

Mettre en mémoire et analyser une image

Comment mettre en mémoire une image ? Très simple ! Il suffit de mettre en mémoire la couleur de chaque pixel. Très simple ? Mais cela demande une quantité énorme de mémoire ! Alors, il vaut mieux faire usage d'astuces et utiliser une méthode qui permet à la fois d'économiser de la mémoire et de comprendre la structure de l'image. Mettre en mémoire une image de manière économique, c'est ce qu'on appelle la compression d'images. Ici, nous allons explorer le format JPEG 2000.

Christiane Rousseau
Université de Montréal



Regardons la photo ci-contre. Elle a 816 pixels de large, et 608 de haut. La couleur de chaque pixel est un mélange des trois couleurs de base : rouge, vert, bleu, et, pour chacune, on a $256 = 2^8$ intensités possibles. Donc, pour chaque pixel il faut $3 \times 8 = 24$ bits d'information, ou encore 3 octets, car un octet vaut 8 bits. Au total, pour donner la couleur des $816 \times 608 = 496\,128$ pixels il faut $496\,128 \times 3 = 1\,488\,384$ octets, soit autour de 1,49 Mo¹. Pourtant, le fichier JPEG de cette photo n'a que 237 Ko, soit autour de 237 000 octets, de l'ordre de 6,28 fois moins !

Effectivement, sur cette photo on a de grandes régions de ciel bleu uniforme, et également de grandes régions blanches très uniformes. On peut imaginer que dans une région très uniforme on peut faire mieux que de lister la couleur de chacun des pixels. Aussi, on n'a pas besoin du même degré de détail si on regarde une petite photo que si on l'agrandit pour en faire une affiche.

1. Quoique le mégaoctet (Mo) soit officiellement défini comme 10^6 octets depuis 1998, cette norme n'est pas universellement appliquée et on trouve encore, en particulier chez les constructeurs de matériel informatique (microprocesseurs, etc.), que le mégaoctet représente $2^{20} = 1\,048\,576$ octets. La différence est de l'ordre de 4,49 %.

Revenons au cas d'une grande région de couleur presque uniforme. Il est plus simple de se donner la couleur globale et de seulement noter les pixels qui ne sont pas de cette couleur. C'est un peu cette idée qu'utilise le standard JPEG 2000, un standard plus performant que le standard JPEG de nos appareils de photo, mais surtout utilisé par les professionnels de l'image. Nous allons le décrire pour une image en tons de gris, un ton de gris étant un nombre de 0 à $255 = 2^8 - 1$, où 0 correspond au noir et 255 au blanc (voir encadré pour la conversion en ton de gris).

La première étape est de transformer l'information. Ainsi, se donner deux nombres, a et b , c'est la même chose que se donner leur moyenne, $x = (a + b)/2$ et la moitié de leur différence, $y = (b - a)/2$. En effet, $b = x + y$ et $a = x - y$.

Appliquons ceci aux 16 pixels des positions 212 à 227 de la ligne 220 en les regroupant 2 par 2.



Leurs tons de gris sont donnés par :

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 25 | 23 | 23 | 22 | 31 | 115 | 124 | 125 | 130 | 127 | 138 | 222 | 222 | 228 | 229 | 229 |
|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

Cela nous donne huit moyennes,

| | | | | | | | |
|----|------|----|-------|-------|-----|-----|-----|
| 24 | 22,5 | 73 | 124,5 | 128,5 | 180 | 225 | 229 |
|----|------|----|-------|-------|-----|-----|-----|

et huit demi-différences.

| | | | | | | | |
|----|------|----|-----|------|----|---|---|
| -1 | -0,5 | 42 | 0,5 | -1,5 | 42 | 3 | 0 |
|----|------|----|-----|------|----|---|---|

Que voit-on ? Que plusieurs des différences sont très petites. Si on garde les nombres tels qu'ils sont, on garde toute l'information sur

la photo. Si on veut compresser l'information, alors on va se permettre d'arrondir à 0 les petits nombres.

On voit aussi que la deuxième ligne met en évidence les endroits où on a un saut de ton de gris.

