

Compression d'image

E. LE PENNEC*

La compression de données est une activité ancienne : l'utilisation d'abréviations en est une preuve. Les langues elles-mêmes utilisent des mots de longueurs variées, les plus fréquents étant les plus courts, afin de réduire la taille des phrases. Il est cependant une application plus visible que les autres : la compression des images numériques. C'est elle qui a permis la diffusion des images sur Internet ou encore la démocratisation des appareils photos numériques. Elle constitue également la base de la compression vidéo. L'objectif de cet article est de présenter à travers certains algorithmes classiques de compression des images numériques (GIF, JPEG, PNG) les fondations mathématiques de ceux-ci : théorie de l'information, modélisation statistique, approximation non-linéaire... La dernière partie sera consacrée à l'introduction récente de la géométrie pour améliorer la compression des images naturelles.

Images numériques et compressions

Une image numérique, telle qu'on peut la voir sur un écran d'ordinateur, est une mosaïque de pixels (picture elements) dont la couleur est choisie dans un ensemble fini : il s'agit d'un objet naturellement discret. Il s'identifie à une matrice à h colonnes et v lignes dont les éléments appartiennent à un ensemble fini E . Typiquement, pour une image en niveau de gris, E est constitué des entiers compris entre 0 et 255 et correspond à l'intensité lumineuse de chaque pixel. Ces 256 valeurs distinctes se codent avec 8 bits ($2^8 = 256$), d'où le nom d'image 8 bits. Pour les images couleurs, chaque pixel est caractérisé par 3 intensités lumineuses, celles des canaux rouge, vert et bleu, définissant des images $3 \times 8 = 24$ bits.

Le stockage en mémoire d'une image couleur requiert donc $h \times v \times 24$ bits. Ceci représente rapidement une grande quantité : une image de 4 mégapixels nécessite ainsi $4 \times 2^{20} \times 24$ bits soit 12 mégaoctets, ne permettant le stockage que d'une dizaine de photos sur une carte de 128 mégaoctets. Les modems téléphoniques fournissaient des contraintes encore plus grandes puisque une image couleur de 320×240 non compressée nécessitait pas loin de 4 minutes pour être transmise.

L'objectif de la compression est de réduire la quantité de mémoire nécessaire pour le stockage d'une image ou de manière équivalente de réduire le temps de transmission de celle-ci. Cette compression peut soit conserver l'image intacte, on parle alors de compression sans perte, soit autoriser une dégradation de l'image pour diminuer encore l'empreinte mémoire, on parle ici de compression avec perte. La première méthode est limitée à des facteurs de compressions (rapport entre la taille mémoire originale et la taille comprimée) de l'ordre de 3 tandis que la seconde permet des facteurs beaucoup plus grand au prix de cette dégradation de l'image. Nous allons maintenant voir comment ce procédé, illustré par la figure 1, est possible.

* Laboratoire de Probabilités et Modèles Aléatoires UMR 7599, Université Paris 7. lepenec@math.jussieu.fr

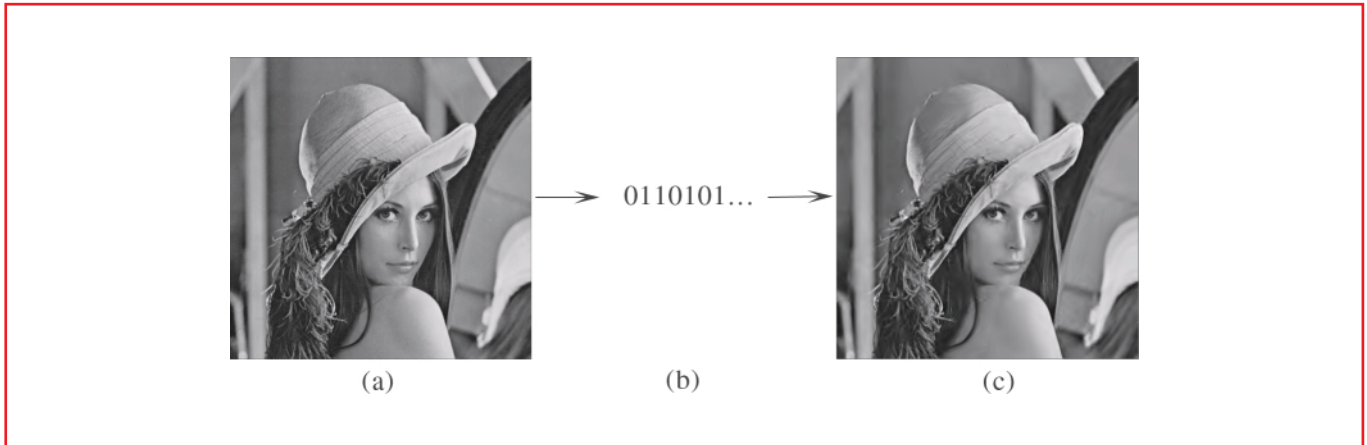


Figure 1 – Compression : une image numérique (a) est transformée en un train de bits (b) qui permet de reconstruire l'image (c) (ici, dans le cas de la compression avec perte, déformée). Le nombre de bits utilisés est plus petit que celui, a priori, nécessaire pour décrire l'image.

