

# Traitement de Signal & Image

VIEVILLE Sébastien - MARTELLI Gino - CIMINO Théo

13 Janvier 2025

## Abstract

Ce document présente un algorithme de compression et décompression d'image type JPG ainsi qu'une méthode de filtrage. Ces algorithmes présentent un grand intérêt car ils permettent stocker des images initialement lourdes en format compressé. Ces formats compressés réduisent considérablement l'espace nécessaire au stockage de l'image. Cependant, Le processus étudié ici ne se fait pas sans introduire d'erreur. Le but étant d'obtenir le meilleur compromis entre le taux d'erreur et le taux de compression.

## 1 Introduction

Une image numérique en 2 dimensions  $n_x * n_y$  peut être représentée comme un tableau tri-dimensionnel de taille  $n_x * n_y * 3$ . Les entrées de cette matrice correspondent aux intensités lumineuses de chaque pixel dans chacun des trois canaux de couleur (rouge, vert et bleu). L'idée derrière la compression d'images avec pertes est de garder uniquement les informations importantes (visibles) de l'image.

Ces informations se trouvent aisément en passant notre image dans le domaine fréquentiel, ce sont les basses fréquences. En effet, elles représentent des variations lentes de l'intensité des couleurs de l'image, l'œil humain y est plus sensible. De plus, les hautes fréquences sont responsables du bruit présent dans une image ce qui peut s'avérer gênant.

L'algorithme de compression a donc pour but de conserver les informations basse fréquence tout en omettant les hautes fréquences afin de ne pas stocker des informations inutiles. Pour cela, nous avons recours à une transformée de Fourier (DFT) discrète puis à une quantification des données..

## 2 Transformée de Cosinus Discrète (DCT) d'une image

### 2.1 Formule générale pour matrices carrées

Avant d'appliquer la DCT, restriction de la DFT aux fonctions paires, on étalonne les données de l'image afin qu'elles soient centrées autour de 0 (cela re-

vient à se débarrasser du coefficient constant dans la transformée). Si on suppose qu'une image  $N \times N$  est représentée par une matrice

$$M = (M_{i,j})_{i,j \leq N-1} \quad (1)$$

de dimension  $N \times N$  (en supposant qu'on ne traite qu'un seul canal de couleur), la DCT de cette image s'écrit alors en dimension 2, pour

$$0 \leq k, l \leq N - 1$$

:

$$D_{k,l} = \frac{1}{4} C_k C_l \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \cos \left[ \frac{(2i+1)k\pi}{2N} \right] \cos \left[ \frac{(2j+1)l\pi}{2N} \right] \quad (2)$$

$$C_u = \begin{cases} \frac{1}{\sqrt{2}}, & \text{si } u = 0, \\ 1, & \text{sinon.} \end{cases}$$

## 2.2 Division de l'image en sous-images

Afin de simplifier les calculs, nous allons travailler sur des blocs  $8 \times 8$  de l'image. La DCT de cette petite image s'écrit alors pour

$$0 \leq k, l \leq 7$$

:

$$D_{k,l} = \frac{1}{4} C_k C_l \sum_{i=0}^7 \sum_{j=0}^7 \cos \left[ \frac{(2i+1)k\pi}{16} \right] \cos \left[ \frac{(2j+1)l\pi}{16} \right] \quad (3)$$

## 2.3 Formule de changement de base

Appliquer la DCT à notre image  $8 \times 8$  revient à effectuer un changement de base de la matrice de départ

$$M = (M_{i,j})_{i,j \leq 7} \quad (4)$$

par une matrice de passage  $P = (P_{i,j})_{i,j \leq 7}$  tel que :

$$D = PMP^{-1} \quad (5)$$

Or le choix de la formule (1) (généralement appelée DCT-II) n'est pas anodin, il permet d'obtenir un opérateur orthogonal c'est à dire que :

