

Du photon au pixel
L'appareil photographique numérique
seconde édition

Henri Maître

20 décembre 2016

Table des matières

Chapitre 1. La représentation des images : codage et formats	7
1.1. Format « naïf » et métadonnées	8
1.2. Le format RAW : brut ou natif	9
1.2.1. Que contient un format natif (RAW) ?	12
1.2.2. Intérêt du format natif	14
1.2.3. Les limites du format natif	14
1.2.4. Vers une normalisation du format natif	15
1.2.4.1. Format natif <i>via</i> TIFF/EP	15
1.2.4.2. Format natif <i>via</i> DNG	16
1.3. Les métadonnées	16
1.3.1. Le standard XMP	17
1.3.2. Le format de métadonnées Exif	17
1.3.2.1. Les données du fichier Exif	18
1.3.2.2. Limitations du fichier Exif	19
1.4. Les formats comprimés sans perte	19
1.4.1. Les algorithmes généraux de codage sans perte	20
1.4.2. Les codages sans perte de JPEG	21
1.5. Les formats d'image issus des arts graphiques	22
1.5.1. Le format PNG	22
1.5.2. Le format TIFF	23
1.5.3. Le format GIF	25
1.6. Les formats de compression d'images avec pertes	25
1.6.1. JPEG	26
1.6.1.1. Les principes du codage JPEG	26
1.6.1.2. JPEG en pratique	30
1.6.2. JPEG 2000	31
1.6.2.1. Principe de JPEG 2000	32
1.6.2.2. Les performances de JPEG 2000	34

6 Du photon au pixel

1.7. Les formats tuilés	35
1.8. Le codage vidéo	37
1.8.1. Vidéo et normalisation	37
1.8.2. Codage MPEG	38
1.8.2.1. Les mécanismes mis en œuvre dans MPEG	38
1.9. Codage comprimé (<i>Compressed sensing</i>)	41
Bibliographie	45
Index	48

Chapitre 1

La représentation des images : codage et formats

La diversité des formats d'archivage et de présentation des images constitue l'un des freins à leur usage au sein de la société. Si le format JPEG a acquis aujourd'hui pour l'utilisateur quotidien de la photographie, et plus encore pour l'utilisateur occasionnel, un statut particulier qui lui donne, de fait, l'apparence d'une norme, de nombreux autres formats sont disponibles, et souvent préférables.

Le recours à ces formats est dicté par des objectifs différents et fréquemment contradictoires : soit diminuer le volume de l'image, soit mieux respecter ses propriétés (dynamique, palette chromatique, géométrie des lignes, qualité des aplats, etc.). Des formats sont par ailleurs adaptés à l'archivage des images, d'autres à leur transmission, d'autres enfin à l'interopérabilité de logiciels différents de traitement ou de visualisation. Les applications concernées par l'image sont très nombreuses et elles sont aussi très diversifiées. Certaines exigent des échanges de documents très rapides tandis que d'autres s'attachent à leur qualité, qualité qui pourra être jugée avec des critères différents selon les domaines : respect des nuances ou précision des détails, finesse des tracés, reproductibilité des couleurs, lisibilité des textes, capacité à entrer dans une composition complexe, etc.

Rappelons qu'une photo ordinaire est aujourd'hui constituée d'une dizaine de millions de pixels au moins, chacun représenté souvent par trois octets, un par canal chromatique, rouge, vert ou bleu. Si l'on ne prend pas de mesure pour comprimer le signal, on est donc amené à stocker, aussi bien dans l'appareil que dans l'ordinateur, quelques dizaines de millions d'octets par image. Par ailleurs, il est devenu très simple d'enregistrer des centaines, voire des milliers d'images à l'occasion d'un événement. Un amateur passionné ou un professionnel maintient donc des centaines de

giga-octets d'images dans ses mémoires. Au-delà du coût de l'archivage, la consultation et le transfert en sont rendus plus coûteux et plus longs. Les nombreux formats développés que nous allons examiner dans ce chapitre reflètent ces besoins. On trouvera par exemple dans [BAR 03b, WOO 11] des analyses plus techniques de chacun d'eux.

1.1. Format « naïf » et métadonnées

Avant même que les utilisateurs disposent de moyens simples pour saisir des photos numériques, s'est posée la question du format de représentation d'une image en ordinateur.