Compression sans perte

Le premier algorithme de compression d'image ayant connu un réel succès est un algorithme de compression sans perte et ceci n'est pas un hasard : il est issu de la compression de données arbitraires dans laquelle il est crucial de ne pas perdre d'information. L'algorithme GIF est ainsi l'héritier direct des algorithmes de compression de données génériques introduits à partir de la fin des années 40 par Shannon, le fondateur de la théorie de l'information.

L'entropie de Shannon

L'une des nombreuses questions que s'est posée Shannon à la fin des années 40 (et qu'il a bien souvent résolues) est la suivante : étant donnée une source X produisant des symboles x_i choisis dans un ensemble fini avec une probabilité respective $p(x_i)$, quel est le nombre minimal de bits nécessaire pour transmettre l'information que c'est le symbole x_{i_0} qui a été produit.

Shannon a pu montrer qu'il existe une quantité $H(X)$, qu'il a appelé entropie de la source, définie par

$$H(X) = - \sum_i p(x_i) \log_2 p(x_i)$$

telle que si un codage est possible en \bar{n} bits en moyenne alors nécessairement

$$\bar{n} \geq H(X).$$

L'entropie définit une borne inférieure sur le nombre moyen de bits nécessaire pour coder une source donnée. Ce résultat est complété par la construction explicite à partir des probabilités $p(x_i)$ d'un code de longueur moyenne \bar{n} satisfaisant

$$\bar{n} \leq \lceil H(X) \rceil$$

où $\lceil H(X) \rceil$ est le plus petit entier plus grand ou égal à $H(X)$. Il est donc possible de construire un code quasi optimal dès que l'on connaît la loi de probabilités et la longueur moyenne de ce code est comprise entre $H(X)$ et $\lceil H(X) \rceil$.

L'entropie mesure l'incertitude dans la production de la source X : l'entropie d'une source X produisant N symboles distincts est maximale, égale à $\log_2 N$, si tous les symboles sont équiprobables et minimale, égale à 0, si l'un des symboles est de probabilité 1.

Supposons maintenant que la source X produise des pixels de 24 bits, soit 2^{24} valeurs distinctes possibles et que l'on souhaite coder une image de 4 mégapixels, la source correspondante pour les images est X^n où $n = 4 \times 2^{20}$ est le nombre de pixels. On vérifie que le nombre d'images possibles est $N = (2^{24})^{4 \times 2^{20}}$. L'entropie $H(X^n)$ de cette source est nécessairement inférieure à $\log_2 N = 4 \times 2^{20} \times 24$ et il est donc possible de coder ces images en moins de $4 \times 2^{20} \times 24$ bits en moyenne. Ceci suppose cependant de connaître la distribution de probabilités de la source qui est inconnue. Il va donc falloir procéder sans connaître cette distribution.

Codage de type dictionnaire

En 1977, Lempel et Ziv propose un algorithme de codage universel : leur algorithme permet automatiquement de coder une suite de n symboles issues d'une même source X avec une longueur moyenne qui tend vers la longueur optimale $\bar{H}(X) = \lim_{n \rightarrow +\infty} \frac{1}{n} H(X^n)$ sous une simple hypothèse de stationnarité de la source. On assure ainsi l'optimalité asymptotique de la longueur moyenne du code sans la connaissance de la loi de la source.

L'algorithme proposé par Lempel et Ziv (LZ77) comme ses nombreuses variantes (LZ78, LZW, ...) est, de plus, simple à programmer et efficace numériquement. Le principe de ces méthodes est de parcourir la chaîne de symboles et de coder les nouvelles occurrences des sous-chaînes déjà observées par une simple référence à l'occurrence précédente. Le terme de codage de type dictionnaire provient de l'implémentation dans laquelle un dictionnaire de sous-chaîne est construit au fur et à mesure du parcours de la chaîne. Sans mentionner toute la littérature existante sur le sujet, il suffit de noter que c'est la famille utilisée dans les algorithmes de compression de données de type ZIP (ou ARJ) pour en mesurer le succès.

L'algorithme de compression d'image GIF est basé sur cette technique ou, plus précisément, sur la variante LZW proposée par Welsh en 1984. En 1987, CompuServe, le leader du moment des fournisseurs d'accès au réseau, introduit ce format afin d'accélérer la transmission des images sur celui-ci. Pour comprimer une image, il suffit de réordonner la matrice des pixels en une liste de valeurs et de comprimer cette liste à l'aide de l'algorithme LZW. Cette technique simple est suffisante pour obtenir un taux de compression de l'ordre de 2 et donc de réduire de moitié le temps de transmission.

Prédiction

Le taux de compression obtenu à l'aide d'un algorithme de compression universel n'est optimal qu'asymptotiquement et, bien qu'il soit impossible de franchir la barrière de $(\log_2 N) / \bar{H}(X)$, il est possible d'accélérer la convergence vers celle-ci.