$$P^{-1} = P^T$$

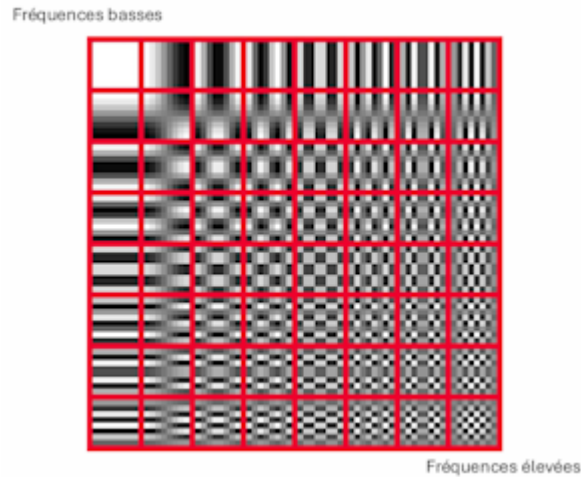
On obtient finalement :

$$D = PMP^T \quad (6)$$

Cela réduit considérablement le nombre de calculs à effectuer car il s'agit dorénavant seulement de permutations.

## 2.4 Forme de la matrice D

Les entrées de la matrice D (cf equation 3) correspondent à fréquence des changements d'intensité lumineuse. Les coefficients dans le coin supérieur gauche de la matrice D correspondent aux petites fréquences en x et en y (variations lentes), alors que les coefficients en bas à droite correspondent aux variations d'intensité aux hautes fréquences (variations rapides)



## 3 Compression

### 3.1 Compression avec perte et JPG

Après avoir ordonné nos données dans le domaine fréquentiel, la propriété de la matrice obtenue reste à être utilisée. Du fait de la séparation des basses et hautes fréquences dans la matrice, il est maintenant facile d'éliminer les hautes-fréquences non essentielles à l'image. C'est l'étape de quantification. Cette élimination d'un grand nombre de données est la clé de l'algorithme de compression JPG. Moins d'informations à stocker signifie un gain en de place considérable. On notera qu'en fonction des différentes caractéristiques des images compressées et de la qualité souhaitée à la décompression, il est possible d'obtenir des taux de compression de l'ordre de 70 à 99%.

### 3.2 Implémentation de la compression

L'étape de quantification nécessite l'usage d'une matrice Q appelée matrice de quantification de taille de l'image (ici 8x8). La manipulation consiste ainsi à effectuer la division terme à terme de la matrice D obtenue à l'étape de la DCT (cf equation 2) par cette matrice Q. La matrice de quantification Q dans la norme de compression JPEG en taille 8x8 est la suivante :

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 13 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad (7)$$

Les termes en bas à droite de la matrice Q étant élevés, la division par Q va quasiment toujours annuler les termes en bas à droite, correspondant aux hautes fréquences (sauf s'ils étaient très élevés dans la matrice D). A l'inverse, l'essentiel de l'information qui se trouve généralement aux basses fréquences sera conservé, les coefficients de Q en haut à gauche étant nettement plus petits. La matrice ainsi obtenue sera très creuse, avec très peu de valeurs non nulles. Enfin pour une image complète, on réitère l'opération pour chaque bloc 8x8 qui la composent puis on enregistre ces informations.

### 3.3 Exemple en image



Image originale



Image compressée

La raison pour laquelle nous obtenons une image grisée avec seulement quelques points de couleurs uniformément répartis est parce que nous avons supprimé les informations hautes-fréquences présents dans la majorité de chaque matrice 8x8 composant l'image. Cette information tronquée à 0 est lue comme une valeur neutre lors de l'impression de l'image, le gris. Les informations conservées donc visibles ici (basses-fréquences) ne sont présentes qu'en haut à gauche de chaque matrice composant l'image, pavant ainsi celle-ci à intervalle régulier en x et en y de leurs valeurs. Bien évidemment, on ne s'arrête pas là car ce n'est pas le résultat escompté.

## 4 Décompression

Une fois l'image compressée et stockée, il s'agit d'être en capacité de la lire. Pour cela, rien de plus facile, on réitère l'algorithme mais cette fois-ci à l'envers. On commence par diviser notre image compressée en blocs 8x8. Puis on multiplie terme à terme ces blocs matriciels par la matrice quantification  $Q$ . On obtient alors l'image décompressée exprimée dans la base des cosinus  $D_1$ . Enfin on effectue la DCT inverse c'est à dire qu'on récupère la matrice  $M_1$  telle que :

$$M_1 = P^T D_1 P \quad (8)$$

(cf equation 6) Cette matrice est ressemblante à la matrice  $M$  de départ (cf equation 4) mais elles ne sont pour autant pas égales. En effet la quantification à l'étape vu section 3.2 introduit une erreur suite à la suppression quasi totale des hautes-fréquences de l'image. Enfin on rassemble toutes les petites images 8x8 pour reformer l'image initiale (ou presque).

