

Les sites Web que nous visitons sur la Toile sont maintenant inondés d'images. Cela constitue naturellement un problème de taille, car une quantité énorme d'informations doit être transférée du serveur jusqu'à notre ordinateur. Pour accélérer le traitement de ces images, il faut compresser celles-ci. Ce procédé diminue le poids de l'image en ne sacrifiant pas ou presque pas la qualité.

UN **défi** DE **taille** LES IMAGES SUR LA TOILE,

Marc Bergeron
Cégep de Ste-Foy

Il existe plusieurs formats de compression d'images : JPEG, PNG, GIF, RAW, ... Chacun possède ses avantages. Par exemple, avec la compression au format PNG, il n'y a pas de perte de qualité et celui-ci gère la transparence de l'image (très utile pour de beaux effets visuels sur votre site préféré), mais le fichier sera habituellement plus volumineux que si nous avons utilisé la compression au format JPEG. Ce dernier est justement l'un des plus utilisés, car il est bien adapté aux images ayant plusieurs couleurs différentes comme les photographies. Regardons de plus près l'algorithme de compression JPEG dans lequel les mathématiques interviennent à plusieurs étapes.

Première étape :

Changement de repère

Sur l'écran d'un ordinateur, une image est divisée en points appelés pixels. La couleur du pixel affichée sera habituellement donnée par trois nombres entre 0 et 255

(norme *RVB*) : le premier représente l'intensité de rouge; le deuxième, l'intensité de vert et le dernier, l'intensité de bleu. Le rouge, le vert et le bleu sont les longueurs d'onde détectées par les 3 types de cônes situés sur la rétine de nos yeux. Le mélange de ces trois couleurs permet de recréer toutes les couleurs détectables par l'humain. Une image peut donc être décomposée en trois images élémentaires (voir illustrations). Remarquons que dans l'image ii), la carrosserie de l'automobile nous semble presque noire. Cela est dû au fait qu'il n'y a pratiquement pas de teinte de vert sur celle-ci.

Avant de procéder à la compression au format JPEG, il faudra effectuer un changement de repère¹ du système à trois composantes *RVB* (rouge, vert, bleu) vers le système *LMN* (luminance, chrominance¹, chrominance²).

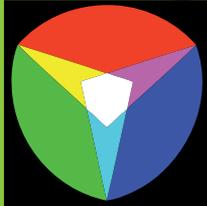
1. Les changements de repère ont déjà été partiellement expliqués dans *Accromath, été-automne 2011*, p. 22



Mélange de couleurs

Si nous mélangeons les lumières rouge, verte et bleue, le résultat perçu par nos yeux sera du blanc, et ce, malgré le fait que le blanc est rigoureusement la somme de toutes les couleurs du spectre lumineux.

Le rouge, le vert et le bleu sont appelés les couleurs primaires additives de la lumière. Si une lumière verte est mélangée avec une lumière rouge,



nous obtenons une lumière jaune comme dans le diagramme ci-contre. Mais n'avons-nous pas appris dans notre enfance que les couleurs primaires étaient le rouge, le bleu et le jaune? Elles sont les couleurs primaires pour les pigments que l'on retrouve, par exemple, dans la peinture. Il faut comprendre que les pigments agissent de façon soustractive, car une partie de la lumière est absorbée par ceux-ci. Une goutte de peinture qui absorbe le bleu sera de couleur jaune, car elle réfléchit le rouge et le vert.

La luminance

La luminance² contient l'information des niveaux de gris de l'image. Pour la définir, il faut remarquer que la sensibilité de nos yeux n'est pas la même pour les couleurs de base (rouge, vert et bleu). La figure suivante représente les trois couleurs en noir et blanc (plus précisément en niveaux de gris) :



Luminance apparente du rouge, du vert et du bleu

Étant donné que le vert est presque au milieu du spectre visible, cette couleur aura la plus grande luminance apparente, suivie par le rouge et le bleu. Après des tests expérimentaux effectués par la Commission internationale de l'éclairage (CIE), la luminance (L) a été définie par la formule suivante :

$$L = 0,299 R + 0,587 V + 0,114 B, \quad (*)$$

2. La luminance est définie par l'intensité lumineuse divisée par unité de surface émettrice. Elle n'est ni plus ni moins la sensation visuelle de la lumière. Si nous sommes éblouis par une lampe de poche ou lorsque nous avons de la difficulté à distinguer les détails dans une pièce sombre, c'est la luminance qui est en cause.