On va appliquer la même idée en deux dimensions.

L'idée est que toute l'information contenue dans un carré 2×2

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

est contenue dans un nouveau carré 2×2 construit ainsi :

| Transformation | |
|--|--|
| Avant | Après |
| $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ | $\begin{bmatrix} a+b+c+d & (b-a)+(d-c) \\ (c-a)+(d-b) & (b+c)-(a+d) \end{bmatrix}$ |
| | $\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$ |

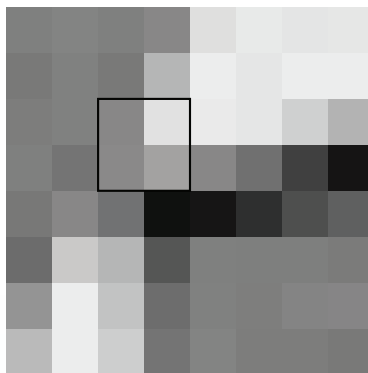
On va appliquer cela à un rectangle $2n \times 2m$ que l'on considère comme un rectangle formé de m rangées de n blocs 2×2 et on va disposer les éléments rouges, bleus, verts et mauves dans des rectangles $n \times m$ disposés comme suit

$$\begin{bmatrix} n \times m & n \times m \\ n \times m & n \times m \end{bmatrix}$$

Regardons la construction. Le coefficient bleu est obtenu en faisant la moyenne des demi-différences horizontales. Il mesure donc des différences verticales. De même, le coefficient vert est la moyenne des demi-différences verticales et mesure des différences horizontales.

Quant au coefficient violet il est la demi-différence de la moyenne sur l'antidiagonale et de la moyenne sur la diagonale. Il mesure les différences obliques.

Regardons le carré 8×8 ci-dessous : il représente les pixels 226 à 233 vers la droite et 216 à 223 vers le bas à partir du coin en haut à gauche de la photo. Sur ce carré on a mis en évidence un petit carré 2×2 sur lequel on va suivre les transformations.



Les tons de gris sont donnés dans le tableau (en caractères gras notre carré 2×2)

| | | | | | | | |
|-----|-----|------------|------------|-----|-----|-----|-----|
| 126 | 130 | 126 | 134 | 221 | 231 | 228 | 230 |
| 120 | 127 | 120 | 181 | 234 | 229 | 235 | 234 |
| 123 | 127 | 134 | 224 | 231 | 228 | 207 | 179 |
| 126 | 116 | 137 | 162 | 135 | 112 | 64 | 22 |
| 119 | 135 | 114 | 17 | 24 | 48 | 78 | 96 |
| 108 | 202 | 182 | 86 | 127 | 125 | 125 | 122 |
| 146 | 234 | 194 | 108 | 128 | 124 | 131 | 134 |
| 185 | 234 | 205 | 115 | 130 | 123 | 124 | 120 |

Maintenant, pour le carré rouge, on va faire les moyennes de chaque carré 2×2 et arrondir.

| | | | |
|-----|------------|-----|-----|
| 126 | 140 | 229 | 232 |
| 123 | 164 | 176 | 118 |
| 141 | 100 | 81 | 106 |
| 200 | 155 | 126 | 127 |

Ci-contre on a, en tons de gris, le carré rouge obtenu de notre carré.

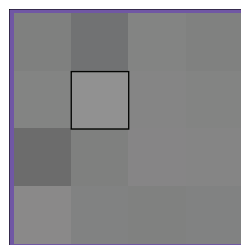
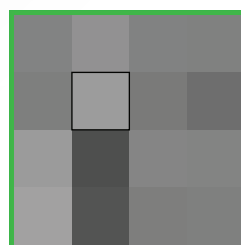
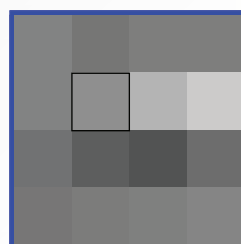
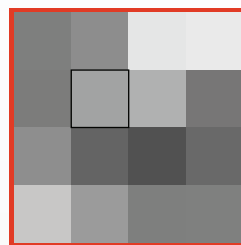
Faisons la même chose pour les carrés bleu, vert et violet, dont les valeurs sont données dans les tableaux suivants et les carrés correspondants ci-contre.

