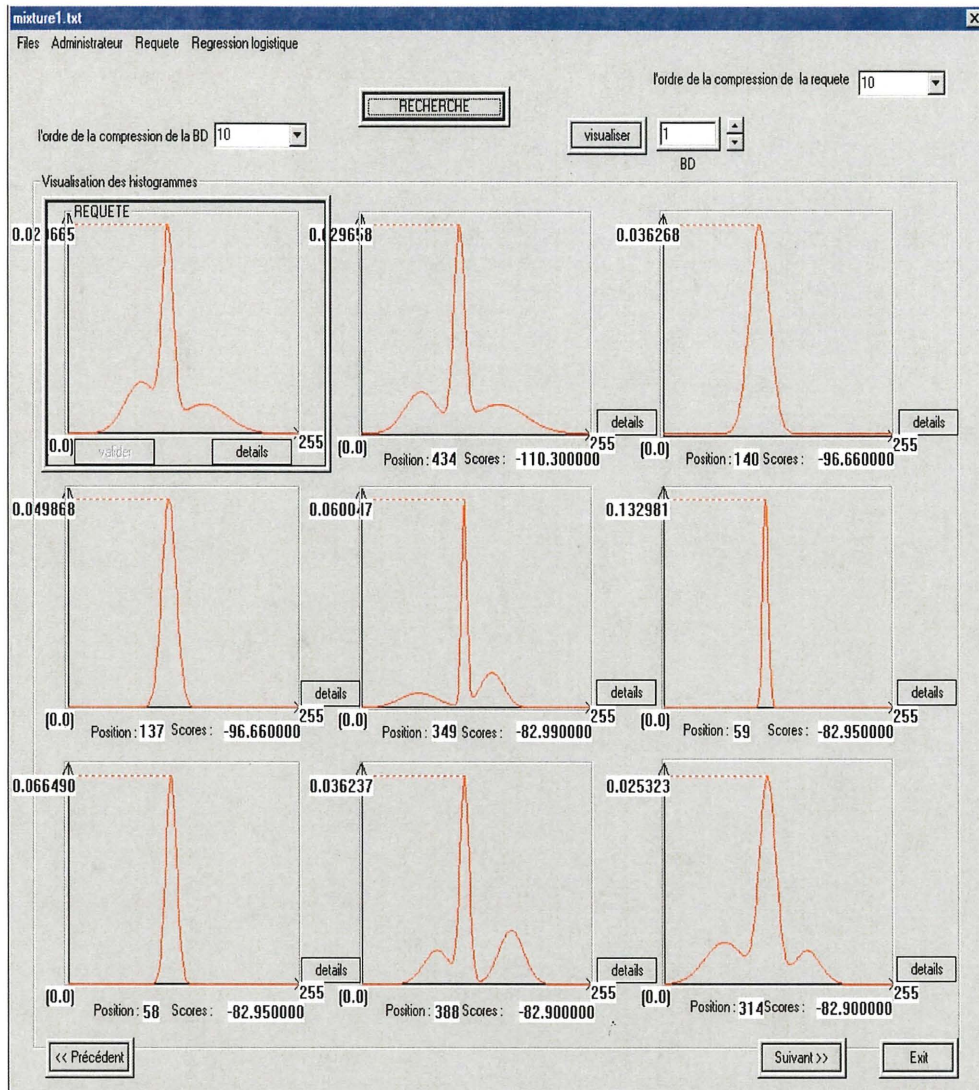
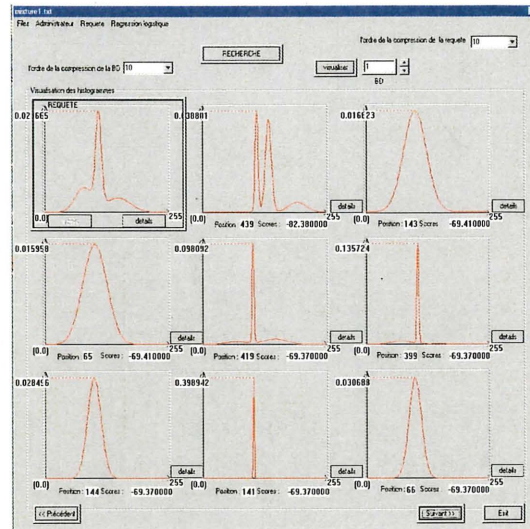
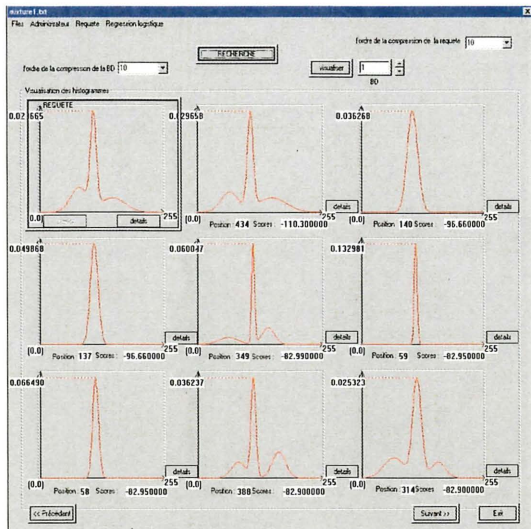
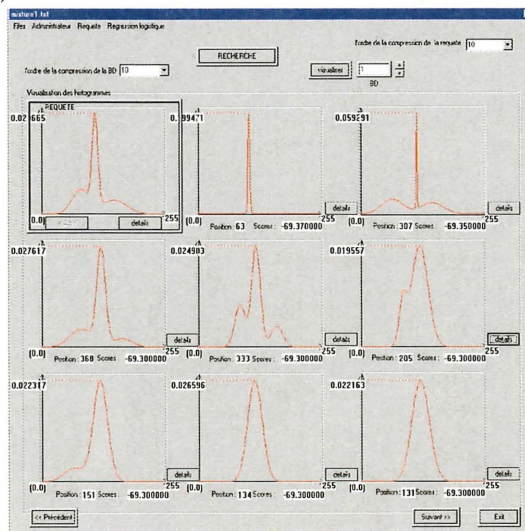


Figure 3.19 – Recherche avec compression à 10 coefficients. Les 8 images cibles 1D les plus proches de la requête.



(a)

(b)



(c)

Figure 3.20 – Recherche avec compression à 10 coefficients. Les 24 images cibles 1D les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

3.3 Évaluation

Dans cette section, nous allons évaluer notre méthode de recherche dans la base de données de mixtures de gaussiennes. Une cible est dite désirée, si elle présente une grande ressemblance à la requête au niveau du nombre de modes et des attributs (e.g. moyennes, variances).

3.3.1 Réglage : nombre de coefficients gardés

Afin de déterminer l'influence de la compression de la requête et des images de la BD, nous prenons aléatoirement un échantillon de 300 mixtures de gaussiennes de notre BD et nous y effectuons la recherche de 5 requêtes différentes. Pour chaque nombre de coefficients gardés d'une requête et des images de l'échantillon de notre BD après la compression, nous déterminons le rang de la dernière cible désirée trouvée parmi les 24 premières images 1D atteintes par la recherche. Par conséquent, pour les 5 requêtes choisies nous calculons les moyennes des rangs des dernières cibles désirées trouvées parmi les 24 premières mixtures atteintes par les différentes recherches, pour différents ordres de compression. La Figure 3.21 montre ces moyennes en fonction des différents ordres de compression.

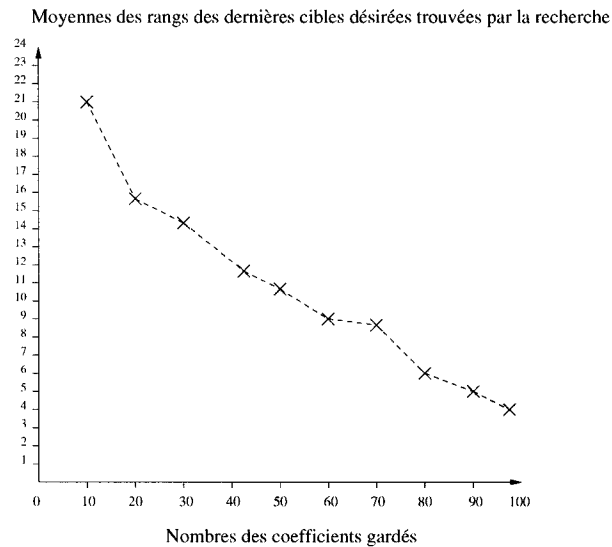


Figure 3.21 – *Les moyennes des rangs des dernières cibles désirées trouvées parmi les 24 premières images atteintes par les recherches, pour plusieurs ordres de compression de chaque requête et des images de l'échantillon de notre BD.*

Grâce à la Figure 3.21, il paraît que la capacité discriminatoire de la recherche devient de plus en plus faible quand l'ordre de la compression de la requête et des images de la BD diminue.

3.3.2 Influence des poids

Les poids w_k représentent le degré de correspondance qu'a un coefficient appartenant à un niveau de résolution $k \in \{0,1,2,3,4,5,6,7\}$, gardé après la compression à m coefficients et la quantification, à un autre élément de la requête appartenant au même niveau de résolution, ayant même position et signe. Grâce à la régression logistique, nous pouvons déterminer des poids w_k qui optimiseront la probabilité qu'une image l soit la cible désirée. Pour tester l'efficacité de la régression logistique nous pouvons effectuer une recherche avec des poids déterminés par cette dernière et autre avec des poids $w_k = 1$ pour tout

$k \in \{0,1,2,3,4,5,6,7\}$. Les Figures 3.22, 3.23 et 3.24 illustrent des comparaisons entre les deux recherches pour différents ordres de compression de la requête et des images de la BD.