Pour une image en Noir et Blanc de n_l lignes de n_p points, où chaque pixel est codé par un octet, la façon la plus simple consiste à placer dans un fichier unique le train des $N = n_p n_l$ octets. Ce format naïf n'est pas sans ambiguïté puisqu'il y a généralement de nombreuses façons de construire une image rectangulaire à partir de N pixels. Si l'on ne veut pas réserver des niveaux de gris au marquage des fins de lignes, il faut donc transmettre également les valeurs n_p et n_l . Ces valeurs n_p et n_l sont les plus simples des métadonnées qui doivent accompagner l'image pour qu'elle soit lisible, mais de nombreuses autres apparaissent au fur et à mesure que l'on diversifie les images que l'on souhaite traiter : le mode de conversion des octets en niveau de gris (code naturel, code de Gray ou code faisant référence à une table de transcodage¹), la façon de représenter une image en couleur (par l'association de trois plans d'images ou par l'entrelacement des trois octets de chaque pixel le long d'une ligne), le type de codage utilisé, le nombre de canaux par pixels (pour les images multispectrales) et d'octets par pixel (pour les images à haute dynamique), le nombre de dimensions de l'image (pour les applications médicales par exemple), etc. Par la suite, ces métadonnées ont recueilli également des éléments d'identification et de propriété ainsi que les informations sur les conditions de leur acquisition, puis, au cours de la vie de l'image, sur les traitements qu'elle a reçus. Pour le photographe, les métadonnées sont donc naturellement le lieu où placer les paramètres de la prise de vue qui pourront l'aider à tirer le meilleur parti de sa photo.

Les métadonnées ont parfois été placées dans un second fichier (dit de format), qui accompagne l'image. Cette solution est encore parfois employée lorsque les métadonnées comportent une grande quantité d'informations étrangères à l'image elle-même (par exemple des informations sur le patient en imagerie médicale, ou sur la mission en imagerie satellitaire astronomique). Séparer fichier image et fichier des métadonnées

1. Une table de transcodage (= LUT, *look-up table* en anglais) permet d'affecter à un pixel de valeur ν , le niveau ou la couleur $T(\nu)$ choisi arbitrairement dans une palette.

permet de traiter de façon plus confidentielle l'un ou l'autre des fichiers. Il fait également courir moins de risques qu'une opération erronée sur l'un des fichiers dégrade indument l'autre.

Cette solution n'a pas été retenue pour l'image numérique de la photographie grand public. métadonnées et données image sont réunies dans un fichier unique contenant donc des informations de types très différents, certaines binaires, d'autres symboliques. Les métadonnées sont généralement placées en tête des données, selon un format syntaxiquement défini et identifié par l'extension du nom de fichier (*.jpg*, ou *.tif*, ou *.bmp*). La lecture de ces métadonnées fournit les clés pour lire et afficher correctement le champ image. Elle permet aussi, à qui sait en tirer profit, d'utiliser les informations placées là par le fabricant de la caméra ou par un utilisateur antérieur de l'image, pour en exploiter mieux les propriétés.

La variété et la complexité des formats d'image provient essentiellement de l'importance accordée par certaines familles d'utilisateurs à tel ou tel type de propriété, importance qui leur fait attribuer une place et un rôle plus ou moins grand au sein des métadonnées.

1.2. Le format RAW : brut ou natif

Enregistrer le signal tel qu'il est saisi par le capteur est une solution de bon sens si l'on prend soin d'enregistrer aussi les paramètres indispensables à son traitement ultérieur. Cela conduit à réserver au signal un nombre de bits supérieur à ce qui est strictement nécessaire à l'image, à conserver la dynamique d'origine du capteur et les éléments de reconstruction de l'image, depuis le courant d'obscurité et les gains des amplificateurs jusqu'aux valeurs de saturation, à garder la géométrie imposée par l'objectif et par le mosaïcage, à enregistrer également les paramètres de la prise de vue, temps de pose, ouverture, réglages divers avec la colorimétrie spécifiques des filtres chromatiques, de façon à pouvoir ensuite procéder à toutes les corrections pour fournir une image de la meilleure qualité possible. Le champ des métadonnées est naturellement là pour recevoir ces informations.

Voici la philosophie de ce que l'on appelle le format natif ou RAW. Mais ce format n'est en fait pas un format car il n'est pas unique et pas partageable comme nous le verrons. Nous emploierons cependant souvent ces termes « format natif ou RAW », suivant en cela un usage fréquent, lorsqu'il n'y aura pas d'ambiguïté.