Les méthodes prédictives proposent de transformer de manière réversible les chaînes de symboles en des chaînes plus simple pour l'algorithme de compression. Il s'agit d'aider celui-ci en introduisant un modèle *ad hoc* permettant de prédire chaque symbole apparaissant dans la chaîne en fonction des symboles précédents et de coder à l'aide de l'algorithme de codage universel l'erreur de prédiction au lieu du symbole lui-même.

Ceci suppose que les symboles possèdent une interprétation, ce qui est le cas pour les images. Il n'est donc pas étonnant que, en 1995, lorsque l'algorithme PNG, conçu comme une alternative non encombrée par des brevets de GIF, a été proposé, cette prédiction ait été ajoutée. Les modèles proposés utilisent des prédictions linéaires de l'intensité lumineuse en fonction de celle des voisins déjà connus. Les erreurs de prédictions sont alors codées par une autre variante de l'algorithme LZ77, l'algorithme deflate.

L'amélioration de performance est notable puisque l'on peut atteindre des taux de l'ordre de 3 avec l'algorithme PNG.

Codage statistique

Les techniques de codage universel ne font pas intervenir la distribution de probabilité de la source mais s'y adaptent asymptotiquement. A l'opposé, on a vu que si l'on connaît cette distribution, on peut construire un code quasi optimal. Deux algorithmes se disputent la prédominance pour ces codages dits statistiques : le plus ancien, l'algorithme de Huffman, est simple et efficace mais est moins performant que le plus récent, l'algorithme de compression arithmétique, dont la complexité est plus grande. Le choix entre ces deux algorithmes se fait suivant les contraintes de performance et de complexité.

La voie de la modélisation statistique est une voie intermédiaire. Elle remplace la distribution inconnue par un modèle connu et utilise cette nouvelle distribution pour coder les symboles à l'aide d'un algorithme de codage statistique. Si le modèle n'est pas trop éloigné de la réalité (au sens de la distance de Kullback-Leibler), il permet de compresser les données efficacement sans avoir recours à un comportement asymptotique.

Les modèles utilisés varient en complexité : les modèles les plus simples font l'hypothèse que tous les éléments de la chaîne sont indépendants et identiquement distribués tandis que les modèles les plus complexes conditionnent le choix de la distribution de probabilités pour un nouveau symbole à tous ceux déjà codés dans le passé.

Les meilleurs résultats de compression sans perte sont obtenus avec des modèles contextuels de type chaîne de Markov où la loi utilisée dépend du voisinage et est apprise au fur et à mesure du parcours de l'image. Ces modèles permettent des taux de compression dépassant 4 au prix d'un algorithme complexe et lent.

Compression avec perte

Ce taux de compression d'un facteur 4 n'est pas toujours suffisant pour la transmission et le stockage des images numériques de grandes tailles. Pour l'améliorer, il va falloir perdre quelque chose : dégrader l'image. Ceci permet d'atteindre des taux arbitrairement grand au prix d'une dégradation toujours plus importante. L'objectif des algorithmes de compression avec perte est de minimiser cette dégradation pour un taux de compression donné.

Dégradation et Quantification

La clé de la compression avec perte est dans une modification non réversible de la source permettant d'obtenir une nouvelle source dont l'entropie est plus faible.

La modification la plus simple pour les images est, sans doute, le changement de résolution : en diminuant le nombre de pixels, l'entropie de la source est diminuée. Cette astuce est largement utilisée, les images transmises le sont, le plus souvent, à une résolution adaptée au récepteur. Cependant, si l'on souhaite conserver la résolution, ce principe doit être abandonné. Une alternative simple est alors de réduire le nombre de couleurs possibles pour chaque pixel.

Cette phase qui consiste à utiliser une partition de l'espace des couleurs et à remplacer chaque couleur par un représentant de la classe à laquelle elle appartient a pour nom la quantification. L'incarnation la plus élémentaire de cette modification est la quantification scalaire uniforme : toute valeur x de luminosité dans l'intervalle $[(n - .5)\Delta, (n + .5)\Delta[$ est remplacée par $n\Delta$ où Δ est un pas de quantification à choisir et codée par l'entier n . On note que ceci a déjà été utilisé de manière implicite puisque l'intensité lumineuse est une valeur continue et qu'elle a déjà été quantifiée par des valeurs entières entre 0 et 255. L'effet de cette quantification est assez rapidement trop perturbant pour qu'elle soit utilisée : elle engendre des *aplats* désagréables pour l'oeil. D'autres choix de partitions pour la luminosité sont possibles et la question de l'optimisation de cette partition se pose.

Cette recherche d'une bonne partition est encore plus cruciale lorsque la quantification des couleurs est vectorielle : la bonne partition se trouve parmi celles de l'ensemble des couleurs et non plus celles de la luminosité canaux par canaux. Cette opération dite de palettisation fait par exemple partie intégrante de l'algorithme GIF qui débute par une partition de l'ensemble des couleurs en 256 classes. On constate une amélioration par rapport à la quantification scalaire mais la encore la limite perceptuelle se rencontre assez vite.