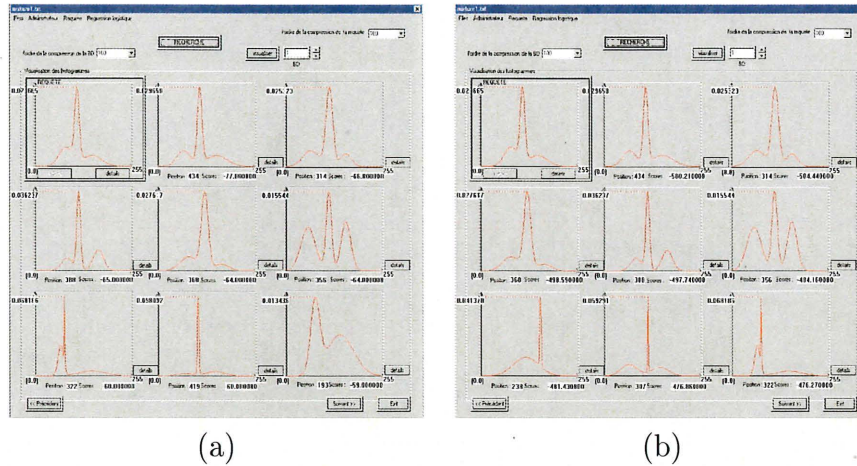


Figure 3.22 – Comparaison avec $m = 100$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.

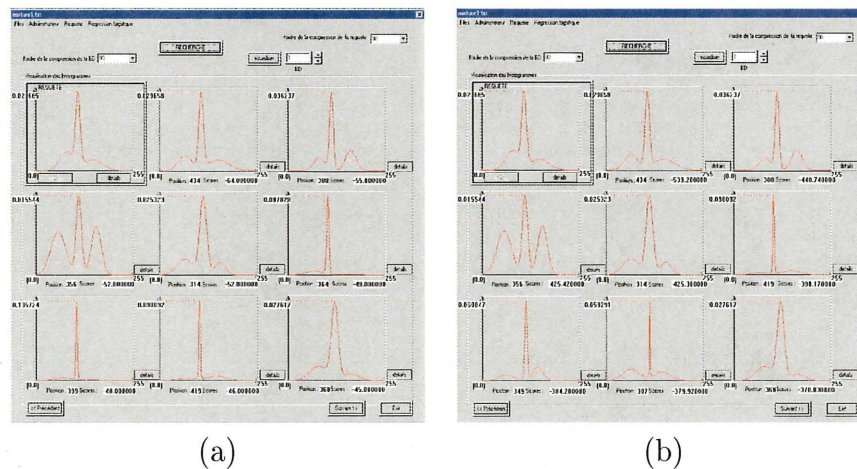


Figure 3.23 – Comparaison avec $m = 80$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.

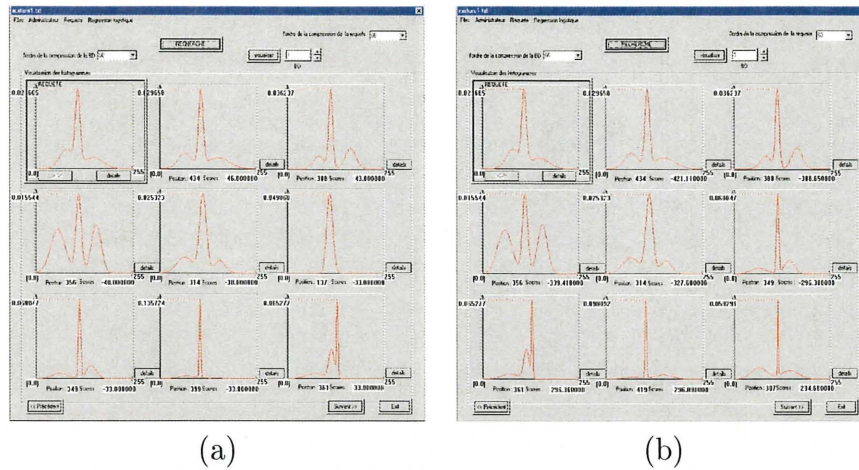


Figure 3.24 – *Comparaison avec $m = 60$: a) Recherche avec des poids égaux à 1. b) Recherche avec des poids déterminés par la régression logistique.*

Grâce aux Figures de comparaisons précédentes, il apparaît que les poids déterminés par la régression logistique permettent d'obtenir une meilleure recherche. En effet, quand l'ordre de la compression augmente la qualité de la recherche avec les poids égaux à 1 devient de plus en plus mauvaise comparativement à la recherche avec les poids déterminés par la régression logistique. Par exemple, la cinquième cible trouvée par la recherche décrite par (a) dans la Figure 3.24 ne contient qu'un seul mode comparativement à la requête qui en contient 3. Il est à noter que des expérimentations additionnelles ont également mené à cette même conclusion.

3.3.3 Translation

Puisque notre requête est une mixture de gaussiennes, il suffit de la régénérer sur un intervalle translaté pour que la translation soit effectuée. Afin de tester la robustesse de notre méthode de recherche à la translation, nous prenons une image de la BD comme requête et on effectue plusieurs recherches pour plusieurs versions translatées de cette dernière. À chaque translation d'une unité de la requête on effectue une recherche, jus-

qu'à ce que la cible la plus proche de notre requête n'est plus la requête elle-même. La Figure 3.25 représente trois boîtes de dialogue générées par Visual C++ 6.0, donnant trois résultats de la recherche d'une image requête appartenant à la BD, dont deux résultats sont fournis, quand la requête est translatée de certaines unités vers le côté gauche.

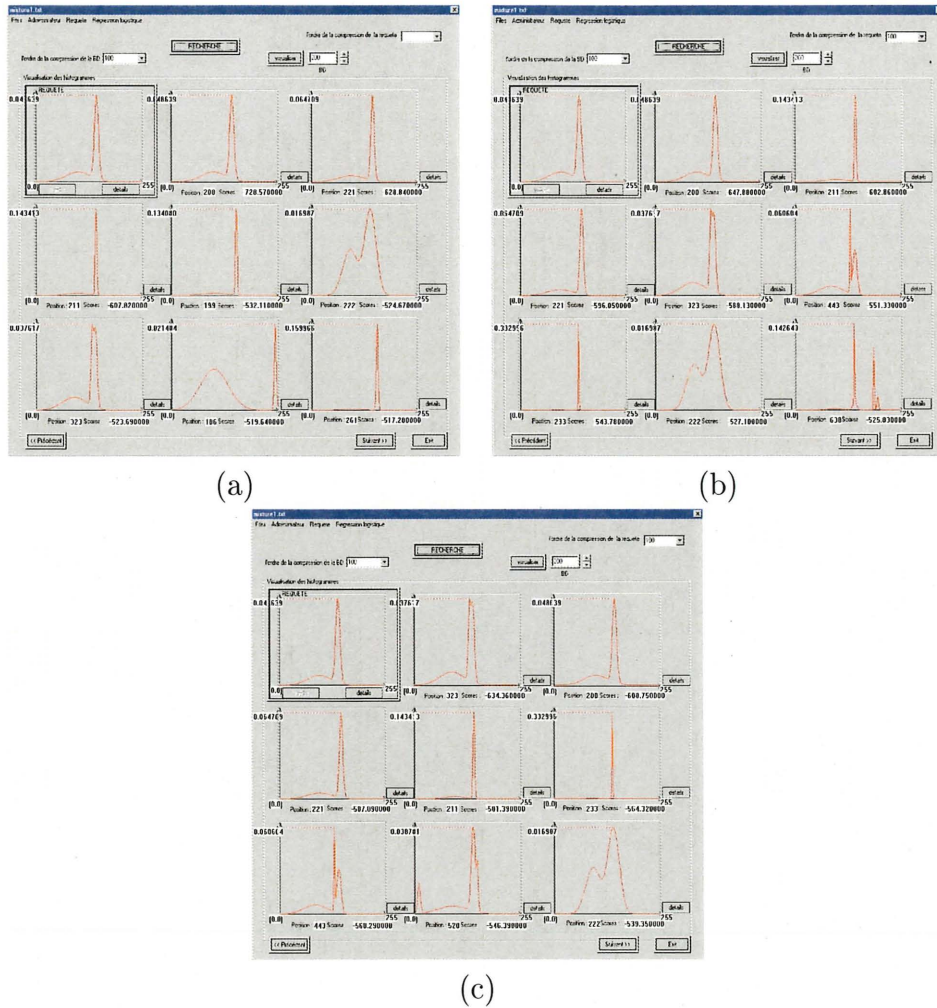


Figure 3.25 – Recherche : a) Requête originale. b) Requête translatée de 3 unités vers le côté gauche. c) Requête translatée de 5 unités vers le côté gauche.

Lorsque la requête est translatée de 5 unités, la cible la plus proche de la requête n'est plus la requête elle-même. Par conséquent, nous pouvons conclure que notre méthode de recherche est sensible à la translation.