On ne peut comme tel visualiser les tableaux bleu, vert et violet car les entrées sont dans $(-127,5; 127,5)$. Pour visualiser le contenu on va ajouter 127,5 à toutes les valeurs et les représenter (après arrondi) comme des tons de gris.

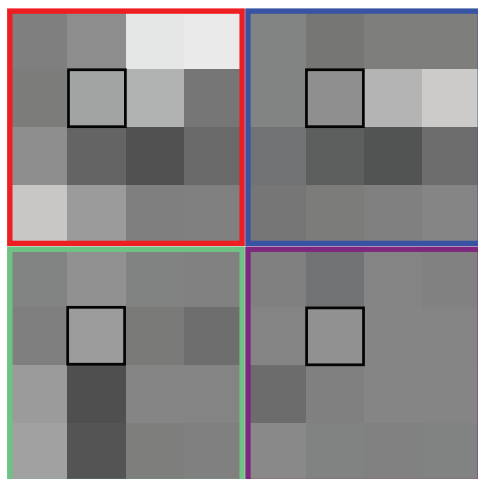
| | | | |
|-------|--------------|-------|--------|
| 2,25 | -10,25 | -2,75 | -2,75 |
| 2 | 14,75 | 53 | 75 |
| -14 | -34,25 | -45 | -18,25 |
| -9,75 | -4,5 | -0,25 | 5,25 |

| | | | |
|-------|--------------|-------|-------|
| 2,75 | 17,25 | 1,25 | 0,25 |
| -1,5 | 28,75 | -6,5 | -17,5 |
| 27,5 | -48,25 | 5,5 | 3,75 |
| 34,25 | -44 | -2,75 | -0,25 |

| | | | |
|-------|--------------|------|------|
| -0,75 | -13,25 | 3,75 | 0,75 |
| 3,5 | 16,25 | 5 | 3,5 |
| -19,5 | -0,25 | 6,5 | 5,25 |
| 9,75 | 1 | 0,75 | 1,75 |



Les quatre tableaux contiennent toute l'information sur l'image, et de manière organisée.



Le carré rouge donne la *structure* de l'image, et les carrés bleu, vert et violet, les *nuances* horizontales, verticales et obliques.

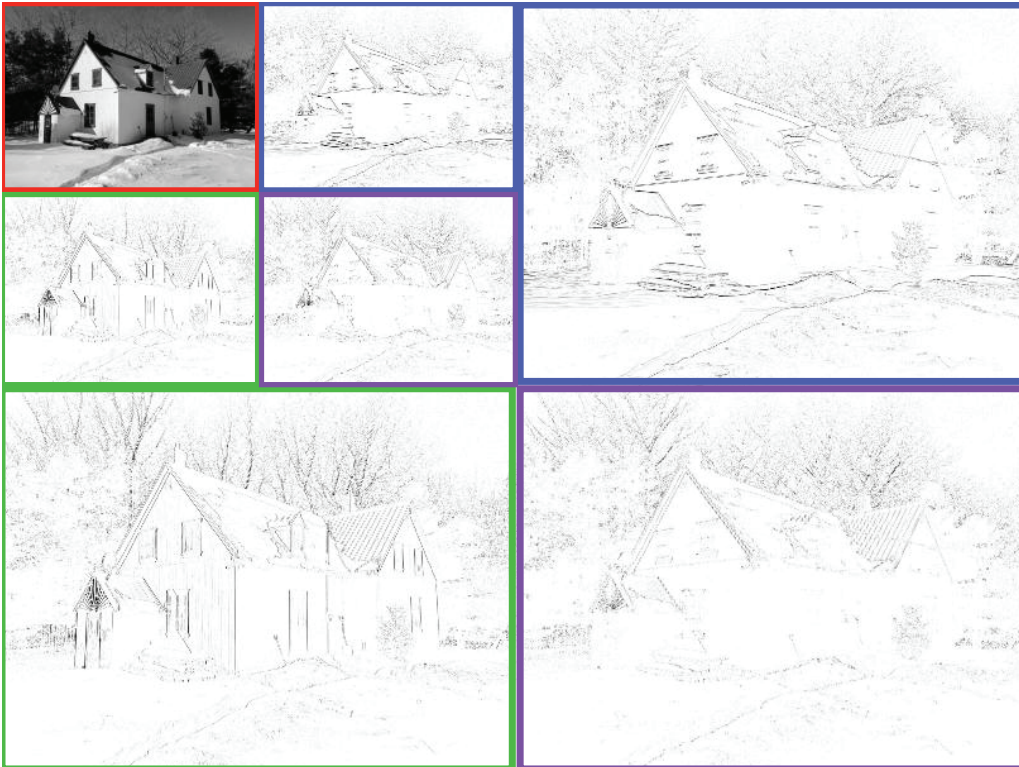
On va maintenant appliquer cela sur le fichier de la photo en tons de gris. Mais les nuances horizontales, verticales et obliques sont difficiles à voir comme différences avec le ton de gris moyen. On va tricher un peu et choisir de représenter en très pâle les petites différences et en foncé les grandes différences, indépendamment de leur signe. Voici ce que cela donne l'illustration en bas de page².