Enfin, la quantification peut-être réalisée en travaillant sur des blocs de pixels plutôt que de travailler pixel par pixel. La recherche d'une partition optimale de l'ensemble des blocs possibles pose alors un problème de complexité mais améliore encore les résultats. Cette technique n'est pas très utilisée car en ajoutant un ingrédient la quantification scalaire suffit.

Transformation

La quantification scalaire devient en effet suffisante si elle est précédée d'un changement de base orthonormée. Les valeurs à quantifier ne sont alors plus les couleurs des pixels mais les coordonnées de l'image dans cette nouvelle base. Ceci crée de manière simple une partition appropriée non pas de l'ensemble des couleurs mais directement de l'ensemble des images pourvu que la base soit bien choisie. L'image est alors reconstruite à l'aide des coordonnées quantifiées.

Ainsi, pour $\{b_n\}_{n \in N}$ une base orthonormée de l'ensemble des images d'énergie finie (l^2) et I l'une de ces images, la relation entre I et ses coefficients c_n dans la base $\{b_n\}$ est donnée par la formule classique suivante

$$I = \sum_{n \in N} c_n b_n \text{ avec } c_n = \langle I, b_n \rangle.$$

Le codage par transformée est obtenu en quantifiant les coefficients par un quantificateur Q_n connu et en codant, sans perte cette fois, les coefficients quantifiés par des méthodes similaires à celles de la section précédente. L'image dégradée \tilde{I} effectivement compressée est alors donnée par

$$\tilde{I} = \sum_{n \in N} Q_n(c_n) b_n.$$

L'application la plus connue de ce principe est sans doute l'algorithme JPEG proposé en 1994 par un comité d'experts (Joint Picture Expert Group). Ce standard possède un statut de quasi monopole pour la compression avec perte des images couleurs et on verra que même le nouveau standard JPEG 2000 a du mal à le détrôner. La base choisie

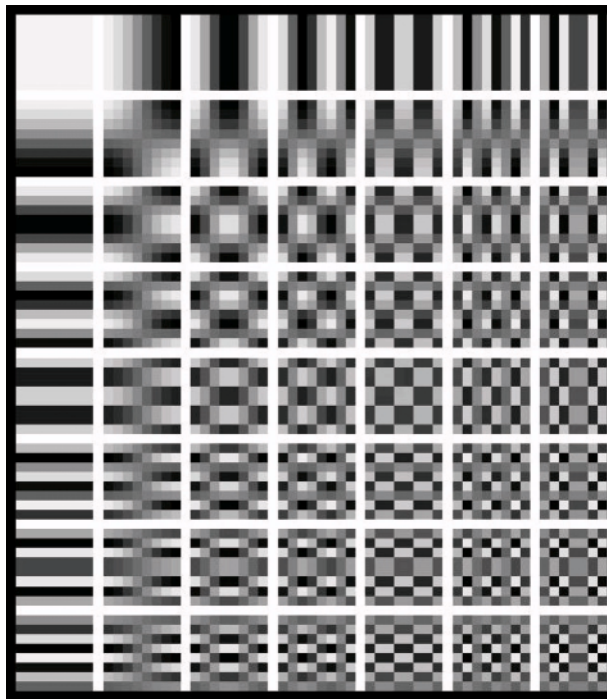


Figure 2 – Base de la DCT-IV : chacun des 64 carrés correspond à un élément de la base utilisée dans l'algorithme JPEG. Le caractère oscillant des éléments (la fréquence) augmente lorsqu'on se déplace sur la droite ou vers le bas.

par le comité est dérivée des bases de Fourier : l'image est découpée en blocs de taille 8×8 et chacun de ces sous-blocs est décomposé dans une base $\{b_{j_1, j_2}\}$ de cosinus locaux (DCT-IV) représentée dans la figure 2 :

$$b_{j_1, j_2}[k_1, k_2] = \frac{1}{4} \cos\left(\frac{\pi}{8}\left(j_1 + \frac{1}{2}\right)\left(k_1 + \frac{1}{2}\right)\right) \cos\left(\frac{\pi}{8}\left(j_2 + \frac{1}{2}\right)\left(k_2 + \frac{1}{2}\right)\right).$$

Ces coefficients sont alors quantifiés et codés à l'aide d'un codage statistique (Huffman le plus souvent) dans lequel les coefficients sont modélisés comme indépendants. La transformée a un effet décorrélant qui justifie cette approximation. Un modèle psycho-visuel est de plus utilisé pour quantifier les coefficients selon leur importance visuelle. Cet algorithme est très efficace pour une large gamme de taux de compression mais présente l'inconvénient de faire apparaître ces blocs 8×8 à fort taux de compression.

Le choix de la base est crucial pour les algorithmes par transformée comme on va le montrer maintenant avec une petite étude théorique.