Le format natif est une facilité des fabricants d'appareils qui peuvent enregistrer le signal dès la sortie du capteur, dans cette toute première étape qui réclame beaucoup de rapidité, sans se préoccuper des opérations nécessaires à sa visualisation et son échange avec d'autres utilisateurs. Le fichier obtenu est l'image fidèle des mesures

faites sur le capteur (voir la figure 1.1). Des traitements ultérieurs, au sein de l'appareil ou dans un ordinateur déporté, permettront de tirer parti au mieux de ces données. Mais c'est un format très dépendant du constructeur et les images ainsi enregistrées n'avaient initialement pas vocation à être conservées dans ce format hors de la chaîne de production de l'appareil photographique (embarquée ou déportée). C'est leur succès qui a rendu leur usage populaire, mais ce « format » reste marqué des défauts de son destin premier.

Certes, il existe une norme standardisée par l'ISO (nous en parlerons à la section 1.2.4), qui tente de faire partager les fichiers entre utilisateurs, néanmoins chaque fabricant maintient sa propre version et la fait évoluer au rythme de ses besoins. Les fichiers natifs se distinguent par une extension spécifique du nom du fichier (au nombre de plus d'une centaine aujourd'hui²) et non pas par l'extension *.raw*, comme on le souhaiterait pour un format partagé. Ces formats ne sont généralement pas lisibles avec les outils ordinaires d'affichage (certains sont même cryptés) mais beaucoup de bibliothèques de traitement d'images reconnaissent les formats les plus courants et l'on trouve sur Internet les mises à jour de ces algorithmes de lecture au fur et à mesure de leur évolution.

Ces formats ne sont pas universels pour de multiples raisons. Tout d'abord, ils reflètent la géométrie du photodétecteur (nous avons vu leur variété à la section ??), ensuite ils sont adaptés à l'architecture employée pour prendre en compte les bruits exacts qui affectent le signal (courant d'obscurité, courant de recharge, bruit des transistors de transferts, présence d'amplificateurs partagés, etc.) et fournir les bons paramètres aux équations de filtrage qui réduisent les bruits (comme par exemple l'équation (??)), ils expriment différemment les opérations de colorimétrie en fonction de leurs propres filtres chromatiques ; enfin ils traitent à leur façon les défauts connus, pixels aveugles, aberrations géométriques, vignettage, etc.

La transformation d'un fichier natif en un format plus universel (comme JPEG par exemple) est assurée par le fabricant de l'appareil photo lors du transfert de la mémoire de l'appareil à un ordinateur. Cette opération est également souvent nécessaire pour afficher l'image sur l'écran de l'appareil. La solution mise en œuvre au sein de l'appareil, ou déportée sur l'ordinateur hôte par un logiciel maison, est souvent la meilleure conversion disponible car le concepteur de l'appareil connaît exactement les paramètres qui sont nécessaires pour exploiter le signal et il sait également où trouver, dans les données ancillaires associées à l'image, les paramètres dont il a besoin. Mais,

2. Ces extensions sont par exemple : *.cr2* ou *.crw* pour Canon, *.nef* ou *.nrw* pour Nikon, *.orf* pour Olympus, *.arw* pour Sony, *.ptx* pour Pentax, *.srw* pour Samsung, *.3fr* pour Hasselblad, *.raf* pour Fuji, etc. Notons que Leica et Panasonic ont l'un et l'autre adopté un *.raw* générique à côté de *.rw1* pour Leica et de *.rw2* pour Panasonic. Enfin *.dng*, proposé par Adobe, que nous verrons à la section 1.2.4, est utilisé par quelques fabricants (comme Leica) en plus des formats maison.

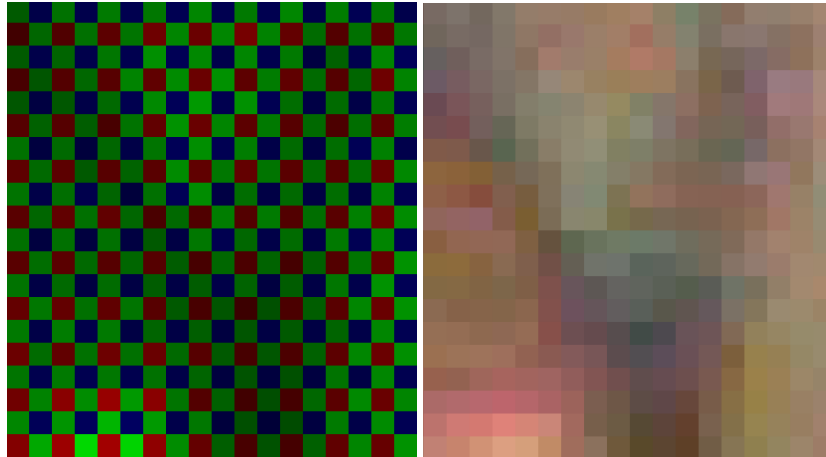


Figure 1.1 – A gauche, une toute petite portion de l’image en couleur de la figure 1.2 (le coin supérieur gauche), telle qu’elle est saisie par la matrice de Bayer et représentée dans un fichier natif (RAW), avec les trois canaux R, V et B superposés. Pour chaque pixel, une seule valeur R, V ou B est non nulle. A droite, cette image reconstruite par démosaïchage.