On voit bien que les rectangles bleu, vert et violet font ressortir les contours horizontaux, verticaux et obliques.

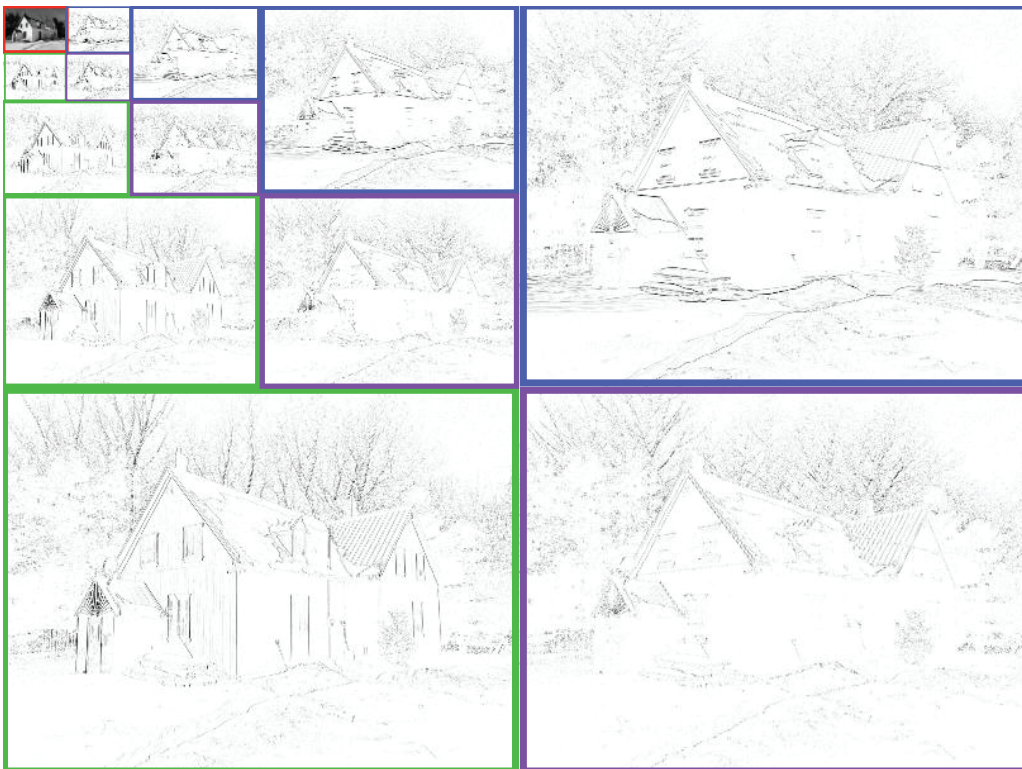


2. Les images ont été programmées par Olivier Rousseau.

On peut itérer le processus une deuxième fois sur l'image en haut à gauche. Ceci permet de mettre en lumière de nouvelles structures à l'échelle 4.