### 4.1 Exemple en image



## 5 Principe de filtrage

Certaines images contiennent du bruit (perturbations hautes fréquences qui altèrent la qualité visuelle de l'image). On aimerait bien éliminer ce bruit afin d'obtenir une image plus agréable. Parfois la diminution bien que radicale des valeurs haute-fréquence de l'image grâce à l'étape de quantification n'est pas suffisante. Il convient donc de supprimer ces valeurs à partir d'un certain seuil de vibration. C'est une étape de troncature. Cette manipulation peut être effectuée soit en substitution de la quantification ou bien en complément. On met à 0 tous les coefficients  $D_l, k$  de la matrice  $D$  dont les indices vérifient  $l + k \leq F$  où  $F$  est la fréquence de coupure. Evidemment plus  $F$  tend vers 0 plus l'erreur engendrée par cette opération augmente. Cependant elle dépend de différents facteurs comme les composantes fréquentielles de l'image.

## 5.1 Images basses-fréquences

On remarque que si l'image est plus typée par les basses fréquences, l'erreur restera petite (peu de hautes-fréquences tuées car peu de hautes-fréquences tout court). Dans ce cas, le filtrage n'est pas utile car pas de bruit dans l'image mais il peut être substitanciel à la quantification en ce qui concerne la compression (plus rapide et tout aussi précis car peu de hautes-fréquences initialement, les supprimer entièrement ne pose donc pas de problème particulier). Si celle-ci est bruitée, l'erreur sera un peu augmentée mais le résultat visuel sera très bon. La fonction de filtrage sera entièrement comblée.

## 5.2 Images hautes-fréquences

Dans le cas des images présentant de petites valeurs basses-fréquences que celles-ci soient bruitées ou non, l'erreur avec l'image initiale sera plus grande. Le choix du seuil sera donc primordial pour permettre de ne cibler que le bruit (si possible). Le résultat obtenu ne sera pas aussi satisfaisant que pour les images basses-fréquences. Il s'agira d'obtenir un compromis entre une image bruitée détaillée et une image non bruitée mais avec un effet de flou.

## 5.3 Images sans tendance fréquentielle particulière

En ce qui concerne les images ne présentant ni particulièrement peu de basses ou de hautes fréquences (images du quotidien), il s'agit de choisir judicieusement le seuil afin d'éviter les déconvenues liées aux détails fins.

## 5.4 Test avec du bruit blanc Gaussien

Le bruit blanc gaussien est un type de bruit statistiquement indépendant et identiquement distribué qui peut être ajouté à une image. Ce bruit agit sur l'intensité lumineuse et suit une distribution normale (ou gaussienne) avec une espérance de zéro et une variance donnée.



PNG BRUIT



Compression+Filtre Haut (Seuil=3)  
Taux de compression : 95%  
Erreur : 17%

## 6 Domaine d'application d'un algorithme de compression avec pertes

Un algorithme tel que celui-ci peut être implémenter dans l'ensemble des activités nécessitant un stockage d'un grand nombre d'images. Cependant, ses paramètres se doivent d'évoluer en fonction de restrictions s'appliquant au type d'image en question et à l'usage qui en est fait. Plus l'image stockée doit être qualitative, moins un algorithme avec perte fera sens car pour avoir une image de bonne qualité on se doit d'avoir un taux de perte minimal, or cela implique un faible taux de compression dans la majorité des cas. Donc un algorithme de ce type ne serait pas de circonstance, il s'agirait de se tourner vers les algorithmes de compression sans perte. De plus la tendance fréquentiel de l'image influe sur les performance de la compression (taux d'erreur et de compression).