dans l’appareil, le temps est compté pour faire ces corrections et l’on peut parfois concevoir des corrections plus complexes qu’il faut alors réaliser sur un ordinateur en temps différé. On peut également s’écarter du choix du fabricant qui a pu privilégier des traitements (renforcement de l’acutance ou accentuation des couleurs) au-delà de l’attente de l’utilisateur. On a alors recours à un traitement spécifique de transformation de l’image native que l’on trouve soit dans un logiciel libre, universellement utilisé, qui fournit un outil de décodage pour tous les appareils du commerce : DCraw³, soit dans des boîtes à outils plus générales (produits libres comme RawTherapee⁴, ou produits propriétaires, PhotoShop, LightRoom, DxO Lab, etc.), qui mettent à la disposition de l’utilisateur une très large famille de traitements.

Des bibliothèques de traitement d’image proposent des algorithmes différents de ceux du fabricant qui conduisent alors à des reconstructions potentiellement différentes : elles peuvent être meilleures car des algorithmes plus complexes peuvent être mis en œuvre, par exemple pour la colorimétrie, le filtrage du bruit ou le démosaïchage. Par ailleurs, ces bibliothèques peuvent intégrer des effets que la caméra ne prend

3. DCraw est développé par Dave Coffin. Ce logiciel dispose des formules de transformation de pratiquement tous les formats natifs. Il est également à la base de nombreux autres logiciels et boîtes à outils. Il est disponible sur [COF 14].

4. RawTherapee est un logiciel spécialisé pour traiter les fichiers natifs [HOR 14].

pas toujours en compte : vignettage optique, vignettage de pixel, aberrations géométriques, etc. La reconstruction est alors supérieure à celle que fournit le fabricant.

1.2.1. *Que contient un format natif (RAW) ?*

Le format natif doit refléter les données qui font la diversité des capteurs. Il est généralement constitué des éléments suivants :

- dans le champ des métadonnées :
 - un en-tête comportant l'ordre des octets (petit boutiste ou gros boutiste⁵), le nom de l'image, l'adresse des données brutes ;
 - les informations décrivant le capteur : nombre de pixels, type de matrice de Bayer (selon une matrice de Bayer, ou selon d'autres matrices, voir section ??), disposition des photosites ⁶ et profil colorimétrique qui exprime la composition des filtres chromatiques ;
 - les informations décrivant l'image : date, type d'appareil, d'objectif, temps de pose, ouverture, focale, choix des réglages, etc., éventuellement géolocalisation ;
 - les éléments de réglage permettant de faire la balance des blancs mesurés par les capteurs ou affichés par l'utilisateur ;
 - la dynamique du signal (entre 8 bits et 14 bits) et les paramètres de la chaîne électronique permettant de corriger ce signal (courant d'obscurité ou mesures permettant de le déterminer, biais sur le zéro, facteurs d'amplification (ou paramètres permettant de les obtenir à partir de la durée d'exposition), la sensibilité ISO, et la mesure d'énergie) ;
- dans le champ image :
 - une vignette qui accompagnera l'image et la représentera chaque fois qu'il faudra consulter la mémoire ;
 - éventuellement une image réduite JPEG permettant de l'afficher en une taille raisonnable ;
 - les données d'image présentées dans la géométrie du capteur (donc avec les RVB entrelacés dans un seul plan image) (voir la figure 1.2).

Comme un grand nombre des éléments de métadonnées sont communs à toutes les images, des formats types de ces métadonnées sont utilisés que nous verrons dans les prochaines lignes, (par exemple Exif). Mais dans le fichier natif se trouvent également des informations très spécifiques au capteur (en particulier sur les paramètres fins qui régissent l'électronique : bruit thermique, position de pixels morts, inhomogénéité des

5. Petit boutiste, gros boutiste ou petit endien, grand endien, traduction des termes anglo-saxons *little endian* et *big endian*, indiquant l'ordre des octets dans un mot de 16 bits.

