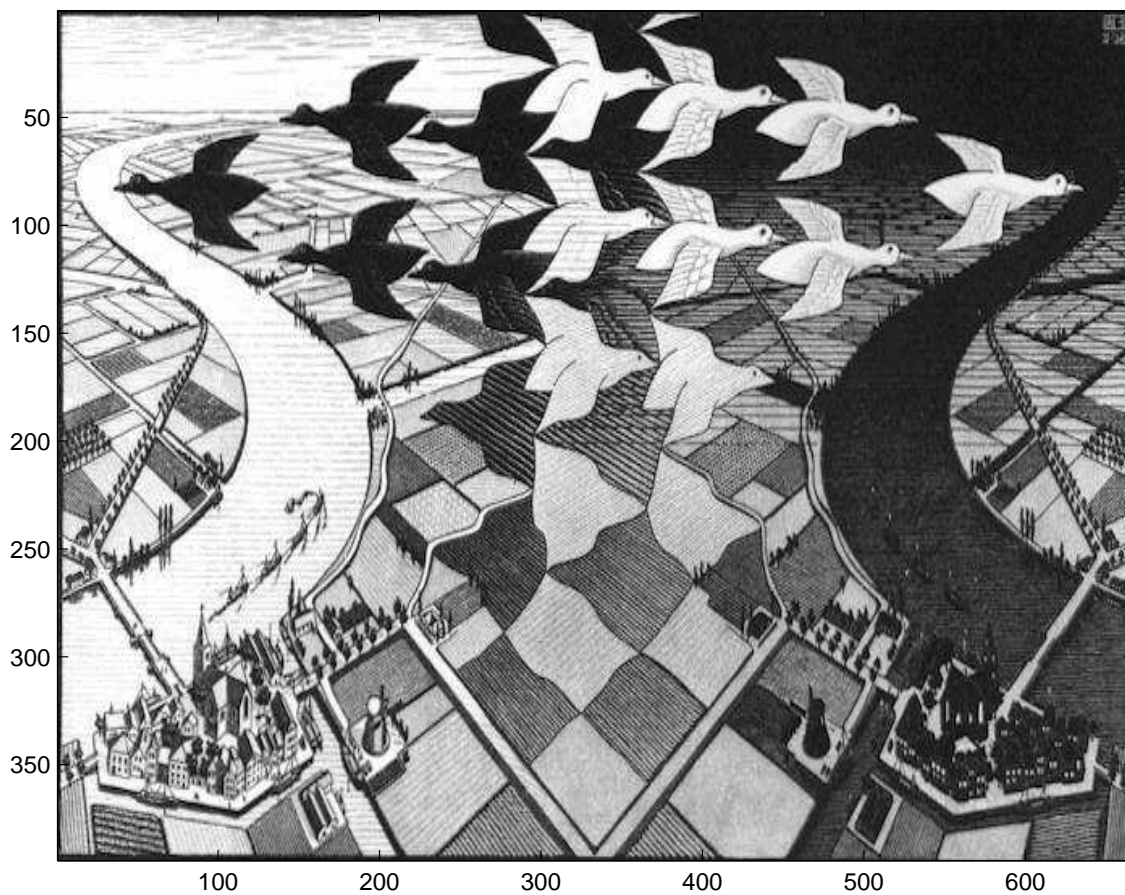


*COMPTE-RENDU DE T.P.:
COMPRESSION D'IMAGES PAR CODAGE EN
SOUS-BANDES*

Eléonore GRAVIER
Céline CATTOEN

novembre 2002



INTRODUCTION

L'objet de ce T.P. est l'étude de la compression d'images par codage en sous-bandes, réalisé par un banc de filtres discrets à reconstruction parfaite. Dans un premier temps on observera l'effet sur une image de divers filtres. On réalisera ensuite un banc de filtres "à reconstruction parfaite", et enfin on étudiera la compression de données d'images à partir de ce banc de filtres.

1 Séance d'introduction

Différents filtrages vont être appliqués sur une image (escher.mat) dont les résultats seront commentés par la suite.

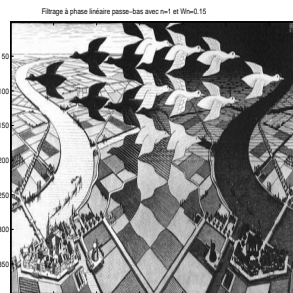
1.1 Filtrage passe-bas à phase linéaire :

On utilise un filtre à réponse impulsionnelle finie (filtre RIF) en particulier un filtre passe-bas à phase linéaire. La fonction de transfert en z s'écrit dans ce cas $H_z(z) = P(z^{-1})$ contre $H_z(z) = \frac{A(z^{-1})}{B(z^{-1})}$ dans le cas général. Ce filtre est caractérisé par 2 paramètres :

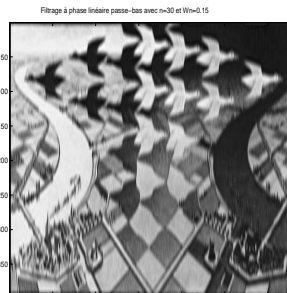
- n , le degré du polynôme H (transformée en z du filtre).
- Wn , fréquence de coupure du filtre.