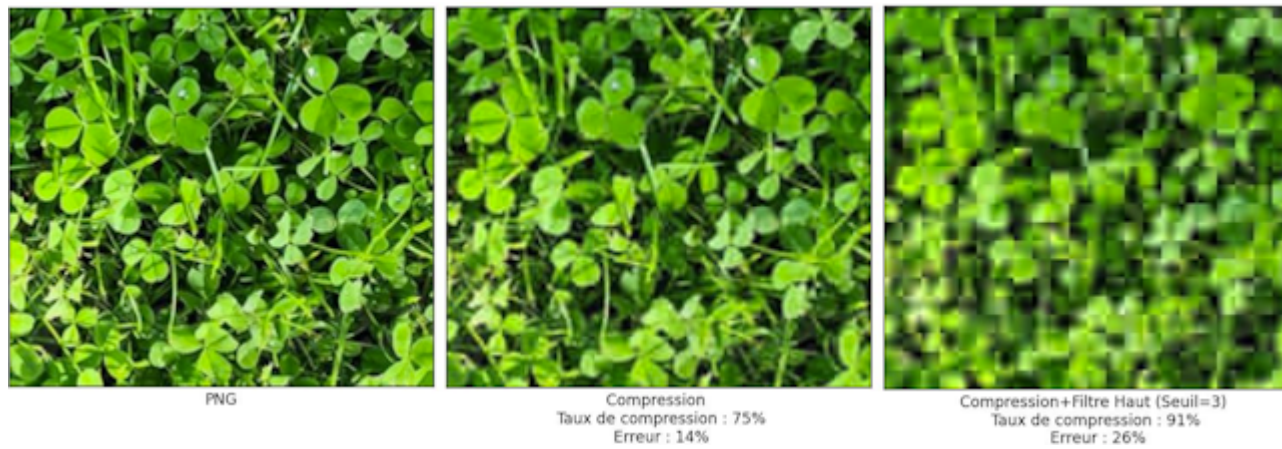
### 6.1 Images hautes-fréquences

La compression des images composées de peu de basses-fréquences peut s'avérer peu efficace. On évalue l'efficacité de la compression en considérant à la fois le taux de compression (que l'on souhaite maximal) et le taux d'erreur (minimal). Prenons en exemple cette image :



Comparons la même image suite à sa compression et à son, filtrage:

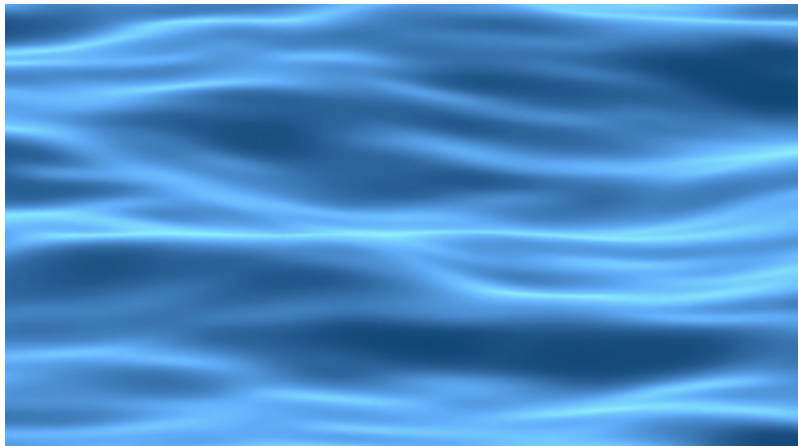




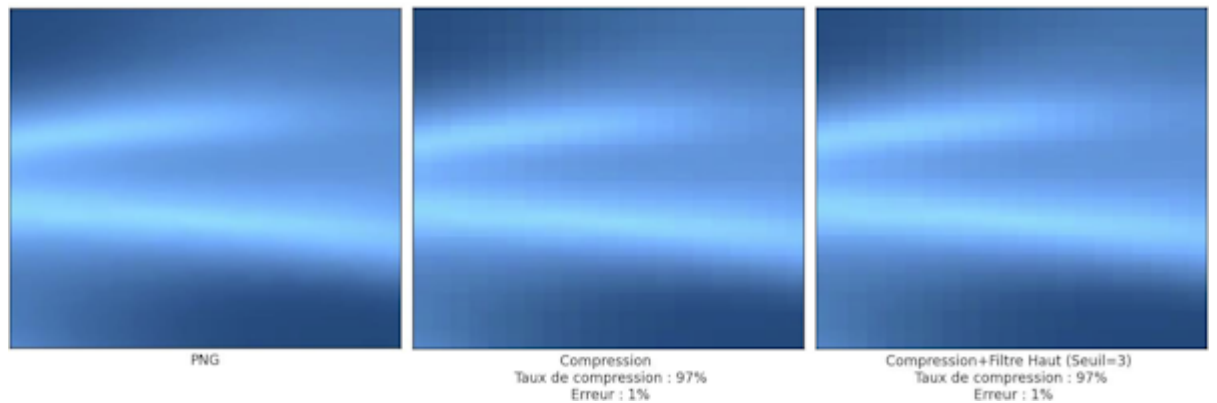
Le résultat obtenu n'est que peu qualitatif, les hautes-fréquences étant essentielles dans cette image, la perte de ces informations influe grandement le rendu final.

## 6.2 Images basses-fréquences

Ici, la composition faible en hautes-fréquences de l'image est avantageuse. La suppression de ces informations n'est que peu visible. Prenons cette image pour illustrer cette idée :







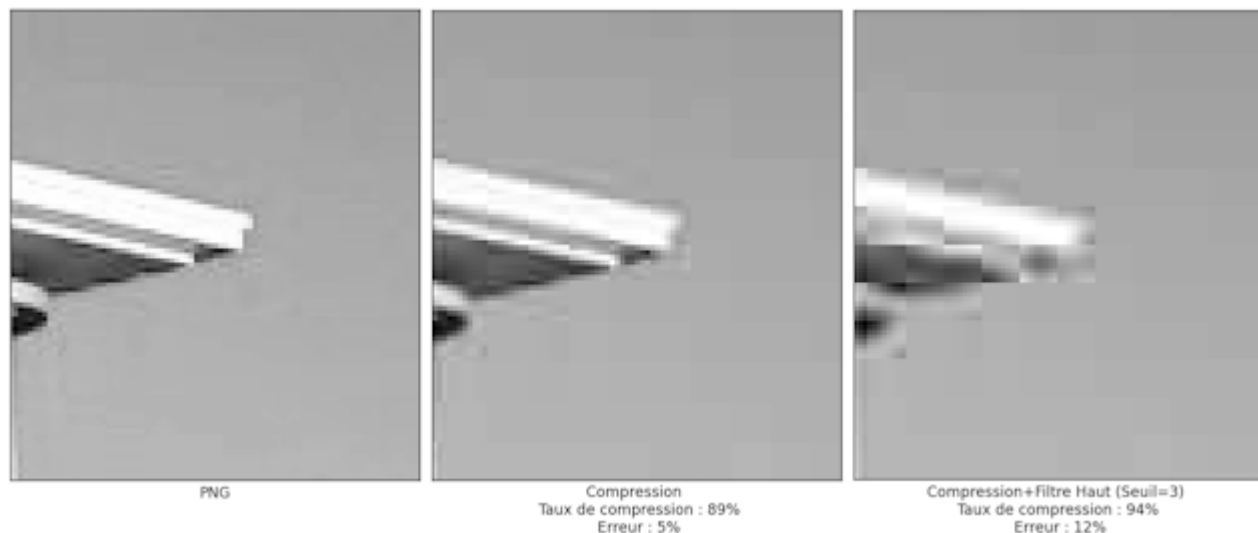
Le résultat est sans appel, malgré une compression de 97% le taux d'erreur lui ne dépasse 1%. L'efficacité de l'algorithme est alors quasiment maximale.