6. La position exacte de la matrice de Bayer est donnée par un quadruplet qui indique l'ordre de rencontre des couleurs R, V et B au début des deux premières lignes de l'image ; par exemple GRGB, ou RGGB, où G (pour *green*) désigne le canal V.

gains, corrections de vignettage pixellique) que le fabricant ne souhaite pas toujours rendre public. Pour cela les métadonnées des fichiers natifs sont parfois cryptées de façon à ne pas être lues par des logiciels que ne contrôle pas le fabricant.

Enfin certains constructeurs ont choisi des formats natifs dont les fichiers images subissent un codage sans perte (type Lempel-Ziv).

Nous l'avons dit, le format natif est volumineux, c'est là un de ses défauts importants. Le fichier d'en-tête est de quelques milliers d'octets et donc généralement négligeable devant le fichier image qui constitue la partie volumineuse à archiver. Si l'image a n pixels, chaque pixel est représenté par un seul signal dans un des canaux R, V ou B, signal généralement codé sur deux octets. Sans compression la taille du fichier sera alors de l'ordre de $2n$ octets. Avec un algorithme de compression sans perte, on peut atteindre un taux de réduction de l'ordre de 2 et l'on retiendra qu'un fichier RAW occupe environ une place de n octets⁷. Cette compression, si elle a été retenue par le fabricant s'appliquera d'office. Les fichiers natifs occupent généralement de 10 Mo à 50 Mo par image, soit environ dix fois plus que le fichier JPEG correspondant.

1.2.2. Intérêt du format natif

Le format natif RAW est celui qui préserve le mieux la qualité du signal capturé par l'appareil. Par ailleurs, il fournit dans les métadonnées tous les éléments indispensables à un traitement efficace des images.

A partir de ces métadonnées, il sera possible de procéder à toutes les opérations de reconstruction du signal d'image sur 8 ou sur 16 bits pour toutes les applications, même les plus exigeantes en qualité comme le tirage sur papier, l'édition, la photocomposition, l'intégration dans des vidéos, etc. Une sortie de type JPEG pour un usage courant est bien sûr toujours possible, mais, si l'on n'a pas d'importantes contraintes de volume, on préférera souvent des formats disposant d'une plus grande dynamique : TIFF, DNG, PNG, etc., qui respectent mieux les détails fins de l'image.

Le passage par le format natif est indispensable à beaucoup d'applications avancées comme celles que nous verrons au chapitre ?? : haute dynamique de rendu, panoramas, accroissement de résolution et de mise au point.

On a souvent intérêt à faire les traitements de reconstruction dans des logiciels déportés dans un ordinateur. Ces logiciels, soit libres soit propriétaires, offrent en effet

7. Ce taux de compression sans perte de deux est élevé pour une image, mais il n'est pas anormal car le signal natif n'utilise généralement pas la totalité des seize bits de dynamique.

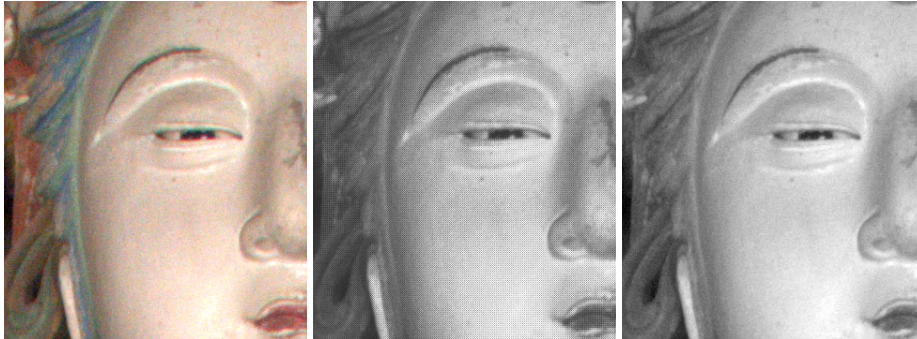


Figure 1.2 – A gauche, une image en couleur, au centre, le fichier natif (RAW) correspondant représenté en niveaux de gris (les pixels R, V et B sont entrelacés mais ont été ramenés en niveaux de gris, chacun codé sur un octet) et à droite l'image de luminance issue de l'image couleur. Comparant les deux images de gauche, on note que la géométrie est identique puisque les canaux R, V et B sont juxtaposés mais sous-échantillonnés. On voit également, sur le fichier du centre, la structure en quinconce spécifique du masque de Bayer, très sensible sur le visage où le vert est faible. Cette structure en quinconce est encore plus marquée sur la figure 1.1 qui est très fortement agrandie. On voit sur les lèvres que l'absence de vert (deux fois plus présent que le rouge et le bleu parmi les échantillons) se traduit par une plage sombre, tandis que le liseret un peu vert dans les cheveux apparaît plus lumineux que les cheveux eux-mêmes, bleus et rouges. L'image de luminance ne présente plus cette structure en quinconce car les canaux RVB ont été interpolés.

la possibilité de réaliser des post-traitements des images souvent exigés par les applications. Les post-traitements sont alors plus efficaces s'ils partent du signal source plutôt que d'un signal traité dans une perspective d'usage général.