A la sortie, on obtient un retard par rapport au signal d'entrée, ce qui se traduit par une perte d'informations sauf si l'on ajoute $\frac{n}{2}$ zéros à la fin de la matrice des données, que l'on enlève ensuite au début de la matrice une fois le filtrage effectué.

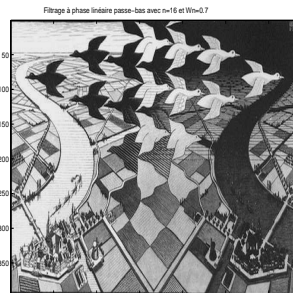
Visualisons ce filtrage pour les situations suivantes :



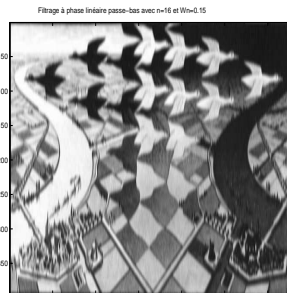
$n = 1$ et $W_n = 0.15$



$n = 30$ et $W_n = 0.15$



$n = 16$ et $W_n = 0.7$



$n = 16$ et $W_n = 0.15$

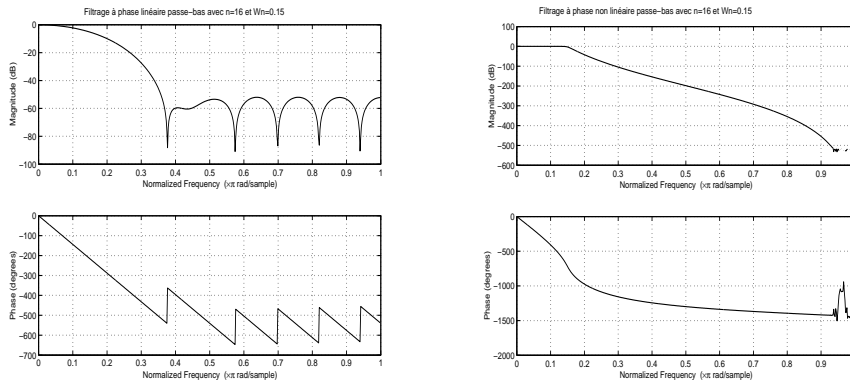
Résultats :

- A Wn constant, on constate que plus n est grand, plus le retard est important (visualisation lors du TP) et plus l'image est floue, ce qui est normal, car les hautes fréquences sont mieux éliminées (ce filtre se rapproche du filtre passe-bas linéaire idéal). En effet, les hautes fréquences codent le contraste (fort gradient de couleur) et on obtient alors une image filtrée plus floue, aux contours plus estompés.
- A n constant, plus Wn est petite, plus l'image est floue, car le filtre laisse passer moins de données en haute fréquence.

1.2 Filtrage passe-bas à phase non linéaire :

On utilise à présent un filtre à réponse impulsionnelle infinie (filtre RII), puisque $a(z^{-1}) \neq cste$.

Résultats : On observe les mêmes résultats qu'avec le filtre précédent, mais les images sont floues plus rapidement, car les hautes fréquences sont mieux supprimées. On peut bien comprendre ce phénomène en visualisant les caractéristiques des filtres employés :



linéaire avec $n = 1$ et $W_n = 0.15$ non linéaire avec $n = 30$ et $W_n = 0.15$

1.3 Filtrage passe-haut :

Résultats : Le filtrage passe-haut ne conserve que les hautes fréquences qui définissent le contraste. Cependant, les basses fréquences étant supprimées, la qualité des couleurs est perdue surtout sur les zones monochromes. Après un filtrage passe-haut, on obtient une image nette mais beaucoup plus grise et uniforme au niveau des couleurs. En effet, les valeurs ne sont plus comprises entre 0 et 255 (intervalle habituel des valeurs pour les images). Le signal est centré sur 0 (ce qui donne une image sombre, les valeurs ≤ 0 codant pour le noir). On contraste l'image en mettant ses valeurs soit à 0 (noir), soit à 255 (blanc).

1.4 Décimation/interpolation puis filtrage passe-bas par un filtre à phase linéaire :

La décimation-interpolation consiste à remplacer une valeur sur deux par 0 puis à filtrer par le même filtre qu'en 1.1.