3.3.4 Bruit gaussien dans la requête

Un bruit blanc gaussien $b \sim N(\mu_b, \sigma_b)$ est une petite perturbation locale des valeurs des pixels d'une image 1D, qui disparaît grâce à la compression. Pour déterminer la sensibilité de notre recherche par rapport à la perturbation de la requête par un bruit gaussien, nous cherchons les différents ordres m de la compression de la requête et des images de la BD, pour lesquels on trouve des résultats de la recherche répondant adéquatement aux critères fixés, c'est-à-dire comprenant un maximum de cibles désirées et ce, malgré les perturbations de la requête par des bruits gaussiens de moyennes $\mu_b = 0$ et de certains écart-types σ_b . Les différentes valeurs des σ_b et des m pour lesquelles on trouve des bons résultats de la recherche, sont décrites dans le tableau suivant.

m	σ_b
60	$1 * 10^{-3}$
40	$2 * 10^{-3}$
40	$3 * 10^{-3}$
30	$4 * 10^{-3}$
30	$5 * 10^{-3}$
30	$6 * 10^{-3}$
20	$7 * 10^{-3}$
10	$8 * 10^{-3}$
10	$9 * 10^{-3}$

Tableau 3.2 – Les valeurs des ordres m de la compression pour lesquelles on trouve des bons résultats de la recherche, malgré les bruits gaussiens d'écart-type σ_b perturbant la requête.

Quand la valeur de l'écart-type du bruit gaussien est supérieure à $9 * 10^{-3}$, quelque soit

la valeur du nombre des coefficients gardés m de la requête et des images de la BD après la compression, nous ne trouvons pas des bons résultats de la recherche. Les Figures 3.26 et 3.27 nous montrent deux exemples de l'effet du bruit gaussien sur la recherche.

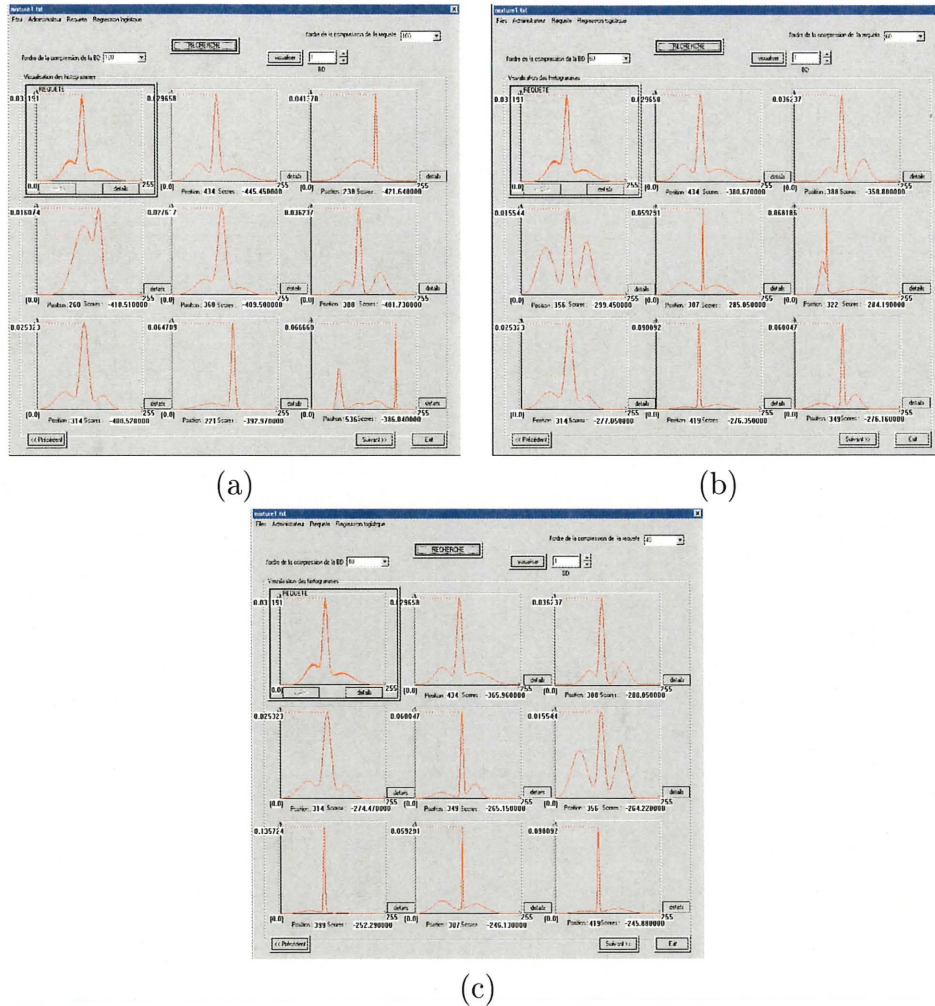
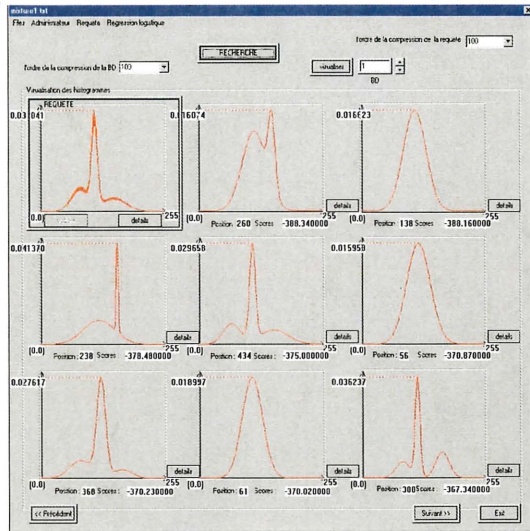
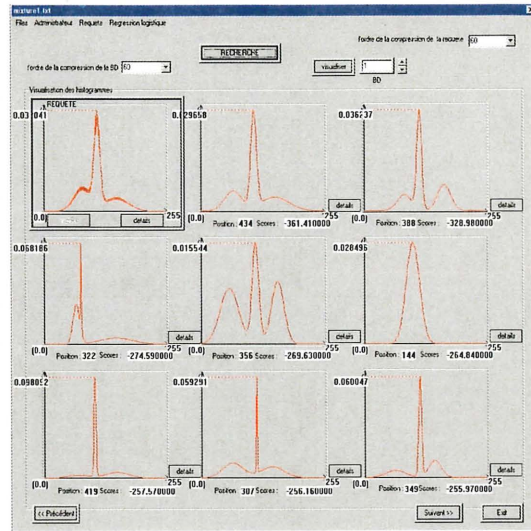


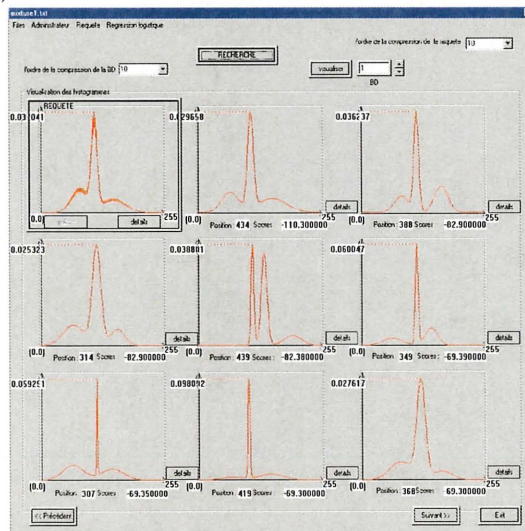
Figure 3.26 – Recherche : requête perturbée par un bruit gaussien $b \sim N(0, \sigma_b)$ avec $\sigma_b = 3 * 10^{-3}$. a) $m = 100$. b) $m = 60$. c) $m = 40$.



(a)



(b)



(c)

Figure 3.27 – Recherche : requête perturbée par un bruit gaussien $b \sim N(0, \sigma_b)$ avec $\sigma_b = 8 * 10^{-3}$. a) $m = 100$. b) $m = 60$. c) $m = 10$.

3.3.5 Multiplication d'intensité

Multiplier par un réel positif a l'image 1D correspond à multiplier tous ses coefficients par a . Par conséquent, l'ensemble des coefficients d'ondelettes de l'image $I(x)$ multipliés par a , est identique à celui des coefficients d'ondelettes de l'image $aI(x)$. Puisque la compression met à zéro les plus petits coefficients d'ondelettes en valeur absolue, les coefficients gardés des images décomposées $I(x)$ et $aI(x)$ auront les mêmes positions. Lorsque les deux images $I(x)$ et $aI(x)$ sont quantifiées, elles deviennent identiques, seuls leurs facteurs d'échelles diffèrent d'un facteur de a . Par conséquent, la multiplication de notre requête par a pourrait avoir un impact dans notre méthode de recherche, car le terme $\tilde{w}_0|Q[0] - T[0]|$ de notre métrique n'est pas linéaire. Cependant, dans le cadre de notre travail, le poids \tilde{w}_0 est nul, par conséquent, la multiplication de la requête par a n'aura aucun effet sur notre recherche.

3.3.6 Vitesse d'exécution (complexité)

L'algorithme de la décomposition en ondelettes de Haar d'une image 1D de N coefficients, nécessite $2 * (N - 1)$ opérations d'affectation. La complexité de l'algorithme de la compression d'une image 1D de N coefficients en m coefficients est de $m + N * (\theta + 1)$ opérations d'affectation, avec θ est un grand entier dépendant des plus grand et plus petit coefficients en valeur absolue de l'image 1D. L'algorithme de la quantification d'une image 1D de N coefficients exige $N - 1$ opérations d'affectation. Le calcul de la métrique entre une image requête 1D compressée à m coefficients et les images 1D d'une base de données contenant n images, nécessite $n * (2 * m + 1)$ opérations d'affectation. Avant d'effectuer la recherche d'une requête de N coefficients dans une base de données de n images 1D, cette dernière doit être décomposée, compressée et quantifiée. Par conséquent, la complexité de la recherche est de $(4 + \theta) * N + m * (2 * n + 1) + n - 3$ opérations d'affectation.