Codage et approximation non linéaire dans des bases

Nous allons ici construire un codeur par transformée très simple mais qui permet de saisir les enjeux du choix de la base et sa relation avec la théorie mathématique de l'approximation : l'image I est transformée en ses coefficients dans la base $\{b_n\}$, ceux-ci sont quantifiés de manière uniforme à l'aide d'un pas Δ et l'on code alors les valeurs quantifiées par deux listes : une liste binaire donnant pour chaque coefficient s'il est nul ou non et une liste des valeurs quantifiées non nuls. Cette stratégie s'explique par le caractère particulier des coefficients quantifiés à 0.

En effet, en reprenant les notations précédentes, l'erreur introduite par la compression est donnée par la différence entre I et \tilde{I} se mesure aisément en norme quadratique :

$$\|I - \tilde{I}\|^2 = \sum_{\substack{n \in N \\ |c_n| \leq \Delta/2}} c_n^2 + \sum_{\substack{n \in N \\ |c_n| > \Delta/2}} (c_n - Q(c_n))^2$$

et en introduisant M_Δ le nombre de coefficients c_n tel que $|c_n| \geq \Delta/2$

$$\|I - \tilde{I}\|^2 \leq \|I - I_{M_\Delta}\|^2 + M_\Delta \Delta^2/4$$

où I_{M_Δ} est obtenu à partir de I en conservant les M_Δ plus grands coefficients en valeurs absolues.

L'étude du terme $\|I - I_{M_\Delta}\|^2$ est le coeur de la théorie de l'approximation dans les bases orthonormées. Pour approcher une fonction (une image) avec M coefficients à choisir librement pour minimiser l'erreur quadratique de reconstruction, la bonne stratégie est de conserver les M plus grands coefficients en valeurs absolues. L'un des objets de la théorie de l'approximation est d'étudier les possibilités d'approximations de classes de fonctions dans une base donnée. Ceci s'exprime bien souvent par une relation entre l'erreur d'approximation $\|I - I_M\|^2$, M et Δ , le seuil associé, de la forme : $\|I - I_M\|^2 \leq CM^{-\gamma}$ et $M_\Delta \leq C' \Delta^{-2\gamma/(\gamma+1)}$, où α est lié à une forme de régularité propre à la classe. L'erreur de compression satisfait alors

$$\|I - \tilde{I}\|^2 \leq C'' M_\Delta^{-\gamma}$$

La taille du code nécessaire pour spécifier les coefficients quantifiés est également reliée à cette quantité M_Δ . La proportion de coefficients quantifiés non nuls est de M_Δ/N . L'entropie de la carte binaire de non nullité des coefficients est donc, si les coefficients sont considérés comme indépendants, de $-M_\Delta \log_2(M_\Delta/N) - (N - M_\Delta) \log_2(1 - M_\Delta/N)$. Enfin, chacun des M_Δ coefficients non quantifiés à 0 requiert au plus $C - \log_2 \Delta$ bits pour être spécifiés. Il en résulte, après un développement limité en M_Δ/N , que le nombre total R de bits nécessaire pour coder l'image satisfait

$$R \simeq M_\Delta(1 + \log_2(M_\Delta/N) + C - \log \Delta) \\ \simeq M_\Delta(C' + \log_2(M_D/N)).$$

La combinaison des estimations de $\|I - \tilde{I}\|^2$ et R donne alors

$$\|I - \tilde{I}\|^2 \leq C R^{-\gamma} \log_2^{\gamma+2}(R)$$

de sorte que l'efficacité de cet algorithme de codage est lié à une performance d'approximation non linéaire dans la théorie de l'approximation. Les performances de l'algorithme JPEG sont ainsi reliées à la capacité de la base de Fourier à approcher les fonctions régulières. La base de Fourier n'est cependant pas optimale pour les images et les bases d'ondelettes, introduites plus récemment, possèdent de meilleures propriétés. C'est donc tout naturellement qu'elles ont été utilisées dans le nouveau standard JPEG 2000.

Ondelettes et JPEG 2000

Les performances des ondelettes, introduites par S. Mallat en 1989, ont pour origine une adaptativité à la taille des structures des images inaccessible pour la DCT et ses blocs 8×8 . Elles s'obtiennent par le procédé récursif de la figure 3. A chaque étape, l'image basse résolution est décomposée en une image de plus basse résolution vivant dans un espace de dimension 4 fois plus petite et une image de détails. Chacune de ces composantes est représentée à l'aide de fonctions de bases dont la taille double à chaque étape. Ainsi, la base finale comprend des fonctions de supports variés : de fonctions à large support pour les grandes tendances à des fonctions à support très petits pour des détails très précis en passant par les situations intermédiaires. Le tour de force de cette construction est de rester dans le cadre des bases orthonormées : la technique de codage par transformée s'applique immédiatement. Pour être un peu plus précis, les bases d'ondelettes sont construites à partir de deux fonctions monodimensionnelles : une fonction d'échelle ϕ , qui est d'intégrale 1, et une ondelette ψ de moyenne nulle. Trois ondelettes bidimensionnelles sont obtenues par simple produit tensoriel comprenant au moins une ondelette : $\phi(x_1)\psi(x_2)$, $\psi(x_1)\phi(x_2)$, $\psi(x_1)\psi(x_2)$. Pour un bon choix de ϕ et ψ , la famille obtenue en dilatant ces fonctions par des puissances de 2 et en les translatant de manière adaptée