Décomposition d'une image en trois images élémentaires



i) Intensité de rouge (R) des pixels de l'image initiale



ii) Intensité de vert (V) des pixels de l'image initiale



iii) Intensité de bleu (B) des pixels de l'image initiale

où R, V, B sont les composantes du système RVB comprises entre 0 et 255. On remarque que L sera aussi compris entre 0 et 255.

Les chrominances

La chrominance¹ (notée M) varie linéairement avec l'écart entre le niveau de bleu et la luminance du pixel. Donc,

$$M = a(B - L) + b. \quad (**)$$

Pour que M soit compris entre 0 et 255 (après troncature), nous trouvons $a = 0,56434$ et $b = 128$. En utilisant l'égalité (*) et (**), nous obtenons, en gardant 5 décimales,

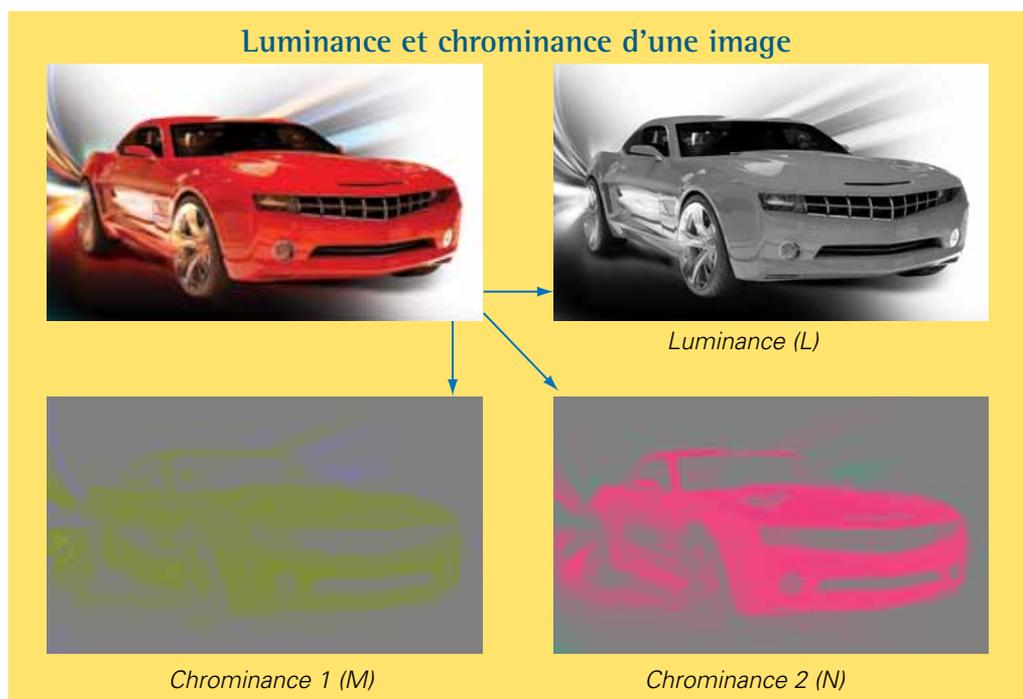
$$M = -0,16874R - 0,33126V + 0,5B + 128$$

La chrominance², pour sa part, contient l'information de l'écart entre le niveau de rouge et la luminance ($a(R - L) + b$, où $a = 0,71327$ et $b = 128$).

Par le même procédé, nous obtenons

$$N = 0,5R - 0,41869V - 0,08131B + 128.$$

Alors, notre image initiale (dans le système RVB) sera maintenant décomposée en trois images dans le système LMN ; (voir illustrations ci-dessous). Nous constatons aisément que la luminance est de loin la plus importante composante du système LMN .