### 6.3 Image noire et blanc

Une image noir et blanc ne contient qu'un seul canal de "couleur", ce sont les nuances de gris. On code 0 pour le noir et 255 pour le blanc. Pour effectuer notre comparaison, nous utilisons cette image :



Voici les résultats obtenus :

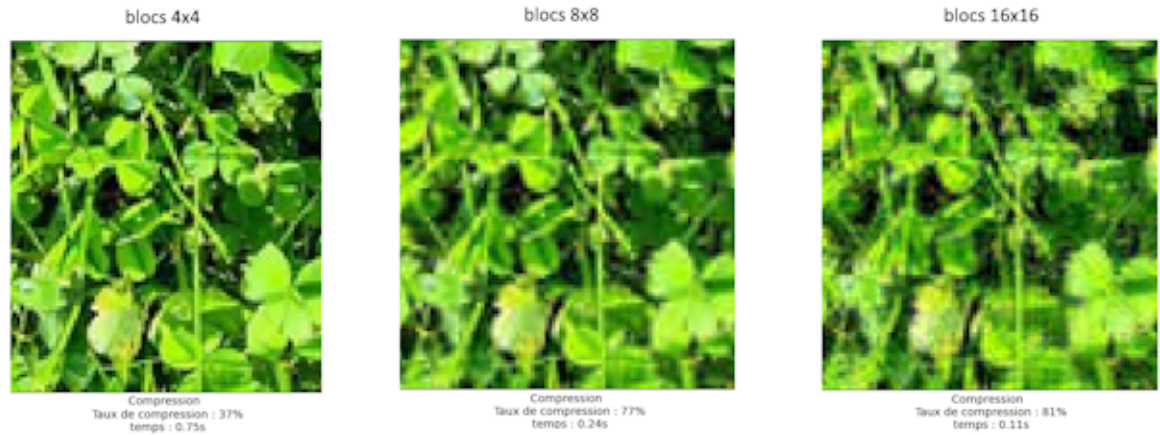


Les taux obtenus sont exactement les mêmes que ceux de l'image originale en couleur. Cela est dû au calcul de l'intensité lumineuse en nuances de gris pour fabriquer l'image noire et blanc. Chaque pixel aura pour valeur la moyenne des 3 valeurs de couleurs. Le calcul de l'erreur se basant sur la norme des matrices composant l'image initiale et finale, nous obtenons donc logiquement le même résultat.

#### 6.4 Modification de la taille des blocs composant l'image

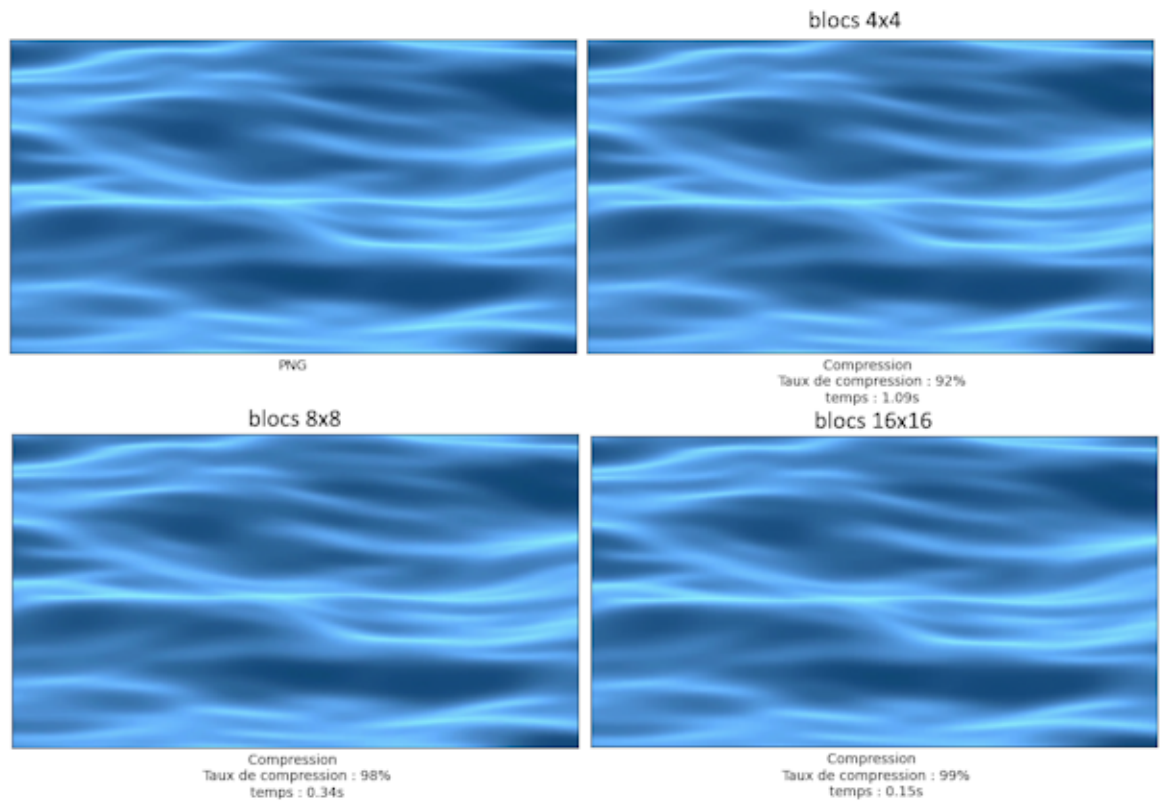
Pour l'instant nous avons seulement considéré la division de l'image en blocs d'image de tailles 8x8. Il est alors intéressant de comparer en temps et en qualité notre algorithme pour différentes tailles de blocs (ici 4x4, 8x8 et 16x16). Pour cela, les matrices de quantifications utilisées sont celles de la norme JPG qui en général est implémenté en 8x8 ou 16x16. Encore une fois nous considérons les images selon leurs tendances fréquentielle. Le temps indiqué dans les sous-sections suivante correspond au temps de compression + décompression de l'image.