$$\left\{ \frac{1}{2^j} \phi\left(\frac{x_1 - k_1 2^j}{2^j}\right) \psi\left(\frac{x_2 - k_2 2^j}{2^j}\right), \frac{1}{2^j} \psi\left(\frac{x_1 - k_1 2^j}{2^j}\right) \phi\left(\frac{x_2 - k_2 2^j}{2^j}\right), \right. \\ \left. \frac{1}{2^j} \psi\left(\frac{x_1 - k_1 2^j}{2^j}\right) \psi\left(\frac{x_2 - k_2 2^j}{2^j}\right) \right\}_{j \in \mathbb{Z}, k_1 \in \mathbb{Z}, k_2 \in \mathbb{Z}}$$

constitue une base orthonormale d'ondelettes.

Ainsi, l'algorithme JPEG 2000 proposé fin 2000 par le comité JPEG, mais terminé uniquement en 2002, débute par une décomposition de l'image dans une base d'ondelettes. Les coefficients ainsi obtenus sont alors quantifiés et codés par plan de bits à l'aide d'un codage arithmétique utilisant un modèle contextuel de dépendance entre les coefficients. Celui-ci permet une amélioration de la qualité par rapport à JPEG pour un même taux de compression mais surtout offre de nouvelles possibilités pratiques : transmission progressive, région d'intérêt. Bien qu'il s'agisse d'un standard libre de droit, son utilisation reste encore limitée. L'une des raisons est sans doute que pour des taux de compression de moins de 10 la différence avec l'algorithme JPEG n'est pas toujours visible. Pour des images géométriques qui constituent un modèle pour les images naturelles, il est de plus prouvé que ni les bases issues de Fourier ni même les ondelettes ne sont optimales. Elles sont incapables en effet d'exploiter la composante géométrique de celles-ci.

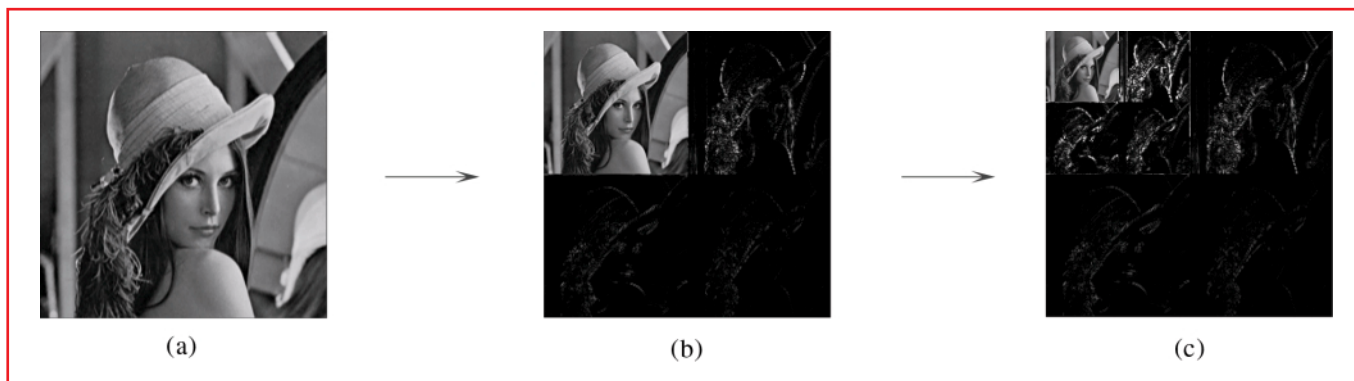


Figure 3 – Transformation en ondelettes. Deux étapes de la transformation en ondelettes sont représentées ici. L'image (a) est transformée en une image basse résolution (en haut à gauche de l'image (b)) et trois sous images de détails. Chacune de ces images correspond à l'amplitude des coefficients dans une base d'ondelettes. L'image (c) de coefficients est obtenues en répétant ce procédé pour l'image basse résolution de (b). Ceci correspond à la structure multi-résolution des bases d'ondelettes.

Et la géométrie ?

La géométrie est une des caractéristiques essentielles des images naturelles : elle constitue un élément de régularité qui n'est pas pris en compte par les bases classiques. L'exploitation de cette régularité géométrique est ainsi une direction prometteuse pour la compression d'image et plus généralement pour le traitement des images. L'objectif de cette section est de présenter un bref panorama des directions explorées actuellement. Elle débute cependant par un cas plus simple, celui des signaux monodimensionnels, qui permet de présenter les enjeux de l'exploitation de la régularité.