D'où le nombre des coefficients gardés m de la requête et des images de la BD après la compression est inversement proportionnel avec la vitesse d'exécution de la recherche.

CHAPITRE 4

Applications

Afin de tester l'efficacité et la validité de notre méthode de recherche, nous appliquons cette dernière sur d'autres bases de données. Nous utiliserons une base de données de poissons de Surray et une base de données d'images couleurs.

4.1 Base de données de poissons de Surray

La base de données de poissons de Surray contient 1000 poissons. Chaque poisson est représenté par un contour simple fermé et décrit par un ensemble de N_p pixels $\{P_0, P_1, \dots, P_{N_p-1}\}$. Chaque pixel est représenté par un couple de coordonnées cartésiennes $P_i = (x_i, y_i)$ avec $i \in \{0, \dots, N_p - 1\}$ et $P_{N_p} = P_0$. Pour pouvoir appliquer notre méthode de recherche, nous associons à chaque poisson de Surray un histogramme qui est construit à partir des distances euclidiennes d_i des pixels $P_i = (x_i, y_i)$ au centre de gravité $G = (x_G, y_G)$ du poisson pour $i = 0, 1, \dots, N_p$, avec $d_i = \sqrt{(x_i - x_G)^2 + (y_i - y_G)^2}$. Le pixel P_0 sera le point de départ du parcours des pixels d'un poisson, lors de la construction de son histogramme. Puisque notre méthode de recherche est très sensible à la translation, les

pixels P_0 des poissons de notre base de données sont choisis sur les bouts des queues des poissons et dans un même repère. De plus, le sens du parcours des pixels doit être le même pour tous les poissons. Dans le cadre de notre travail, nous avons choisi le sens trigonométrique. Chaque histogramme associé à un poisson de Surray est donné en $N_p + 1$ points ξ_k ($k = 0, \dots, N_p$) tels que $\xi_0 = 0$, $\xi_k = \sum_{j=1}^k \Delta x_{j-1,j}$ pour tout $k \in \{1, \dots, N_p\}$ où $\Delta x_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ est la distance euclidienne entre deux pixels P_i et P_j . Par conséquent, chaque histogramme d'un poisson de Surray est défini par une famille de couples $\{(\xi_0, d_0), (\xi_1, d_1), \dots, (\xi_{N_p}, d_{N_p})\}$. Afin de rendre notre recherche possible, nous devons transformer chaque histogramme d'un poisson de Surray de notre base de données en une image 1D de $N = 2^8$ coefficients, pour cela nous devons effectuer une interpolation.

Notre interpolation se fait suivant la procédure suivante. Elle transforme un histogramme d'un poisson de Surray de N_p pixels en une image 1D de $N = 2^8$ coefficients c_i ($i = 0, 1, \dots, 2^8 - 1$).

$S_x \leftarrow 0$

pour $i \leftarrow 0$ **jusqu'à** N_p **faire**

$\xi_i \leftarrow 0$

fin pour

pour $k \leftarrow 1$ **jusqu'à** $N_p - 1$ **faire**

$\xi_k \leftarrow \xi_{k-1} + \Delta x_{k-1,k}$

fin pour

$\xi_{N_p} \leftarrow \xi_{N_p-1} + \Delta x_{N_p-1,0}$

$c_0 \leftarrow d_0$

pour $i \leftarrow 1$ **jusqu'à** $N - 1$ **faire**

$S_x \leftarrow i * \frac{\xi_{N_p}}{N}$

pour $k \leftarrow 1$ **jusqu'à** N_p **faire**

si $\xi_{k-1} < S_x \leq \xi_k$ **alors**

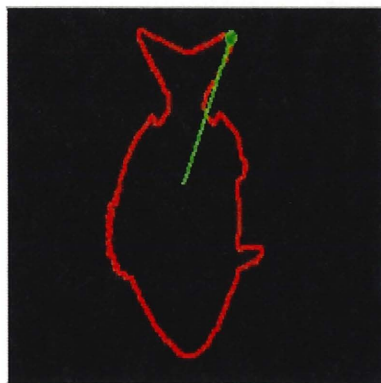
$c_i \leftarrow \left(\frac{S_x - \xi_k}{\xi_{k-1} - \xi_k} \right) * d_{k-1} + \left(\frac{S_x - \xi_{k-1}}{\xi_k - \xi_{k-1}} \right) * d_k$ (Interpolation linéaire)

fin si

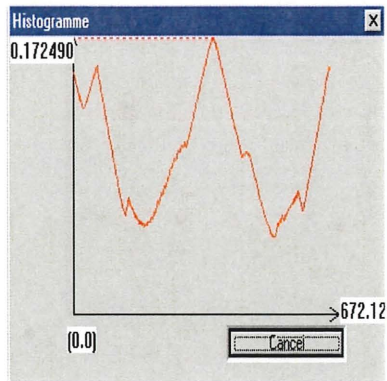
fin pour

fin pour

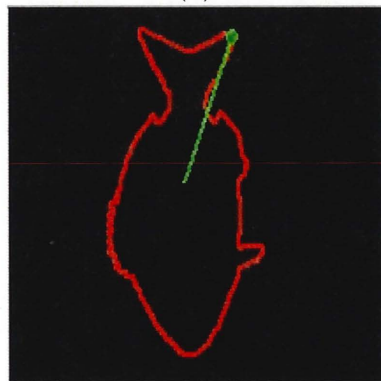
Grâce à cette interpolation, à chaque poisson de Surray de notre base de données on associe une image 1D supportée par 2^8 coefficients. Par conséquent, la recherche dans notre base de données de poissons est équivalente à rechercher l'image 1D de 2^8 coefficients qui lui est associée dans une base de données de 1000 images 1D ayant chacune 2^8 pixels. La Figure 4.1 montre un exemple d'un poisson de Surray de 582 pixels de notre base de données de poissons avec son histogramme et l'image 1D de 2^8 coefficients qu'on lui a associée après l'interpolation.



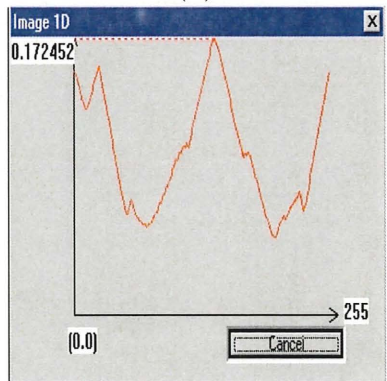
(a)



(b)



(a)

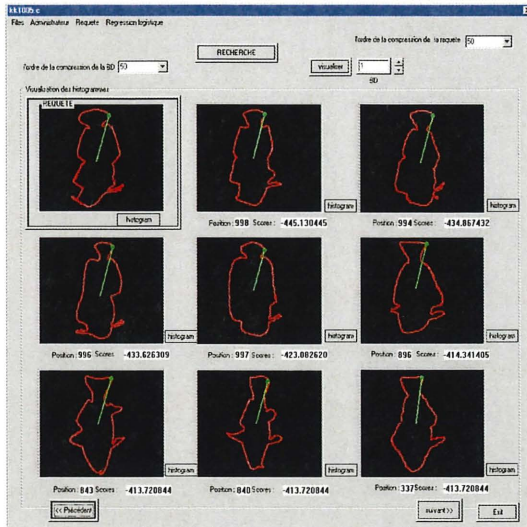


(c)

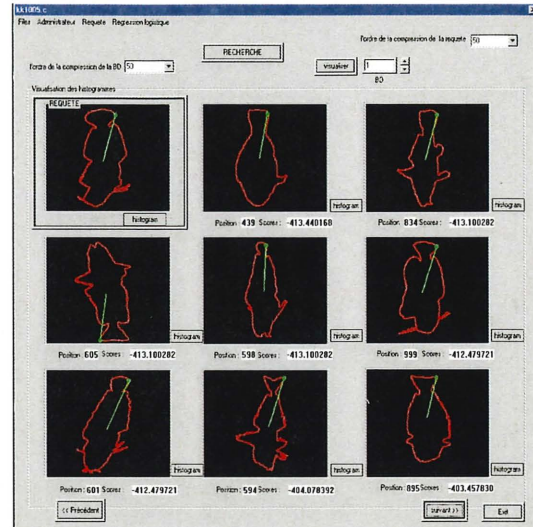
Figure 4.1 – Exemple : a) Un poisson de Surray de la BD. b) L’histogramme du poisson et c) L’image 1D de 2^8 pixels associée au poisson de Surray.

Dans le cas de la base de données de poissons de Surrey, à cause du terme $\tilde{w}_0|Q[0] - T[0]|$ de notre métrique, cette dernière discrimine les poissons de notre BD, ayant des tailles différentes à celle du poisson requête. Par conséquent, afin que notre recherche ne tienne compte de ces tailles, nous fixons \tilde{w}_0 à zéro avant l'échantillonnage.

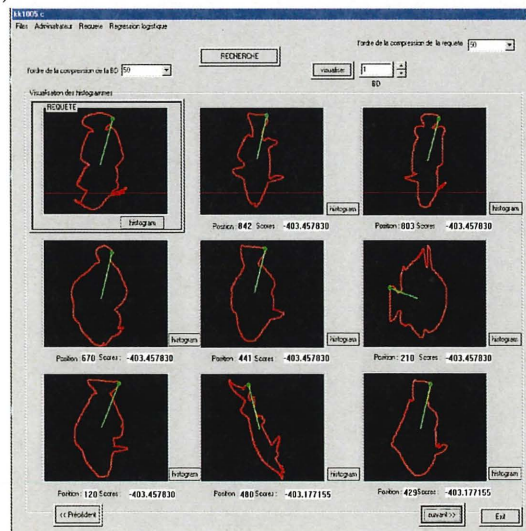
Les Figures 4.2 à 4.6 présentent des boîtes de dialogue générées par Visual C++ 6.0 donnant des exemples de résultats des recherches de plusieurs poissons requêtes dans notre base de données de poissons de Surrey, avec un ordre de compression $m = 50$. Chaque boîte de dialogue est sous forme d'une interface contenant neuf poissons de Surrey représentant respectivement le poisson requête et les poissons cibles les plus proches de ce dernier. Les poids calculés par la régression logistique et associés à notre métrique sont $\tilde{w}_0 = 0$, $w_0 = 12.54$, $w_1 = 11.77$, $w_2 = 9.18$, $w_3 = 11.16$, $w_4 = 10.88$, $w_5 = 10.26$, $w_6 = 0.09$ et $w_7 = 0$.