1.2.3. Les limites du format natif

Si le format natif garantit la meilleure fidélité de rendu de l'image, on doit lui opposer les défauts suivants :

- le volume des images est souvent prohibitif ; pour un usage grand public, il multiplie par un facteur de 5 à 10 le stockage nécessaire ;
- la taille de l'image réduit la cadence de prise de vue car le bus qui conduit du capteur à la carte mémoire a un débit limité ;
- les temps d'affichage ou de navigation dans les bases d'images sont accrus d'autant ;
- de nombreux logiciels de visualisation courants ne savent pas traiter les fichiers natifs ;

– le manque de standardisation du format natif laisse planer un doute sur la capacité d’accéder dans le futur aux images stockées de cette façon. Cette menace est d’autant plus sérieuse que la restructuration très rapide des fabricants d’appareils photo fait disparaître des acteurs majeurs qui avaient protégé leurs logiciels de lecture et d’écriture par des codages cryptés. Par ailleurs, l’évolution de la technologie fait aussi évoluer les informations stockées et un même fabricant peut être amené à créer un nouveau format pour satisfaire des besoins que l’ancien n’autorisait pas.

1.2.4. Vers une normalisation du format natif

1.2.4.1. Format natif via TIFF/EP

Le format normalisé par l’ISO (ISO 12234-2 [ISO 10]) est dénommé TIFF/EP (pour *Tagged Image File Format - Electronic Photography*) et dérive du format TIFF 6.0 plus répandu mais propriété d’Adobe. Il se voudrait un sur-ensemble des formats d’image⁸.

A la différence de TIFF, TIFF/EP est librement accessible pour les utilisateurs. Comme les produits TIFF, il vise en particulier des applications nombreuses dans les images graphiques⁹. Pour cela, une attention particulière a été portée à la possibilité de comprimer les documents, ce qui procure des gains beaucoup plus forts pour les images graphiques que photographiques. Trois types de codage sans perte sont disponibles : un codage de Huffman par plages, un codage (de type Huffman) par blocs, le codage Lempel-Ziv depuis qu’il est du domaine public (voir section 1.4). En compression avec perte, seul JPEG/DCT est en pratique couramment disponible.

Dans l’exemple cité plus haut d’un fichier natif issu d’un capteur de Bayer, reconstruit puis transcodé en TIFF, le codage se fait dans TIFF/EP sur deux octets par canal et par pixel, soit six octets par pixel et donc $6n$ octets pour l’image (120 Mo pour une image de 20 Mpixel). C’est le volume le plus important pour représenter cette image. L’application d’une compression sans perte est donc souhaitable pour ramener cette taille à des nombres plus raisonnables : même avec une compression sans perte, on peut alors espérer une compression relativement forte car la dynamique n’utilise généralement pas les 16 bits.

8. Notons que tous les formats finissent par se vouloir surensemble des autres formats. Ils le font au prix d’extensions introduites postérieurement à la définition du format, extensions qui sont rarement prises en charge par les logiciels sensés les utiliser. Ces extensions, dont nous verrons au cours de ce chapitre quelques exemples, rendent la cartographie des logiciels de représentation particulièrement confuse.

9. Les images graphiques se distinguent des images photographiques par de larges plages d’aplats, des couleurs très saturées, des détails (lignes, textes), très fins et contrastés, ainsi que des propriétés particulières : nombres en représentation flottante, attributs de transparence ou d’animation, etc.

Le format TIFF/EP comporte un fichier de métadonnées assez semblable au fichier Exif que nous verrons plus loin, mais issu du fichier de métadonnées du format TIFF dont il ignore un certain nombre de champs. La plupart des champs d'Exif se retrouvent dans TIFF/EP mais insérés dans une structure différente et avec des définitions parfois un peu modifiées.

Le format TIFF/EP est également capable de supporter une représentation d'image par tuiles (voir section 1.7). Il est donc bien adapté à la manipulation de très grandes images.

La normalisation ISO n'est pas suivie par la plupart des fabricants qui s'en tiennent donc à des formats propriétaires. Néanmoins, les formats adoptés sont souvent dérivés de la norme TIFF/EP qu'ils étendent.