Une troisième itération permet de mettre en lumière les structures à l'échelle 8, une quatrième itération met en évidence les structures à l'échelle 16, etc. Voici la quatrième itération.



À chaque étape, on découvre de nouvelles structures à une nouvelle échelle. Ceci nous donne un outil très polyvalent pour analyser l'image.

Comment reconstruire l'image ?

Remarquons que le processus décrit est réversible. On peut donc récupérer l'image en reconstruisant les rectangles rouges (les images moyennes) en partant du plus petit et en s'aidant des rectangles bleu, vert et violet de la même taille pour reconstruire le rectangle rouge précédent et, en itérant jusqu'à reconstruire l'image finale.

A-t-on vraiment avancé ?

Oui ! À condition d'encoder astucieusement l'information. Rappelons que chaque valeur était représentée par 8 bits. On peut décider que certaines valeurs qui apparaissent souvent soient représentées par moins de 8 bits et que les valeurs rares soient représentées par plus de 8 bits ; aussi, que certains groupes de valeurs (par exemple des petits carrés 2x2 qui apparaissent souvent) soient représentés par un unique petit symbole. On pourrait décider d'encoder 30 zéros de suite par l'information 30 et 0. Etc. Encoder astucieusement sans perte d'information est un très grand chapitre de la théorie de l'information, dont l'exemple le plus simple est le codage de Huffman pour l'encodage d'un texte. Un tel encodage pour le format JEG2000 est très sophistiqué et nous ne rentrerons pas dans les détails. Il permet une très bonne compression sans aucune perte d'information !

On peut pousser plus loin la compression en choisissant d'arrondir les entrées et, en particulier, de mettre à zéro toutes les petites entrées. Ensuite, à partir des données compressées on reconstruit l'image initiale. Que signifie « petite entrée » ? Cela dépend du degré de compression voulu. Plus on arrondit les entrées et plus on en annule, plus on compresse !

On peut pousser plus loin la compression en choisissant d'arrondir les entrées et, en particulier, de mettre à zéro toutes les petites entrées. Ensuite, à partir des données compressées on reconstruit l'image initiale. Que signifie « petite entrée » ? Cela dépend du degré de compression voulu. Plus on arrondit les entrées et plus on en annule, plus on compresse !

De multiples applications du format JPEG 2000

La première application du format JPEG 2000 est bien sûr celle qu'on a annoncée, soit de compresser l'image.

Lors de l'encodage, on a regardé l'image à différentes échelles. On peut choisir de la reconstruire en privilégiant une échelle particulière et ainsi mettre en évidence des motifs subtils à cette échelle, mais pouvant passer inaperçus quand tous les détails sont présents.

La conversion en tons de gris et le standard YCbCr

Pour une image en couleurs, on se donne les niveaux d'intensité de rouge, vert et bleu de chaque pixel. Selon les standards, on fonctionne avec 256 niveaux pour chaque couleur, allant de 0 à 255, ou encore avec un nombre entre 0 et 1 correspondant au niveau de couleur divisé par 255. Nous prenons ce deuxième point de vue. La couleur d'un pixel est donc un vecteur

$$V = (r, v, b) \in [0, 1]^3.$$

C'est le format RGB.

La luminance, qui correspond au niveau de gris, est donnée par

$$Y = 0,299r + 0,587v + 0,114b.$$

En fait, même quand on compresse une image en couleurs, il existe un meilleur choix de coordonnées pour donner la couleur de chaque pixel que r, v, b . La première coordonnée est la luminance Y définie ci-dessus.

Comme la lumière blanche est la somme des trois couleurs, pour obtenir toutes les couleurs, il faut retrancher du bleu ou du rouge au ton de gris. Les deux quantités obtenues sont deux informations de *chrominance*, soit la *différence bleu* et la *différence rouge*, données respectivement par

$$C_b = 0,565(b - Y)$$

et

$$C_r = 0,713(r - Y).$$

Le standard YCbCr consiste à se donner le vecteur V par ses coordonnées (Y, C_b, C_r) .