(a)

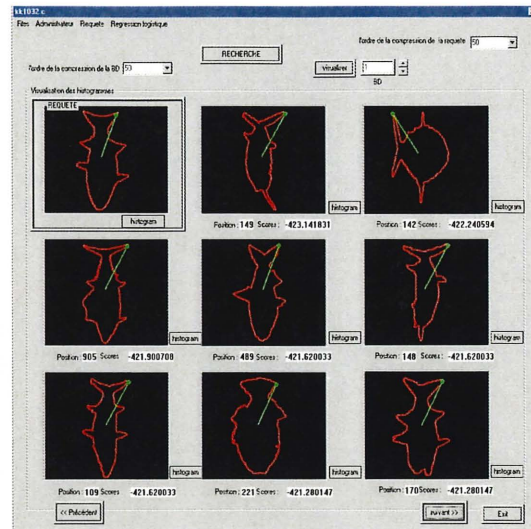
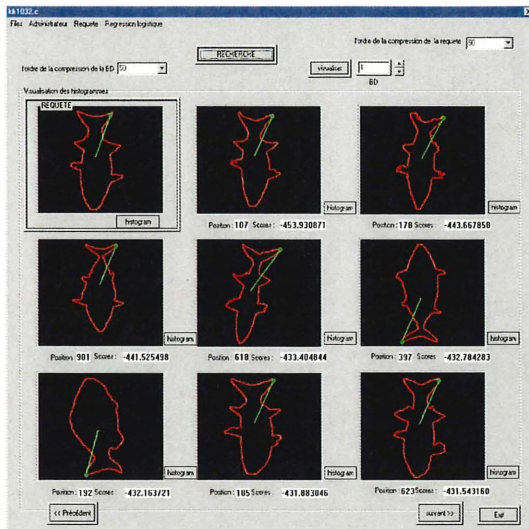


(b)



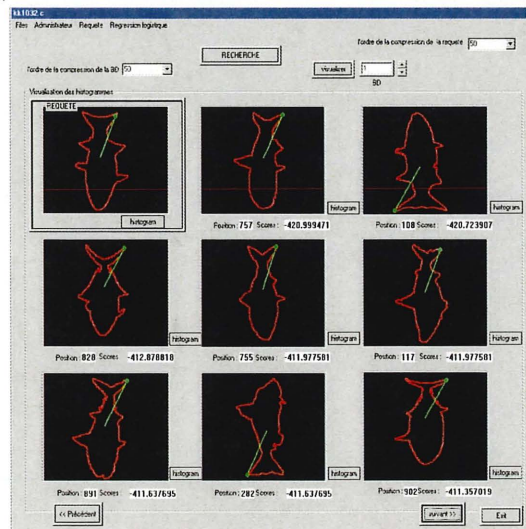
(c)

Figure 4.2 – Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



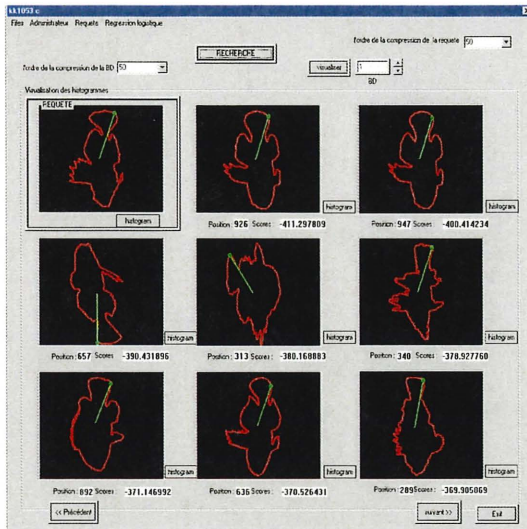
(a)

(b)



(c)

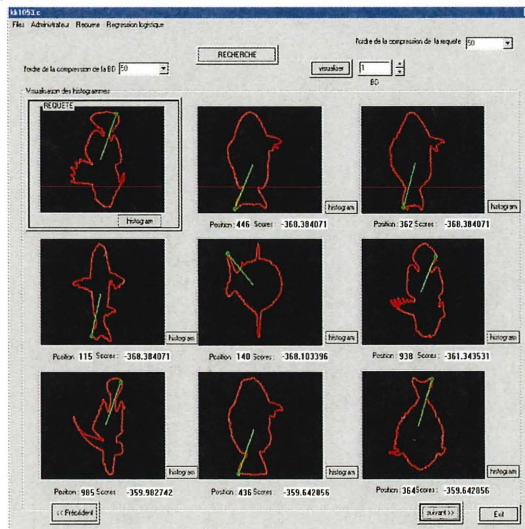
Figure 4.3 – Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



(a)

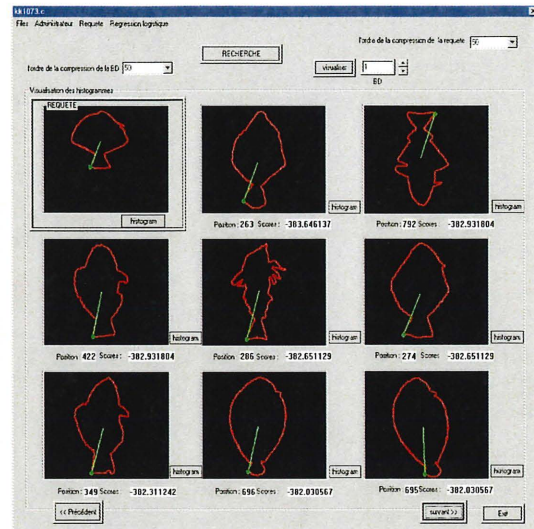


(b)



(c)

Figure 4.4 – Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



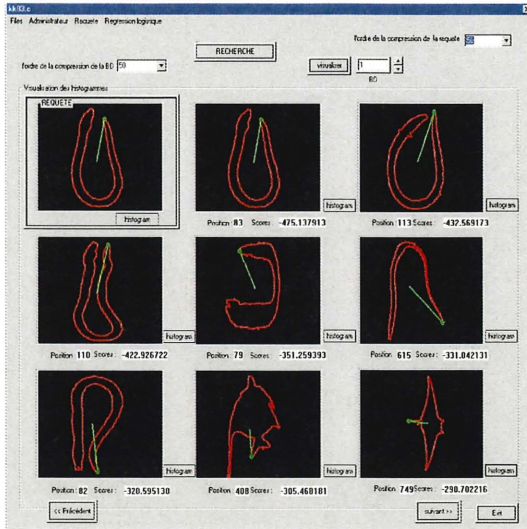
(a)

(b)



(c)

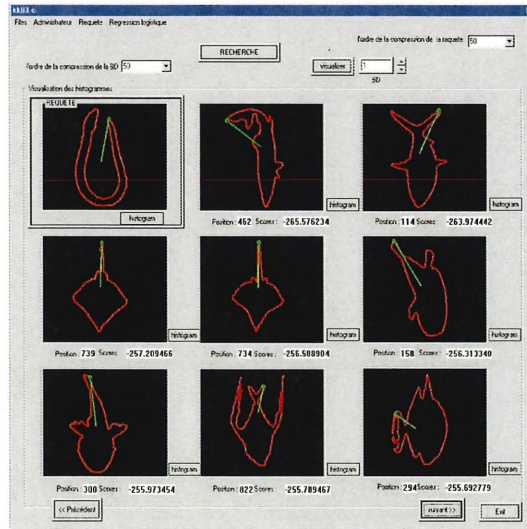
Figure 4.5 – Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



(a)



(b)



(c)

Figure 4.6 – Recherche avec compression à 50 coefficients. Les 24 poissons de Surray cibles les plus proches et ordonnés : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

4.2 Base de données d'images

Une image à niveaux de gris est une image dont la surface est divisée en éléments de tailles fixes appelés pixels, ayant chacun comme caractéristique un niveau de gris. Le niveau de gris est la valeur de l'intensité lumineuse en un point. L'intensité du pixel peut prendre des valeurs allant du noir au blanc en passant par un nombre fini de niveaux intermédiaires. Donc pour représenter une image à niveaux de gris, on peut attribuer à chaque pixel de l'image une valeur correspondant à la quantité de lumière renvoyée. Cette valeur est entre 0 et 255. Par conséquent, une image à niveau de gris peut être numériquement définie par une matrice bidimensionnelle $M \times N$ de valeurs $I(x,y) \in \{0, \dots, 255\}$ où $M \times N$ est la taille de l'image, x et y sont les coordonnées cartésiennes d'un pixel de l'image et $I(x,y)$ est le niveau de gris en ce pixel.