Résultats : L'image obtenue est un peu plus floue que l'originale mais ce procédé permet d'économiser la moitié de la place en mémoire de l'image pour le stockage après la décimation, pour obtenir ensuite une image d'assez bonne qualité après interpolation puis filtrage. L'image donnée par la décimation est très noire car une valeur sur deux vaut zéro, mais le filtrage permet de lisser ces irrégularités.

2 Réalisation d'un banc de filtres à reconstruction parfaite

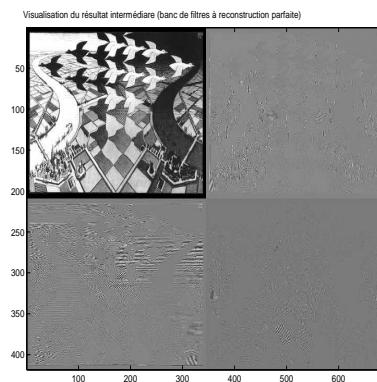
On applique ici un banc de filtres à reconstruction parfaite à l' image escher.mat. A la sortie, on veut obtenir une image identique, à un facteur de retard près. L'intérêt du codage en sous-bandes, lorsque le procédé est itéré est de faciliter la compression d'images.

2.1 Principe du banc de filtres à reconstruction parfaite :

Le filtre que étudié est constitué de deux parties :

2.1.1 Décomposition de l'image en ses composantes fréquentielles et compression de celles-ci :

Les lignes de la matrice du signal d'entrée x sont tout d'abord filtrées avec deux filtres RIF : un passe-bas (h_0) et l'autre passe-haut (h_1). On obtient ainsi deux signaux : x_0 et x_1 . On décime ensuite leurs lignes d'un facteur 2 puis on filtre leurs colonnes avec h_0 et h_1 avant de décimer leurs colonnes d'un facteur 2. On obtient finalement 4 images notées $x_{00}, x_{01}, x_{10}, x_{11}$. Chacune de ces quatre sous-images a subi deux décimations : l'espace mémoire occupé par le stockage de ces images est le même que celui occupé par l'image initiale.



Visualisation du résultat intermédiaire

- Etude de x_{00} :

C'est l'image qui résulte de deux filtrages passe-bas (lignes et colonnes) de l'image initiale. Cette image est plus floue que celle d'origine car ces deux filtrages ont conservé les basses fréquences, c'est à dire les couleurs de grandes zones monochromes et les uniformités de nuances. Les contrastes ont quant à eux été perdus, par suppression des hautes fréquences. Ils ont cependant été conservés dans les trois autres images qui ont subi un filtrage passe-haut. Cette image est celle qui conserve le plus d'informations.

Les trois images que nous allons étudier ont subi un filtre passe-haut et ont donc une moyenne nulle.

- Etude de x_{01} :

Elle résulte d'un filtrage passe-bas sur les lignes et d'un filtrage passe-haut sur les colonnes. Elle conserve donc les contrastes quand on parcourt verticalement l'image. On peut remarquer par exemple que les lignes horizontales en haut à droite de l'image ont été gardées.

- Etude de x_{10} :

Elle résulte d'un filtrage passe-haut sur les lignes et d'un filtrage passe-bas sur les colonnes. Le contraste est donc ici conservé quand on parcourt horizontalement l'image. Ici, on ne peut pas citer d'exemples précis.

- Etude de x_{11} :

Cette image correspond à deux filtrages passe-haut. Elle conserve notamment les contrastes que l'on a perdu dans les autres images. Ceci peut se voir avec le quadrillage de la figure, en bas au centre, qui ressort.

2.1.2 Reconstruction de l'image :

Elle consiste en une interpolation d'un facteur 2 sur les colonnes puis en un filtrage des colonnes par g_0 et g_1 avant d'interpoler les lignes d'un facteur 2 et de les filtrer avec g_0 et g_1 . On obtient alors une reconstruction parfaite. On note cependant un retard k dû aux différents filtrages calculé avec $conv(h_0, g_0) + conv(h_1, g_1)$. On ajoute donc k zéros en fin de lignes et en fin de colonnes avant de faire passer l'image dans le banc de filtres.

2.2 Intérêt pour la compression d'images :

Juste avant la reconstruction de l'image, on n'a rien compressé car l'information contenue dans les quatre sous-images est égale à celle contenue dans l'image initiale. Pour compresser, deux méthodes s'offrent à nous :
-On ne garde que x_{00} qui contient le maximum d'information. Cependant, l'image est de très mauvaise qualité et floue. On préférera donc la seconde méthode.