Changement de repère

Le changement de repère du système RVB vers le système LMN se fait facilement en utilisant les matrices et les opérations sur celles-ci :

$$\begin{pmatrix} L \\ M \\ N \end{pmatrix} = \underbrace{\begin{pmatrix} 0,299 & 0,587 & 0,114 \\ -0,16874 & -0,33126 & 0,5 \\ 0,5 & -0,41869 & -0,08131 \end{pmatrix}}_T \begin{pmatrix} R \\ V \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

Pour passer du système LMN au système RVB , lors de la décompression, il suffit de prendre l'inverse de la matrice T :

$$\begin{pmatrix} R \\ V \\ B \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 1,402 \\ 1 & -0,34413 & -0,71414 \\ 1 & 1,772 & 0 \end{pmatrix}}_{T^{-1}} \begin{pmatrix} L \\ M - 128 \\ N - 128 \end{pmatrix}$$

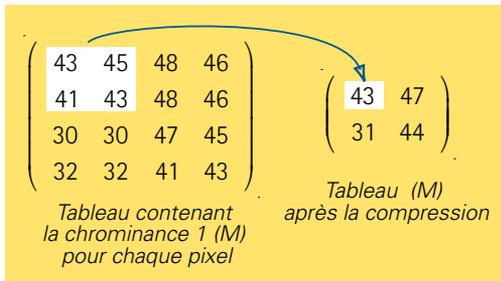
Deuxième étape :

Compression des chrominances

Pour l'instant, nous n'avons pas diminué la taille du fichier image. Étant donné que nos yeux sont moins sensibles aux chrominances (donc aux écarts entre les couleurs), nous allons compresser une première fois en diminuant la quantité d'informations sur les chrominances :

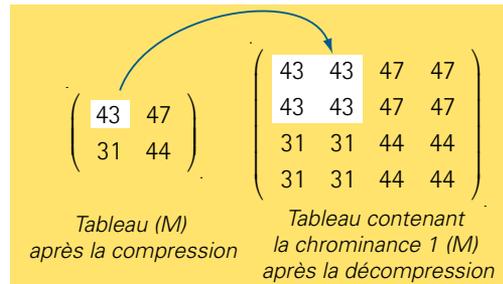
1. Considérons le tableau contenant les valeurs de chrominance1 associées à chaque pixel.
2. Partageons celui-ci en plusieurs petits tableaux de 4 nombres (2 lignes par 2 colonnes).
3. Gardons seulement la moyenne des 4 nombres de chaque petit tableau.

Par exemple, prenons une « image » de 4 pixels par 4 pixels que nous avons décomposée dans le système *LMN* et regardons la chrominance1 (*M*) pour chacun des pixels :



Naturellement, cette compression sera aussi effectuée sur la chrominance2. Donc, l'image de 4 pixels par 4 pixels, avant compression, est décrite par 16 nombres pour la chrominance1 (*M*), 16 pour la chrominance2 (*N*) et 16 pour la luminance (*L*), pour un total de 48 nombres. Après la compression, elle est décrite par 4 nombres pour *M*, 4 pour *N* et nous gardons les 16 nombres de *L*, pour un total de 24 nombres. La taille de notre fichier image sera diminuée de moitié !

Lorsque nous voudrions afficher l'image à partir de ce fichier compressé, lors de la décompression, chaque nombre gardé pour la chrominance1 (ou chrominance2) sera reproduit quatre fois.



Troisième étape :

Compression de la luminance

Étant donné l'importance des valeurs de la luminance, celles-ci ne pourront pas être compressées comme précédemment. La conséquence serait une grande perte de la qualité de l'image à la décompression. Le format JPEG utilise plutôt la transformée en cosinus discrète (*TCD*) où interviennent les fonctions trigonométriques. Ce procédé, énormément utilisé en télécommunication et dans le traitement des sons et des images, permet d'extraire des informations plus importantes que d'autres d'un ensemble de données. L'idée de base de la *TCD* est de réorganiser les données sur la luminance en soulignant les contrastes entre les pixels plutôt que de garder les informations sur chacun des pixels individuellement. Les contrastes les moins importants sont tout simplement éliminés, diminuant par le fait même la taille de notre fichier image.