Dans le domaine de la recherche d'images, plusieurs travaux dans les secteurs académiques et industriels, ont porté sur la description des niveaux de gris d'une image [13]. L'histogramme des niveaux de gris d'une image, qui est une fonction donnant la fréquence d'apparition de chaque niveau de gris dans l'image, est le plus souvent utilisé pour la description d'images à niveaux de gris. Cependant, cet histogramme présente l'inconvénient de perdre toute information spatiale sur les niveaux de gris et de rendre compte que de leur distribution globale dans l'image. D'après [10], des études de la perception ont montré que les contours et les coins d'une image à niveaux de gris sont riches en information. Il est d'usage courant d'utiliser le laplacien d'une image pour capturer cette information. Par conséquent, un histogramme pondéré par le laplacien est un meilleur outil dans ce contexte pour la description d'une image à niveaux de gris [10]. Pour effectuer la recherche d'une image requête dans notre base de données d'images, nous décrivons chaque image par un histogramme pondéré par le laplacien. Étant donnée une image I de taille $M \times N$ de notre base de données, l'espace de ses niveaux de gris est échantillonné d'une

manière régulière pour constituer les éléments de l'histogramme pondéré par le laplacien $h(c)$ qui s'écrit:

$$h(c) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta(I(i,j) - c) \Delta I(i,j)^2 \quad \text{pour tout } c \in \{0, \dots, 255\}, \quad (4.1)$$

où δ est le symbole de Kroenecker en 0, c'est-à-dire

$$\delta(I(i,j) - c) = \begin{cases} 1 & \text{si } I(i,j) = c, \\ 0 & \text{sinon,} \end{cases} \quad (4.2)$$

et $\Delta(i,j)$ est le laplacien au pixel de coordonnées cartésiennes i et j , donné par le produit de convolution entre le laplacien d'un masque gaussien G et l'image I dans le cas discret, c'est-à-dire

$$\Delta I(i,j) = (I * \Delta G)(i,j) = \frac{1}{M_G^2} \sum_{k=-\frac{M_G}{2}}^{\frac{M_G}{2}} \sum_{l=-\frac{M_G}{2}}^{\frac{M_G}{2}} I(i-k, j-l) \Delta G(k,l), \quad (4.3)$$

avec $\frac{M_G}{2} \leq i < M - \frac{M_G}{2}$, $\frac{M_G}{2} \leq j < N - \frac{M_G}{2}$ et $\Delta G(x,y) = \frac{\partial^2 G(x,y)}{\partial^2 x} + \frac{\partial^2 G(x,y)}{\partial^2 y}$ est le laplacien du masque gaussien $G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$ de variance σ et de taille M_G . Puisque le laplacien est nul au niveau des contours, l'histogramme pondéré par le laplacien distingue les pixels aux voisinages des contours d'une image.

Une image couleur est une image dont chaque couleur est décomposée dans ses trois composantes primaires de base (rouge, vert et bleu). Par conséquent, chaque pixel ou point de coordonnées cartésiennes x et y d'une image couleur se retrouve associé à trois entiers $R(x,y)$, $G(x,y)$ et $B(x,y)$ dont les valeurs sont entre 0 et 255, représentant respectivement les intensités lumineuses des couleurs primaires rouge, vert et bleu en ce pixel. Dans le cadre de notre travail, nous pouvons effectuer notre recherche dans une base de données d'images couleurs, en convertissant chaque image couleur en une images à niveaux de gris. Afin de convertir une image couleur en image à niveaux de gris, nous associons à chaque niveau de gris de cette dernière la valeur $I(x,y) =$

$[0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y)]$. Notre base de données d'images contient 1000 images couleurs converties en images à niveaux de gris.

La Figure 4.7 montre un exemple d'une image couleur de notre base de données d'images, avec sa version convertie en image à niveaux de gris et son histogramme pondéré par le laplacien.

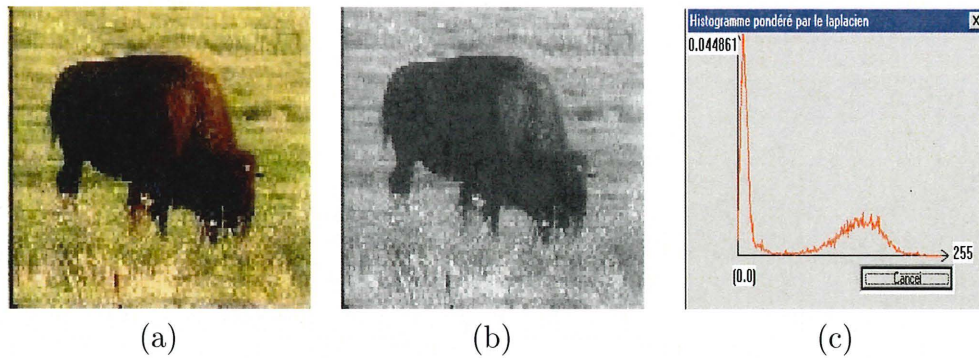
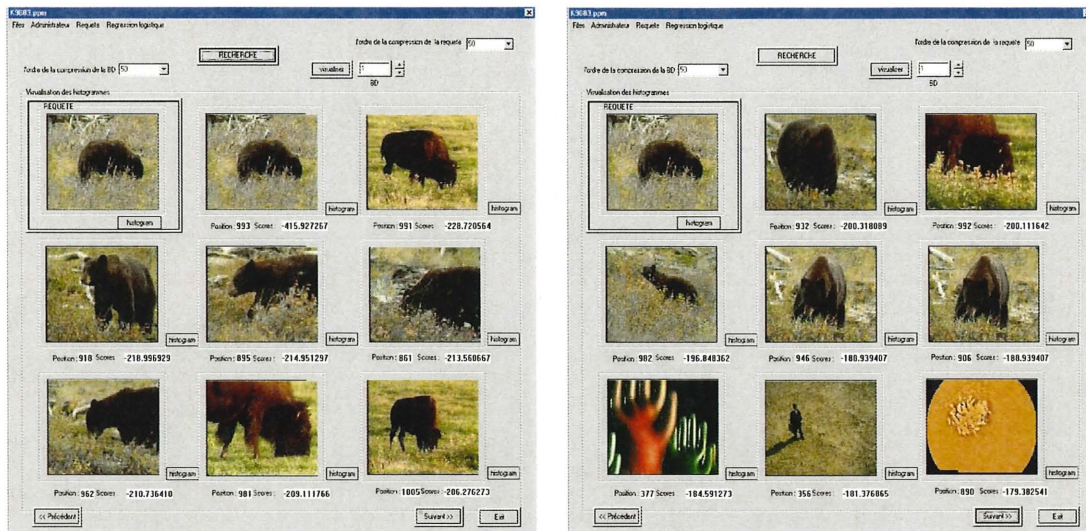


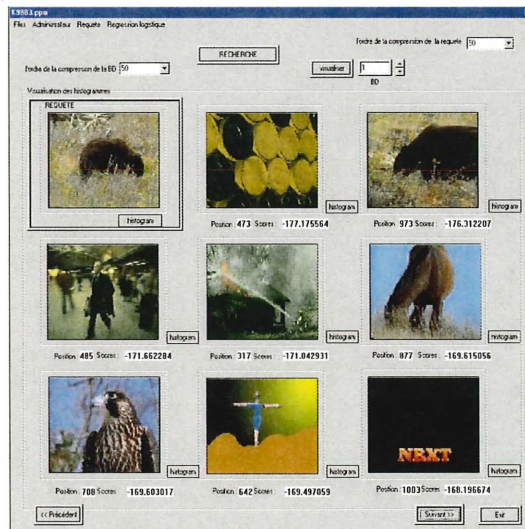
Figure 4.7 – Exemple : a) Une image couleurs de la BD. b) La version convertie de l'image couleurs en une image à niveaux de gris. c) L'histogramme pondéré par le laplacien de l'image.

Grâce à la description de chaque image de notre base de données par un histogramme pondéré par le laplacien, la recherche d'une image couleur requête dans notre base de données d'images correspond à la recherche de l'histogramme pondéré par le laplacien de sa version convertie en image à niveaux de gris, dans une base de données contenant 1000 histogrammes. Les Figures 4.8 à 4.12 présentent des boîtes de dialogue générées par Visual C++ 6.0 donnant des exemples de résultats des recherches de plusieurs images couleurs requêtes dans notre base de données d'images couleurs, avec un ordre de compression $m = 50$. Chaque boîte de dialogue est sous forme d'une interface contenant neuf images couleurs représentant respectivement l'image requête et les images cibles les plus proches de cette dernière. Les poids associés à notre métrique sont $\tilde{w}_0 = 10.25$, $w_0 = 10.01$, $w_1 = 9.74$, $w_2 = 10.16$, $w_3 = 9.02$, $w_4 = 9.03$, $w_5 = 2.68$, $w_6 = 9.70$ et $w_7 = 10.84$.