Signaux monodimensionnels et régularités

Les performances de l'algorithme de compression présenté dans la section 3.3 sont contrôlées par la vitesse de décroissance de l'approximation non linéaire du signal dans la base. L'erreur obtenue en ne conservant que les grands coefficients et en quantifiant les coefficients restants est en effet du même ordre de grandeur que celle obtenue sans la quantification. Cette décroissance dépend à la fois du signal considéré et de la base utilisée.

On étudie classiquement la décroissance de cette erreur pour une base donnée en fonction de l'appartenance à une classe de régularité. Pour les signaux monodimensionnels, la régularité est souvent mesurée par l'appartenance à la classe C^α des fonctions α fois dérivables et dont la dérivée d'ordre α est continue (cette définition valable pour α entier s'étend au cas α positif). On démontre que si f est C^α avec $\alpha \geq 1$ alors l'erreur d'approximation obtenue en conservant les M plus grands produits scalaires $\|f - f_M\|^2$ décroît comme $M^{-2\alpha}$ que se soit avec la base de Fourier ou avec une base d'ondelettes. Cette décroissance s'avère de plus optimale : il est impossible de trouver une base dans laquelle la décroissance est plus rapide.

Si l'on considère maintenant des signaux qui ne sont plus que C^α par morceaux, les deux bases induisent des comportements différents. Dans la base de Fourier, la décroissance de l'erreur ne peut être bornée que par M^{-1} tandis que, grâce à la structure de multirésolution de la base d'ondelettes, la décroissance de l'erreur en $M^{-2\alpha}$ est optimale.

La construction d'une nouvelle base, celle des ondelettes, a permis d'exploiter la régularité de manière plus fine que ce que ne permet l'autre base, la base de Fourier. La situation pour les images est similaire.

Modèle géométrique des images

Si l'on considère des images uniformément régulières, C^α , on vérifie que les bases de Fourier ainsi que celles d'ondelettes permettent d'obtenir une décroissance optimale de l'erreur de $M^{-\alpha}$. Ce modèle est cependant trop grossier pour les images puisque celles-ci sont plutôt régulières par morceaux.

Un modèle géométrique d'image simple a été proposé par Tsybakov et Korostelev : une image géométrique est obtenue par discrétisation d'une fonction régulière C^α en dehors de contours eux-mêmes réguliers C^α . Ce modèle simple néglige la partie texturée des images mais donne un cadre raisonnable pour comparer les différentes méthodes d'un point de vue théorique.

On montre que les bases d'ondelettes, bien que plus efficaces que la base de Fourier, sont incapables de capturer la régularité géométrique des contours : les ondelettes qui touchent les bords donnent de grands coefficients et leur nombre important limite fortement la décroissance de l'erreur qui se comporte comme M^{-1} alors que la décroissance optimale est en $M^{-\alpha}$. Une première approche plus géométrique des images est possible en abandonnant l'idée de base et en recherchant une triangulation adaptée. A l'aide de triangles allongés s'adaptant aux contours, le nombre de paramètres nécessaires est beaucoup plus petit que dans le cas des ondelettes.

La recherche d'une triangulation optimale et son codage efficace reste cependant une question ouverte. L'idéal serait d'exhiber une transformation présentant un comportement similaire en terme de nombre total de paramètres.

Curvelets, Wedgelets, ...

En 1999, Candès et Donoho ont construit une transformation en curvelets réalisant quasiment ce programme. Ses éléments de base sont des structures anisotropes à des échelles, des orientations et des positions variées. Ils forment une structure multirésolution similaire à celle des bases d'ondelettes à laquelle est été ajouté un filtrage orienté et permettent de capturer la régularité des fonctions de type $C^2 - C^2$. La décroissance de l'erreur d'approximation pour cette classe de fonction se comporte de manière quasi optimale en $\log M M^{-2}$.

Cependant la structure obtenue n'est pas celle d'une base mais d'un repère oblique (frame) : la famille est génératrice mais elle n'est pas libre. Ceci implique une certaine redondance dans la représentation et constitue un handicap pour un algorithme de compression. Cette construction est de plus continue et en obtenir une discrétisation efficace est un problème difficile.

Baraniuk *et al.* propose une direction différente pour l'utilisation de la géométrie. La transformée utilisée est une transformée en ondelettes classique et la géométrie n'intervient que pour définir un contexte pour le codage des coefficients. Les wedgelets sont des éléments de contours rectilignes placés sur une grille dyadique. Chacun d'entre eux fournit un contexte pour un grand nombre d'ondelettes dans son voisinage. Un algorithme d'optimisation de cette géométrie existe et l'algorithme de compression ainsi obtenu est asymptotiquement plus efficace que l'algorithme classique de compression en ondelette qui n'incorpore pas la géométrie. Il a les mêmes performances qu'un algorithme basé sur une transformée en curvelets.