Voici un exemple très simplifié qui nous permettra de mieux comprendre le mécanisme derrière la *TCD* (pour quelques détails sur le vrai calcul de la *TCD*, voir l'encadré) :

Considérons les valeurs de luminance pour 4 pixels :

$$L = \begin{pmatrix} 120 & 124 \\ 123 & 128 \end{pmatrix}$$

Le premier élément du tableau deviendra la somme de toutes les valeurs (495). Le deuxième élément sera la somme des valeurs de la première colonne moins les valeurs de la deuxième colonne (-9). Ce nombre représente le contraste entre la partie gauche et la partie droite du tableau. Le troisième élément sera la somme des valeurs de la première ligne moins la somme des valeurs de la deuxième ligne (-7). Cela représente le contraste entre la partie du haut et celle du bas. Finalement, le quatrième élément sera la somme de la diagonale principale moins la somme de la diagonale secondaire (1). Le tableau deviendra :

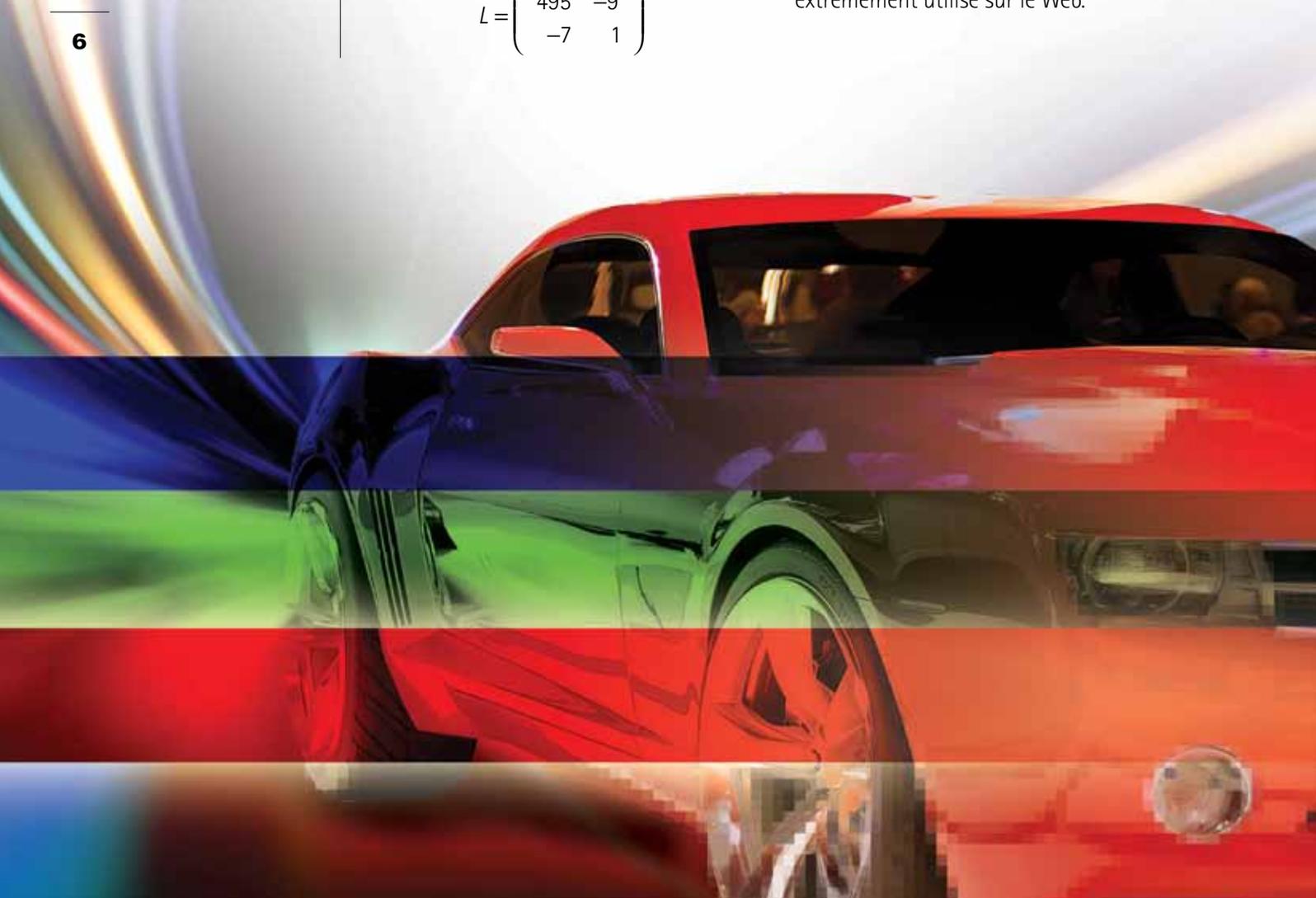
$$L = \begin{pmatrix} 495 & -9 \\ -7 & 1 \end{pmatrix}$$