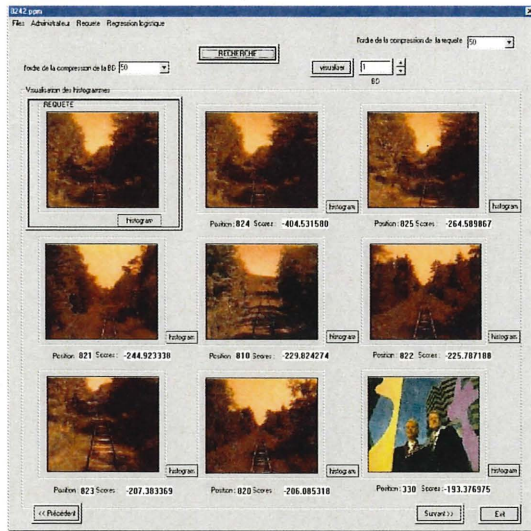
(a)

(b)

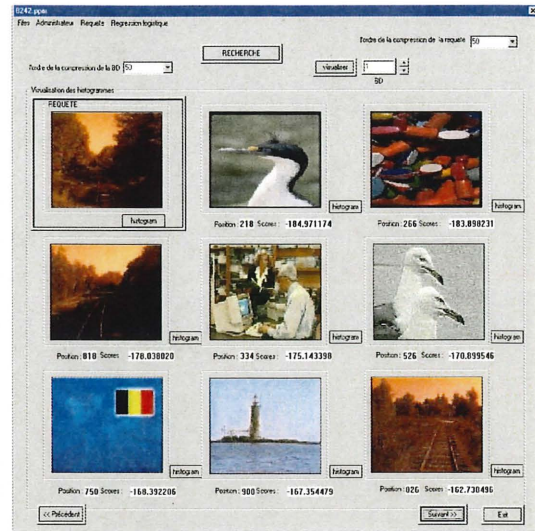


(c)

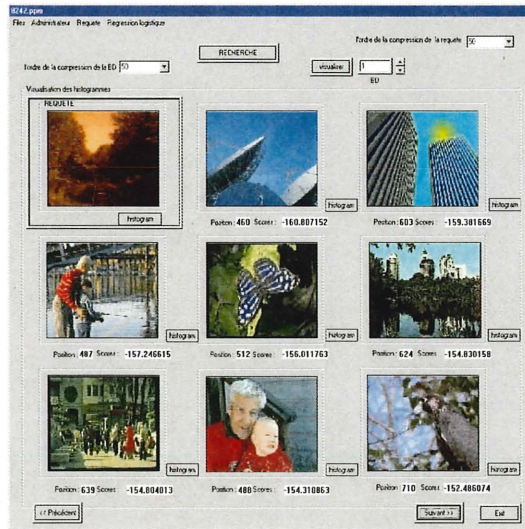
Figure 4.8 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



(a)

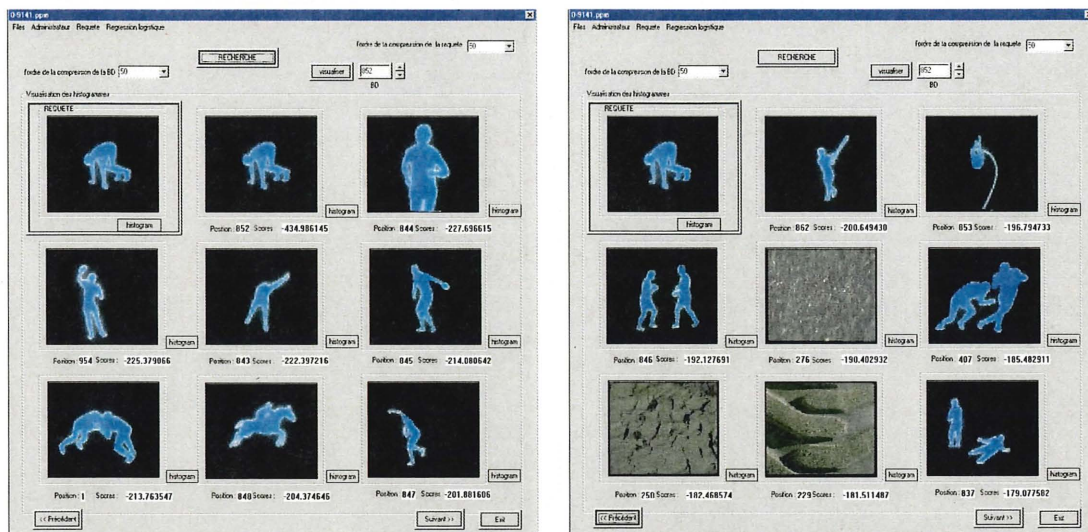


(b)



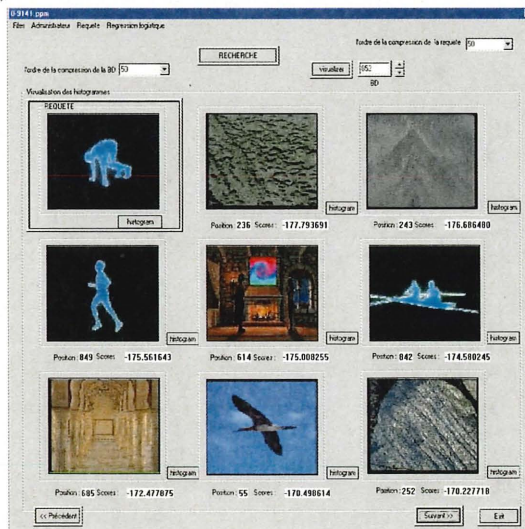
(c)

Figure 4.9 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16. et c) les cibles de 17 à 24.



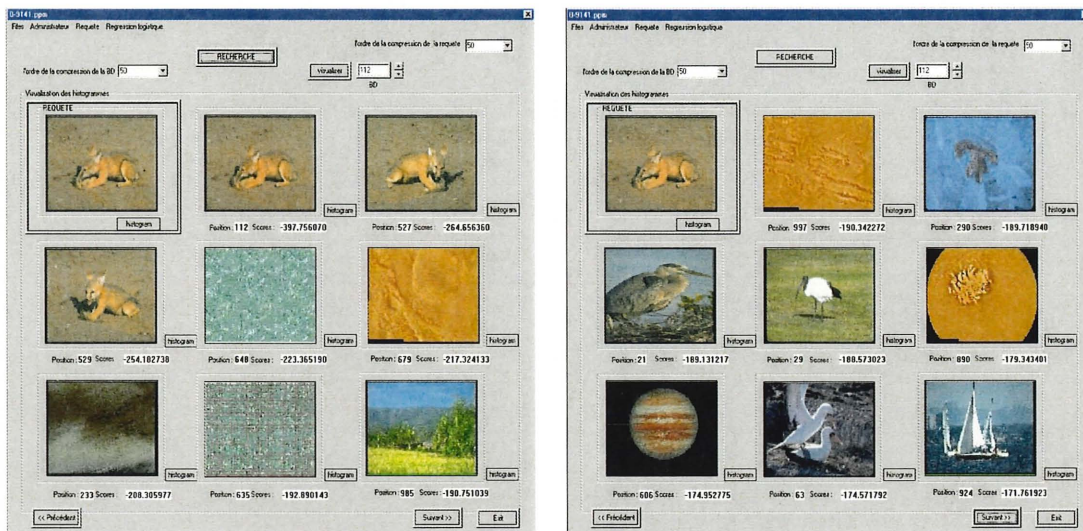
(a)

(b)



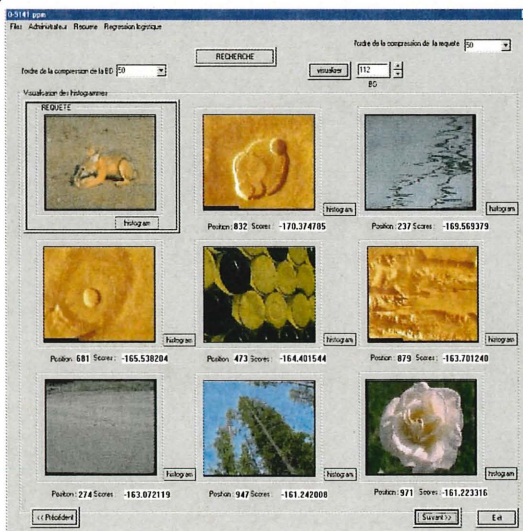
(c)

Figure 4.10 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



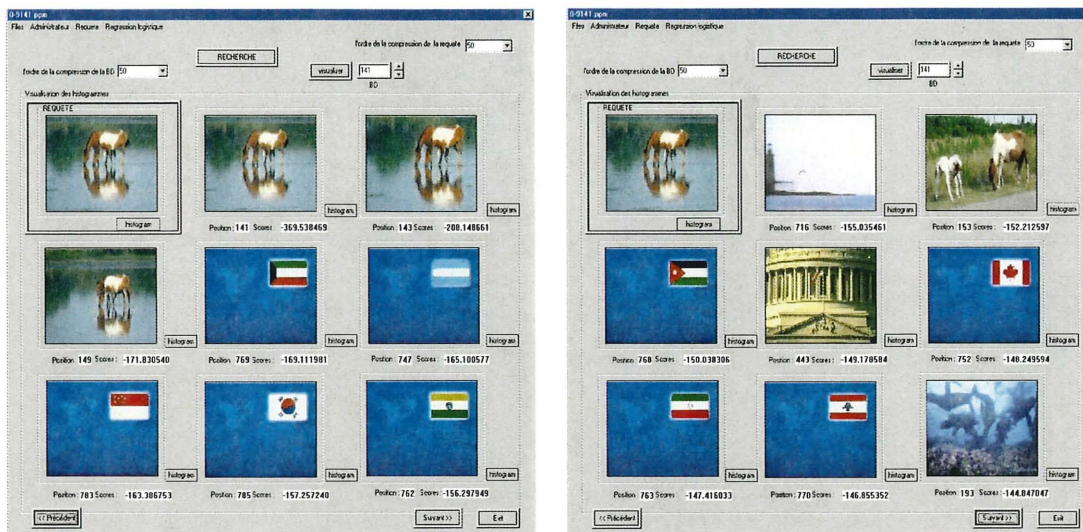
(a)

(b)



(c)