La partie traitant du codage sans perte de TIFF/EP est compatible en particulier avec un format proposé par Adobe, mais ouvert lui-aussi et libre d'accès, le format DNG qui dispose de bibliothèques de traitement publiques.

1.2.4.2. *Format natif via DNG*

Adobe a, en effet, également proposé un autre format universel pour transporter une image RAW indépendamment du format propriétaire qui l'a créée [ADO 12]. Ce format DNG (pour *Digital Negative*) est assez largement reconnu par les logiciels de traitement d'image. Il a pour extension *.dng*. Il existe des logiciels libres pour convertir toute image RAW en DNG, mais peu de fabricants d'appareils l'utilisent ; ils préfèrent généralement leur format natif. Par contre, DNG est très fréquemment employé dans les logiciels de traitement et d'exploitation des images.

DNG est une extension du TIFF 6.0 dont il reprend toutes les règles de formatage.

DNG conserve la notion de fichier de métadonnées soit EXIF soit TIFF/EP ou d'autres encore. Il conserve les données propriétaires de ces fichiers et propose des mécanismes pour leur permettre d'être reconnus par les logiciels ultérieurs. Il transmet ainsi les informations propres au système de prise de vue concernant l'illuminant, la calibration, les matrices de conversion de couleur, les formats de représentation, le niveau de bruit, les pixels morts, etc. Il prévoit des champs pour les traitements suivant l'acquisition (représentation en flottants, pixels transparents, etc.). Il supporte des compressions JPEG (avec ou sans perte) et des compressions prédictives sans perte (par DEFLATE, voir section 1.4). Il peut utiliser des images représentées en RVB ou en YCrCb (voir section ??).

1.3. Les métadonnées

Nous venons de voir combien les métadonnées étaient importantes pour exploiter convenablement l'image numérique et nous avons rencontré deux types de fichiers

pour les contenir : celui de TIFF/EP et le fichier Exif. Nous reviendrons un peu plus loin sur le fichier Exif qui est le plus utilisé pour la photographie, mais disons préalablement un mot des efforts de normalisation de ce fichier.

1.3.1. *Le standard XMP*

L'ISO a consacré une norme (16684 – 1 : 2012) à la description de ces données, à partir d'un profil proposé par ADOBE : XMP (*extensible metadata platform*). Ce format permet théoriquement de maintenir conjointement des métadonnées existantes simultanément à un profil XMP, mais en pratique il est fortement recommandé de convertir toutes les métadonnées dans le format XMP. Les métadonnées doivent être regroupées en « paquets » indépendants les uns des autres et potentiellement traités par des applications séparées. Tout utilisateur peut créer ses propres paquets, charge à lui de développer les applications les utilisant, en prenant soin de ne pas affecter les autres paquets.

Le langage RDF (*resource description framework*), popularisé par le web sémantique, est à la base des descriptions de ces paquets et la syntaxe utilisée est le plus souvent la syntaxe XML (*extensible markup language* langage à balises extensibles).

Très utilisé dans le monde de la production artistique, de la photocomposition, de la création d'images de synthèse et de la réalité virtuelle, le format XMP voit de nombreux développements propriétaires au sein des outils professionnels et semi-professionnels de traitement et s'introduit progressivement dans les boîtes à outils de traitement des photographies. L'utilisateur y est cependant peu exposé et dispose rarement des outils pour l'utiliser lui-même, au contraire du fichier Exif qui est d'accès beaucoup plus simple.

1.3.2. *Le format de métadonnées Exif*

Exif (*exchangeable image file format*) est initialement un format complet d'échange d'images et de sons adapté à l'audiovisuel et composé de deux parties : une partie de données comprimées et une partie de métadonnées. Il n'est pratiquement plus utilisé aujourd'hui que pour le fichier des métadonnées qui accompagne une image. Pour la partie compression du signal on lui préfère JPEG, TIFF ou PNG (mais le standard d'échange Exif comporte ces modes de compression parmi les algorithmes qu'il préconise). Pour les métadonnées, on retrouve au contraire les éléments du fichier Exif au cœur de nombreux autres standards : TIFF, JPEG, PNG, etc.

Mis en place par les industriels japonais à la fin des années 1995, il s'est naturellement retrouvé dans tous les matériels et même s'il n'est pas normalisé ni maintenu par une organisation (la dernière version date de 2010), il persiste comme standard de fait. Malheureusement il a tendance à dériver et perdre son universalité.