-Les trois sous-images filtrées au moins une fois par un filtre passe-haut sont très uniformisées au niveau des couleurs puisqu'elles sont grises. Certaines valeurs reviennent donc très souvent et permettent donc une compression avec gzip qui est meilleure car gzip utilise l'algorithme de Huffman pour compresser. Plus les valeurs sont fréquentes et plus la compression est bonne. On peut ainsi obtenir des taux de compression allant jusqu'à 100.

Remarque 1 :

Les matrices de ces images intermédiaires comportent des zéros sur les bords, que l'on a introduits pour compenser le retard. Ceci explique l'apparition de bandes noires sur les bords des quatre images. Ces zéros seront supprimés avant la visualisation finale.

Remarque 2 :

Les trois sous-images ayant subies un filtrage passe-haut sont de moyenne nulle : il faut avant visualisation recalculer leurs valeurs entre 0 et 255 par une transformation affine.

3 Compression :

3.1 Principe de la méthode :

La première partie du travail consiste à effectuer une analyse fréquentielle de l'image de manière itérative. Chaque analyse de niveau $n + 1$ porte sur l'image x_{00} du niveau n , c'est à dire l'image reflétant les basses fréquences. Ensuite, on effectue une synthèse à partir des $3n + 1$ images obtenues.

La deuxième partie, qui est plus empirique, consiste à compresser chacune des images intermédiaires obtenues par un taux de compression q_i propre. Ainsi, une fois chaque analyse effectuée, on "quantifie" chaque pixel de chaque sous-image x_{ij}^k . Plus q_i^k est élevé et plus un grand nombre de valeurs de la sous-image est ramené à 0. Comme x_{00}^k est l'image qui contient le plus d'information il est souhaitable de ne pas la compresser.

L'objectif est d'obtenir un taux de compression total de l'ordre de 10 tout en conservant la meilleure qualité possible pour l'image finale.

Tout d'abord, pour $(q_i = 1)_{i=1, \dots, n}$, on constate que l'image obtenue après 4 analyses et autant de synthèses est identique à l'image d'origine. Ce résultat semble logique puisque le procédé n'est rien d'autre qu'une itération de l'analyse fréquentielle avec reconstruction parfaite donnée en (2). Pour évaluer la performance de la compression, on dispose de trois critères d'évaluation des informations perdues :

- le critère visuel, qui permet d'apprécier les écarts importants de définition.
- l'erreur en norme ∞ $\| \cdot \|_{\infty}$, qui mesure l'écart maximum entre les valeurs de la matrice de départ et la matrice d'arrivée.

- l'erreur en norme 1 $\|\cdot\|_1$, qui mesure la somme des écarts en valeur absolue entre les valeurs des 2 matrices.

Le troisième critère d'évaluation rend le mieux compte de la qualité de compression atteinte, le deuxième portant sur une seule valeur qui peut être isolée.

Pour déterminer les q_i , on procède de la manière suivante :

Tout d'abord, l'analyse fréquentielle de niveau $n + 1$ codant pour des plus basses fréquences que celle de niveau n , les q_i de niveau $n + 1$ seront moins élevés que ceux de niveau n . De même les q_i pour les images x_{01} et x_{10} devront être plus petits que celui de x_{11} du même niveau. En effet, il vaut mieux garder plus d'informations dans le domaine des basses fréquences, car la représentation de l'image telle que l'on peut la percevoir, a plus de composantes en basses fréquences.

3.2 Compression :

Par manipulations successives sur l'image lenar.mat, on arrive à la matrice de compression suivante :

$$Q = \begin{pmatrix} 1 & 40 & 40 & 200 \\ 1 & 25 & 25 & 60 \\ 1 & 15 & 15 & 60 \\ 1 & 10 & 10 & 30 \end{pmatrix}$$



Résultats : On trouve après analyse, compression avec gzip, décompression puis synthèse l'image présentée ci-dessus. L'image est un petit peu plus floue mais reste acceptable.

Les erreurs présentées dans l'analyse du problème sont pour l'image lenar.mat :

- $erreur_\infty = 45.4075$
- $erreur_1 = 4.6047$

Les erreurs sont assez faibles et confortent la qualité visuelle de l'image.

Concernant la compression, après avoir compresser les matrices lenar.mat et image_analysee.mat, on trouve comme taille initiale 79926 bytes pour lenar.mat.gz et comme taille finale 7876 octets pour image_analysee.mat.gz .

Le taux de compression est donc $\tau = 10.148$, c'est à dire très proche de celui demandé de 10.

CONCLUSION

La méthode de compression consistant à quantifier chaque pixel après analyse de l'image est beaucoup plus efficace que la méthode de compression de gzip seule. On pourrait étudier ces applications dans d'autres domaines tels que les fichiers codant pour la musique et les comparer à la compression des formats MP3.