Figure 4.11 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



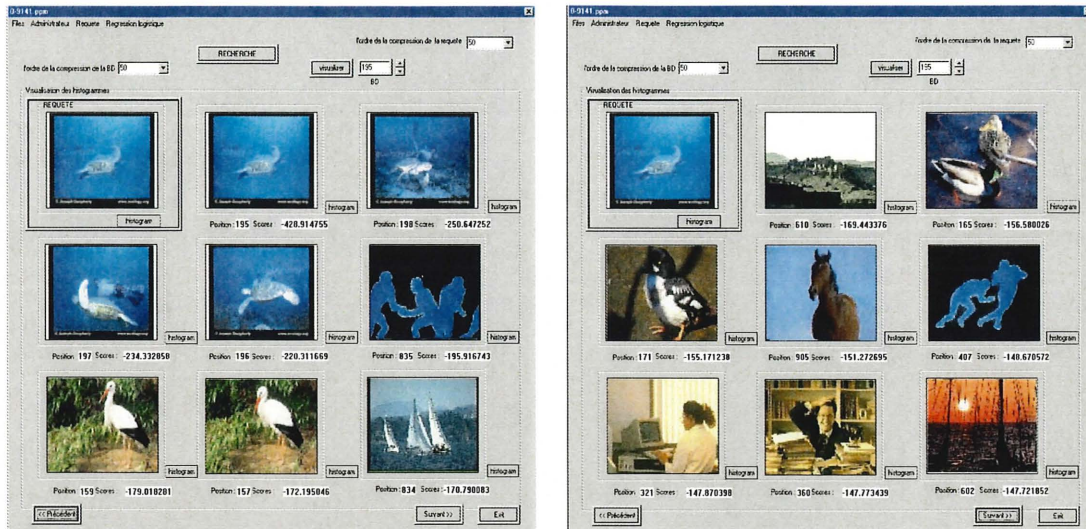
(a)

(b)



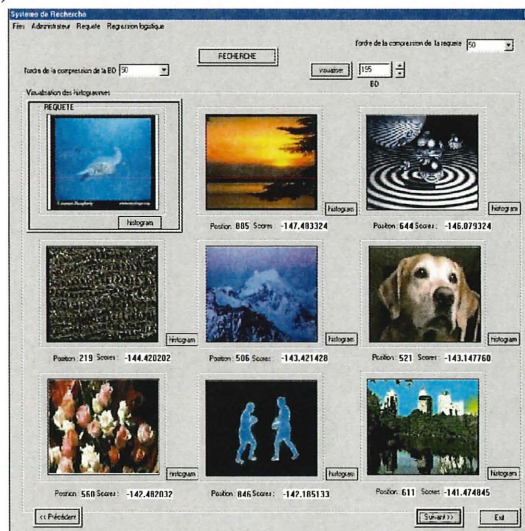
(c)

Figure 4.12 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.



(a)

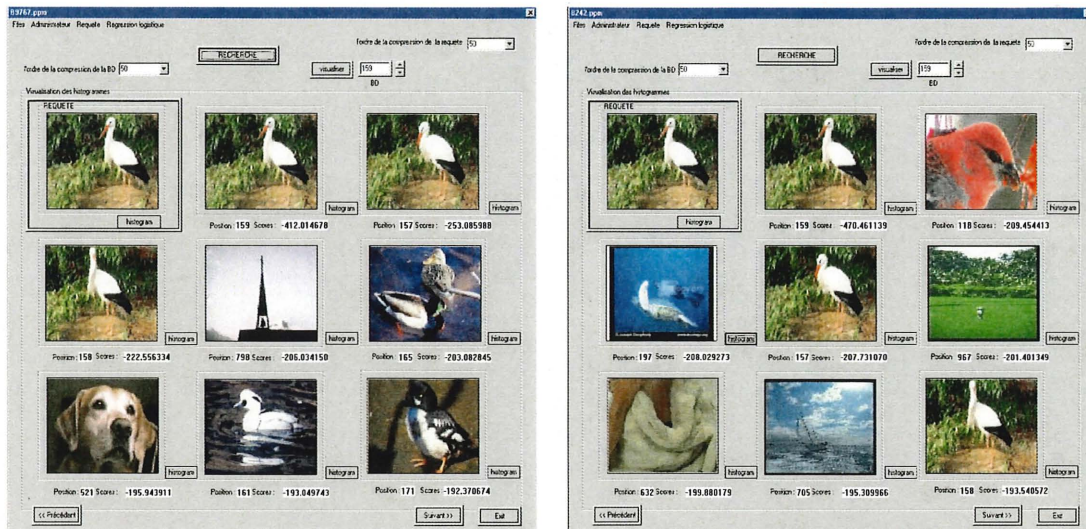
(b)



(c)

Figure 4.13 – Recherche avec compression à 50 coefficients. Les 24 images couleurs cibles les plus proches et ordonnées : a) les cibles de 1 à 8 , b) les cibles de 9 à 16 et c) les cibles de 17 à 24.

La Figure 4.13 montre la différence entre une recherche dans une base de données d'images décrites par des histogrammes à niveaux de gris, et une autre recherche dans une base de données d'images décrites par des histogrammes pondérés par le laplacien.



(a)

(b)

Figure 4.14 – Comparaison: a) Recherche basée sur les histogrammes pondérés par le laplacien. b) Recherche basée sur les histogrammes à niveaux de gris.

CONCLUSION

Dans ce mémoire, nous avons étudié les ondelettes de Haar et nous les avons appliquées dans la recherche d'images dans des bases de données d'images 1D et d'images 2D. La méthodologie de recherche décrite dans le cadre de notre travail est efficace et rapide et nécessite seulement une petite quantité d'information stockée dans chaque image cible de la BD. Grâce à une base de fonctions 1D représentant des caractéristiques des images 2D, nous avons pu effectuer avec succès de nombreuses recherches dans une base d'images couleurs 2D et une base de poissons de Surray. Néanmoins, notons que notre méthode de recherche est sensible à la translation de la requête et que son efficacité dépend de l'ordre de compression de la requête et des images de la BD. De plus, la décomposition en ondelettes de Haar est une opération élémentaire et la base de Haar est la plus simple base d'ondelettes.

Dans les travaux futurs, nous désirons obtenir de meilleurs résultats de recherche à l'aide d'une décomposition des images 1D de notre BD en ondelettes BSplines qui est une décomposition beaucoup moins simpliste que celle en ondelettes de Haar [3]. Ensuite, nous prévoyons améliorer l'efficacité de l'algorithme de recherche, en rendant ce dernier robuste à la translation. Enfin, dans l'application de notre méthode de recherche pour une base de données d'images couleurs 2D, chaque image est représentée par un histogramme pondéré par le laplacien qui ajoute une information sur les pixels aux voisinages

des contours l'image. D'après [10], il est possible de représenter chaque image couleur par un histogramme cumulé qui ajoute une information sur la répartition géométrique des couleurs. En résumé, l'image est subdivisée en fenêtres, l'historgramme de couleurs usuel est calculé sur ces fenêtres. Par conséquent, le cumul multiplicatif ou additif des histogrammes usuels locaux donnera une information sur la dispersion des couleurs dans l'image.

Bibliographie

- [1] E. Aitnouri, F.Dubeau, S.Wang et D.Ziou. "Controlling Mixture Component Overlap for Clustering Algorithms Evaluation". *Pattern Recognition and Image Analysis*, Vol. 12, No. 4, 2002, pp. 331-346.

- [2] D.H. Salesin, C.E. Jacob et A. Finkelstein. "Fast Multiresolution Image Quering". *Proceedings of SIGGRAPH'95*, 277-286, New York, 1995. ACM.

- [3] D.H. Salesin, E.J. Stollnitz et T.D. Deroose. "Wavelets for Computer Graphics, Theory and Applications". Morgan Kaufmann Pub, San Francisco, 1996.

- [4] S.S. Soliman et M.D. Srinath. "Continuous and Discrete Signals and Systems" (2nd. Ed.). Prentice Hall, 1998.

- [5] K. Hirata et T. Kato. "Query by Visual Example - Content Based Image Retrieval". *Advances in Database Technology (EDBT'92)* A. Pirotte, C. Delobel, and G.Gottlob, Eds, 56-71, Springer-Verlag, Berlin, 1992. ACM.

- [6] E. Quak et N. Weyrich. "Decomposition and Reconstruction Algorithms for Spline Wavelets on the Bounded Interval". *Applied and Computational Harmonic*

Analysis, 1(3):217-231, June 1994.

- [7] S. Mallat. "Multiresolution Representation and Wavelets". Ph.D. Thesis, University of Pennsylvania, 1988.
- [8] Thomas S. Huang and Yong Rui, "Image retrieval: Past, present and future". *in Proc. of Int. Symposium on Multimedia Information Processing*, Taipei, Taiwan, Dec 1997.
- [9] K. Aas et L. Eikvil, "A Survey on Content-based Access to Image and Video Databases", Technical Report N. 915, Norwegian Computing Center, Oslo, Mar. 1997.
- [10] N. Boujemaa, S. Boughorbel et C. Vertan, "Description de la répartition spatiale de la couleur pour l'Indexation d'Images", *Reconnaissance des Formes et Intelligence Artificielle*, vol. 2 pp 405-414, 2002.
- [11] William H. Press, Brian P. Flannery, Saul A. Teukolsky et William T. Fetterling. "Numerical Recipes" (2nd. Ed.). Cambridge University Press, 1992.
- [12] F. Quian, M. Li, I. Zhang, H. Zhang et B. Zhang, "Gaussian Mixture Model for Relevance Feedback in Image Retrieval", *IEEE International Conference on Multimedia and Expo*, August 26-29, 2002, Lausanne, Switzerland.
- [13] M. Swain et D. Ballard, "Color Indexing", *International Journal of Computer Vision*, 7(1):11-32, 1991., 1991.