#### 6.4.1 Images hautes-fréquences



Pour ce type d'image, la division en matrices 4x4 rend une qualité visuelle bien supérieure aux autres. Cependant le taux de compression en est bien inférieur et le temps bien plus élevé. Malheureusement aucune des solutions présentée ne semble être viable et confirme le résultat de la section 6.

### 6.4.2 Images basses-fréquences



Ici, la taille des blocs ne semble pas avoir de conséquence sur la qualité visuelle de l'image. On compare donc seulement les images selon le temps et le taux de compression. On obtient un algorithme bien plus efficace avec la division en matrices 16x16 pour ce type d'image. Cela est intéressant.

### 6.4.3 Images quelconques



Pour cette image ayant un profil à la fois hautes et basses-fréquences, les 2 résultats précédents sont confirmés. Dans les régions de l'image peu texturées (ciel par exemple), on observe aucune différence visuelle notable. Cependant pour les régions très texturées, on est capable de distinguer de mieux en mieux les blocs de l'image plus la taille de ceux-ci augmente. De plus on semble introduire un flou à l'intérieur de ces blocs de plus en plus importants par la même occasion. Plus la taille d'un bloc est grand, plus il est probable qu'il contienne des morceaux de l'image bien différents et donc on note une augmentation des hautes-fréquences à l'intérieur de ces blocs. Cela induit une erreur liée à la quantification des données plus importante. Cependant, on conserve les résultats de temps et de taux de compression liés à la taille du maillage de notre image.

### 6.5 Maillage idéal

Il n'y a pas de maillage idéal de l'image, celui-ci dépend des tendances fréquentielles de cette dernière ainsi que du résultat escompté de notre algorithme. Plus on veut une image fidèle à la réalité plus un maillage fin sera requis dans la majorité des cas (hors image basses-fréquences) et inversement. Mais cela influera de manière non négligeable sur le temps de compression et décompression ainsi que sur le taux de compression. Si ces critères sont plus importants que la qualité de l'image par contre, il est alors intéressant d'augmenter la taille du maillage.

## 7 Conclusion

Les algorithmes de compression d'image tels que le JPG ont permis des avancées majeures dans le domaine du traitement d'images, en permettant de réduire efficacement la taille des fichiers tout en préservant une qualité visuelle acceptable. Le JPG utilise une combinaison de techniques comme la transformation en cosinus discrète (DCT) et la quantification pour atteindre cette compression. Ils peuvent être adaptés à différents types d'images ou à différents usages (définition d'une certaine qualité en sortie ou bien d'un taux ou temps de compression voulu) grâce à la modification de la matrice de quantification, ou encore à celle du maillage de l'image. Cependant, bien que performant, il présente certaines limites, notamment en ce qui concerne les zones à forte variation d'intensité de couleur. Il serait ainsi intéressant de comparer leur performance sur ce domaine avec d'autres algorithmes telles que ceux de compression sans perte par exemple.