Bandelettes

Introduites avec S. Mallat, elles s'adaptent aux images utilisées. Les bases de bandelettes sont des bases indicées par une géométrie : leurs vecteurs de bases sont allongés le long de courbes qui sont choisies pour épouser au mieux les contours. Plus précisément, l'image est segmentée en carrés dyadiques et chacun de ces carrés est muni d'une base adaptée. Ces bases locales sont construites à partir d'une base d'ondelettes anisotropes (obtenues par un simple produit tensoriel de deux bases monodimensionnelles) déformée pour suivre le contour principal dans chaque carré. La figure 4 illustre l'intérêt de cette représentation.

Le choix de la segmentation et de sa géométrie, et donc de la base utilisée, dépend alors de l'image à comprimer et du taux de compression. Le surcoût de la spécification de cette base *via* la géométrie est pris en compte dans un algorithme rapide de recherche de meilleure base qui permet d'obtenir automatiquement le meilleur compromis entre l'adaptation de la base à l'image (le bon suivi des contours) et son coût de codage.

La décroissance de l'erreur d'approximation en bandelettes est optimale pour les fonctions régulières en dehors de contours réguliers : $M^{-\alpha}$ pour les fonctions $C^\alpha - C^\alpha$. Ce résultat théorique s'accompagne de bonnes performances pour les images réelles.

Les méthodes géométriques de compression d'images constituent une direction de recherche très dynamique, aussi ce panorama est loin d'être complet. On pourrait mentionner la construction de transformations qui s'adaptent automatiquement à la géométrie sans le besoin de spécifier celle-ci ou encore des approches de géométrie discrètes, ... Elles s'accordent cependant toutes sur le fait que la géométrie est la clé pour améliorer significativement les méthodes actuelles de compression.

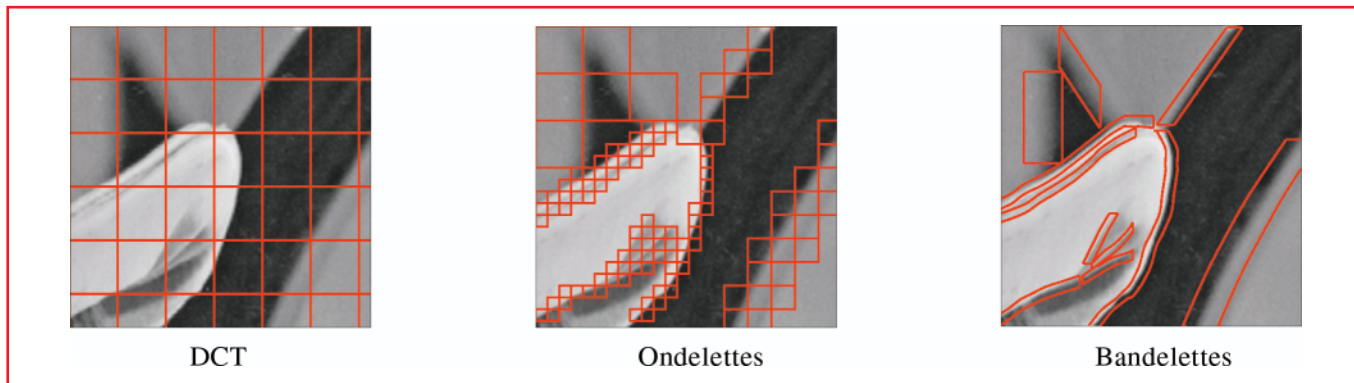


Figure 4 – DCT, Ondelettes et Bandelettes. Cette figure illustre l'évolution des représentations des images. La DCT correspond à un découpage uniforme de l'image tandis que les ondelettes permettent d'adapter la taille des structures utilisées à celles présentes dans l'image. Enfin, les bandelettes s'adaptent à la composante géométrique des images.

Pour en savoir plus

- [1] CANDÈS (E.), DONOHO (D.), Curvelets : A Surprisingly Effective Nonadaptive Representation of Objects with Edges, in Schumaker L. L., Cohen A. and Rabut C., *Curves and Surfaces fitting*, Vanderbilt University Press (1999).
- [2] DO (M. N.), VETTERLI (M.), Contourlets, in Stoeckler J. and Welland G. V. *Beyond Wavelets*, Academic Press, (2003).
- [3] LE PENNEC (E.), MALLAT (S.), Sparse Geometrical Image Representation with Bandelets, *IEEE Transaction on Image Processing* (2005).
- [4] MALLAT (S.), *A wavelet tour of signal processing*, Academic Press, 2nd edition (1998).
- [5] NELSON (M.), GAILLY (J.-L.), *The data compression book (2nd ed.)*, MIS :Press, New York, NY, USA (1996).
- [6] SHANNON (C. E.), A mathematical theory of communication, *Bell System Technical Journal* (1948).
- [7] WAKIN (M.), ROMBERG (J.), CHOI (H.), BARANIUK (R.), Rate-Distortion Optimized Image Compression using Wedgelets, in *IEEE Internat. Conf. on Image Processing* (2002 sep.).
- [8] WELCH (T.), A Technique for High-Performance Data Compression, *Computer* (1984).
- [9] ZIV (J.), LEMPEL (A.), A Universal Algorithm for Sequential Data Compression, *IEEE Transactions on Information Theory* (1977).