(Soulignons qu'il est facile, à partir de ce dernier tableau, de retrouver le tableau initial). Mais quel est l'avantage de cette réorganisation des données? Notre œil sera davantage sensible aux grandes valeurs du tableau. Sans entrer dans les détails, la compression consistera à éliminer les plus petites valeurs (en valeur absolue) et réduira alors la taille du fichier image, car il y aura moins de valeurs pour décrire la luminance (dans notre exemple, seules les valeurs 495, -9 et -7 seraient gardées).

Naturellement, le chemin inverse sera encore possible, mais à cause des données éliminées, il y aura une perte de qualité de l'image.

Conclusion

Pour convertir au format JPEG, nous devons donc calculer les composantes *LMN*, compresser les deux composantes de chrominance, puis finalement compresser la luminance à l'aide de la *TCD*. Un fichier au format JPEG pourra être le 1/4 de la taille du fichier original avec une perte minime de qualité. On comprend pourquoi le format JPEG est extrêmement utilisé sur le Web.



Exemple du calcul

Première étape :

En pratique, pour effectuer la compression de la luminance à l'aide de la *TCD*, on débute par découper le tableau contenant les composantes de luminance en petits tableaux de 8 lignes par 8 colonnes. Prenons, par exemple, un de ces tableaux 8x8 :

$$\begin{pmatrix} 120 & 122 & 127 & 110 & 126 & 131 & 142 & 143 \\ 125 & 131 & 119 & 126 & 111 & 118 & 135 & 145 \\ 131 & 135 & 112 & 148 & 149 & 157 & 135 & 159 \\ 157 & 126 & 132 & 147 & 127 & 126 & 110 & 138 \\ 139 & 130 & 139 & 125 & 124 & 118 & 122 & 143 \\ 127 & 147 & 120 & 136 & 137 & 121 & 115 & 142 \\ 116 & 139 & 105 & 141 & 140 & 123 & 117 & 127 \\ 128 & 136 & 118 & 132 & 132 & 127 & 127 & 117 \end{pmatrix}$$

Tableau 1 : contenant la composante de luminance (*L*) pour chaque pixel

$$TCD(x, y) = 0,25 \cdot c(x)c(y) \sum_{i=1}^8 \sum_{j=1}^8 L(i, j) \cdot \cos \left[\frac{\pi}{8} (x-1) \left(i - \frac{1}{2} \right) \right] \cos \left[\frac{\pi}{8} (y-1) \left(j - \frac{1}{2} \right) \right],$$

où $c(k) = \frac{1}{\sqrt{2}}$ si $k = 1$ et $c(k) = 1$ sinon

et où $L(i, j)$ est la luminance de la ligne i et de la colonne j .

Deuxième étape :

Puis, on applique la transformée en cosinus discrète (*TCD*) à chacun des éléments de ces petits tableaux.