Les coordonnées Y, C_b, C_r sont en général beaucoup moins corrélées que les coordonnées r, v, b , ce qui est plus adapté à la compression.

Les ondelettes et le format JPEG

On a utilisé dans nos exemples des moyennes et des demi-différences. Prendre la moyenne ou la demi-différence de deux pixels voisins, c'est passer un *filtre* sur les données. Les ondelettes sont des *filtres* qu'on passe sur les données d'un signal ou d'une image pour en extraire de l'information. Ici on a utilisé une *ondelette* discrète, l'*ondelette de Haar*. Il existe toute une famille d'ondelettes discrètes, la plus simple étant l'ondelette de Haar qui agit sur deux pixels à la fois

dans chaque direction. On utilise en général dans le format JPEG 2000 des ondelettes plus compliquées que l'ondelette de Haar, et qui agissent sur plus de deux pixels à la fois dans chaque direction. Plusieurs de ces ondelettes ont été construites par la mathématicienne Ingrid Daubechies. C'est l'ondelette 9/7 de Daubechies qui est la plus utilisée dans la compression avec perte d'information du format JPEG 2000.

On peut choisir de garder l'information très précise pour une petite région d'une image et compresser énormément l'information dans les autres régions de l'image. C'est un atout très important lorsqu'on doit gérer des quantités très grandes de données et qu'on a besoin d'énormément de précision dans une petite région. Pour ce but précis, le format JPEG 2000 surpasse de loin les autres méthodes de compression d'images.

JPEG versus JPEG 2000

Même s'il date de près de 25 ans, le format JPEG 2000 est peu répandu, alors que le format JPEG est presque universellement supporté par les outils de lecture d'images. Ainsi, Safari est le seul navigateur supportant JPEG 2000 en 2025. C'est parce que l'encodage et le décodage de JPEG 2000 demandent beaucoup plus de calculs que celui de JPEG ou d'autres formats standards d'images. Cependant, le format JPEG 2000 est privilégié lorsque la qualité des images est importante. En particulier, le rendu des contours contrastés est meilleur. Sur le site d'Adobe, qui supporte ce format, il est écrit : « Grâce à ses taux de compression élevés, JPEG 2000 peut également compresser une image jusqu'à 200% de plus que JPEG tout en conservant la même qualité par rapport à un fichier de même taille. »³ L'Organisation météorologique mondiale a intégré JPEG 2000 dans

son nouveau format de fichier GRIB2, lequel est conçu pour la distribution mondiale des données météorologiques. Le protocole standard international DICOM utilisé en imagerie médicale (Digital Imaging and Communications in Medicine) supporte le format JPEG 2000. Ce format est idéal puisqu'il permet une meilleure compression tout en respectant l'intégrité de l'image.⁴

Ingrid Daubechies

Ingrid Daubechies est une mathématicienne et physicienne d'origine flamande. En 1987, elle s'installe définitivement aux États-Unis. Ses travaux sur les ondelettes lui vaudront d'être conférencière plénière au Congrès international des mathématiciens de 1994 et d'être la première femme à recevoir le très prestigieux prix Wolf de mathématiques en 2023. Ingrid Daubechies a également été la première femme présidente de l'Union mathématique internationale de 2011 à 2014. Elle a travaillé avec des conservateurs de musées à analyser des peintures. En particulier, à l'aide de techniques d'analyse d'images, elle a pu reconstruire une ancienne peinture de van Gogh cachée sous une peinture beaucoup plus récente. En collaboration avec des géophysiciens, elle a aussi utilisé des techniques d'analyse de signaux sismiques pour « voir » les *plumes volcaniques*, ces cheminées au travers du manteau terrestre qui sont à l'origine des chapelets d'îles comme Hawaïi.

3. <https://www.adobe.com/ca/creativecloud/file-types/image/comparison/jpeg-vs-jpeg-2000.html>

4. <https://minnovaa.com/jpeg-2000/>