On obtient :

$$\begin{pmatrix} 1041 & -7 & 13 & -6 & 26 & -17 & -9 & -19 \\ 10 & -25 & 21 & -4 & -7 & 10 & 15 & 6 \\ -21 & -18 & -3 & 8 & -22 & -1 & -18 & 6 \\ -16 & 1 & 10 & -15 & -9 & -9 & -1 & 15 \\ -6 & 15 & 7 & 0 & -7 & 15 & 19 & 19 \\ 6 & -2 & -15 & 11 & 12 & -5 & 1 & -7 \\ 21 & -13 & -7 & 10 & -1 & -11 & 1 & -2 \\ 8 & -20 & -5 & 6 & -5 & -6 & -1 & 3 \end{pmatrix}$$

Tableau 2 : chaque élément est calculé avec la formule $TCD(x, y)$.

Troisième étape :

C'est à cette étape que la luminance est réellement compressée. On choisit d'abord un facteur de compression (une valeur facteur comprise entre 1 à 25). Plus ce facteur est grand, plus on élimine des valeurs du tableau 2, donc, plus la luminance est compressée. Pour déterminer les valeurs qui seront éliminées, il faut considérer deux critères. Premièrement, plus une valeur obtenue dans le tableau 2 est petite, plus cette donnée pourra être éliminée. Deuxièmement, plus la valeur est près du coin inférieur droit, plus cette donnée pourra aussi être éliminée. Ce dernier critère provient du fait que les valeurs de la *TCD* situées près du coin inférieur gauche correspondent à des irrégularités de l'image initiale moins perceptibles par notre œil. Pour tenir compte de ces deux critères, il suffit de diviser chaque valeur $TCD(x, y)$ par

$$1 + \text{facteur} * (x + y - 1)$$

et d'arrondir à l'entier le plus près.

Constatations :

1. Plus x et y sont grands (donc plus la valeur est près du coin inférieur droit), plus le résultat par la division sera petit, voire égal à 0 après l'arrondi.
 2. Plus la valeur de *facteur* est grande, plus il y aura de zéros dans le tableau, après l'arrondi.
- Continuons avec notre exemple. Si on prend *facteur* = 4 et que l'on divise chaque valeur du tableau 2 par

$$1 + 4 * (x + y - 1),$$

alors on obtient

$$\begin{pmatrix} 208 & -1 & 1 & 0 & 1 & -1 & 0 & -1 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tableau 3 : tableau obtenu avec facteur égal à 4

La compression consiste à éliminer les valeurs nulles du tableau 3 situées à la fin du zigzag comme illustré dans la figure suivante :

$$\begin{pmatrix} 208 & -1 & 1 & 0 & 1 & -1 & 0 & -1 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tableau 4 : chemin de lecture des valeurs en zigzag

Pour notre exemple, on obtient

208, -1, 1, -2, -2, 1, 0, 1, -1, -1, 0, 0, 0, 0, 1, -1, 0, 0, 0, 1, 0, 1, 0, 0, -1, -1, 0, 0, -1, 0, 0, 0, 0, 0, 0, -1.

Les 27 autres valeurs étant toutes nulles, elles ont été éliminées. Cela constitue presque la moitié des 64 valeurs du tableau. Il faut faire ces calculs pour chacun des petits tableaux 8×8 de notre image initiale. La taille de notre fichier image en sera énormément diminuée.

La décompression : afficher l'image à partir du fichier compressé

On peut facilement faire le chemin inverse, la décompression, pour retrouver le tableau 1 du début. Il suffit de multiplier chaque valeur du tableau 3 par

$$1 + 4 * (x + y - 1)$$

et d'appliquer l'inverse de la TCD. Les valeurs ainsi obtenues ne seront pas exactement les mêmes que celles du tableau 1 étant donné les valeurs éliminées à la compression. Cela entraîne naturellement une perte de qualité de l'image initiale.

La possibilité de choisir la valeur *facteur* constitue un grand avantage du format JPEG. On a vu que si *facteur* est grand, alors le fichier image compressé est petit. Par contre, plus *facteur* est grand, plus la qualité de l'image est diminuée après la décompression.



Image initiale



Image décompressée avec facteur = 4 (perte minimale de qualité)



Image décompressée avec facteur = 25 (perte appréciable de qualité, mais la taille du fichier compressé est plus petite)