La partie métadonnées d'Exif (ce que nous exprimerons dorénavant par Exif) s'inspire beaucoup de la structure des fichiers TIFF. C'est un fichier qui s'inscrit à l'intérieur du fichier image (JPEG ou natif), contenant à la fois des données publiques d'utilité très générale et des données propres au fabricant qui n'ont pas vocation à être utilisées hors des logiciels de conversion de l'image. Malheureusement, tous les programmes de manipulation d'images, s'ils reconnaissent le fichier Exif au sein d'une image et s'ils utilisent certaines données, ne le transmettent pas toujours dans les images dérivées, ou le transmettent de façon partielle seulement, entraînant une dilution des informations Exif et une incompatibilité de fait. Le fichier Exif se lit aisément avec des outils simples de reconnaissance de chaînes de caractères (par exemple en langage PERL) et de nombreuses applications pour décoder le fichier sont disponibles en ligne.

1.3.2.1. Les données du fichier Exif

Les données inscrites dans le fichier Exif sont relatives à plusieurs domaines :

- les caractéristiques du matériel :
 - boîtier et objectif : fabricants, modèles, numéros de série de l'appareil,
 - logiciel : version du logiciel de l'appareil, version d'Exif, éventuellement version de FlashPix (logiciel de tuilage),
 - flash si utilisé : fabricant, modèle,
 - dimension du champ réservé aux commentaires de l'utilisateur ;
- les conditions de la prise de vue :
 - date, heure de la prise de vue,
 - orientation de l'appareil (format paysage ou portrait, éventuellement orientation par rapport au Nord magnétique), positionnement GPS si l'on en dispose,
 - position respective des canaux chromatiques et configuration des composants,
 - type de compression et paramètres associés si JPEG,
 - résolution en x , en y et unité employée (mm ou inch), nombre de pixels en x et en y ,
 - distance focale, distance de mise au point,
 - choix de la mesure de lumière, de la mesure de distance,
 - nombre d'ISO de la sensibilité ;
- les réglages internes de l'appareil :
 - espace de couleur : CMYK ou sRGB,
 - paramètres de balance des blancs,
 - temps de pose, ouverture, corrections éventuelles d'exposition,
 - contrôle du gain, de la saturation, du contraste, de la netteté, du zoom numérique,
 - indice d'interopérabilité et version,
 - mode et programme choisi si plusieurs programmes sont disponibles, paramètres,
 - identification d'un traitement personnalisé et paramètres ;

- les données propres au logiciel de traitement :
 - date du traitement,
 - logiciel : marque et version,
 - fonctions employées et paramètres,
 - identification d'un traitement personnalisé et paramètres.

1.3.2.2. Limitations du fichier Exif

Le fichier Exif est un élément précieux pour une utilisation rationnelle et automatisée des images. Il pêche néanmoins par plusieurs défauts :

- Exif ne considère que des composantes chromatiques codées sur 8 bits, ce qui est insuffisant pour les appareils modernes ;
- Exif n'a pas prévu de décrire complètement des fichiers RAW qui exigeraient des champs différents, poussant les fabricants à utiliser des formats de repli : TIFF ou DNG ;
- le fichier Exif est fragile et supporte mal la réécriture ou la modification de données ;
- Exif, n'étant pas maintenu par une instance officielle, évolue de façon anarchique sous la pression des diverses innovations qui en adaptent des morceaux sans souci de compatibilité avec les autres applications ;
- Exif expose son utilisateur à une perte de confidentialité et de vie privée car si l'utilisateur ne prend pas des dispositions explicites, ses photos comporteront des éléments précis sur les conditions de prise de vue : référence unique à l'appareil photo, date de prise de vue et de plus en plus souvent géolocalisation. Des logiciels permettent d'ores et déjà d'exploiter les contenus EXIF au sein de bases de données constituées à partir d'archives web, voire de contenus mails ou de réseaux sociaux, pour déterminer la source de documents.

1.4. Les formats comprimés sans perte

La compression sans perte est une opération réversible qui permet de diminuer le volume occupé par les images sans porter atteinte à leur qualité. Les algorithmes utilisés recherchent les redondances présentes dans l'image et s'appuient sur la théorie de l'information de Shannon pour s'approcher de la limite entropique de volume [GUI 03a, PRO 03].

Les images en noir et blanc ne présentent généralement pas beaucoup de redondance si on les examine pixel par pixel. Leur entropie¹⁰ est souvent supérieure à 6

10. Nous avons vu que l'entropie est une bonne mesure de la quantité d'information portée par l'image, qu'elle s'exprime en premier approximation par l'équation (??) : $\mathcal{E} =$

bit/pixel indiquant un gain possible inférieur à deux bit/pixel qui ne mérite pas de mettre en œuvre un codeur nécessairement un peu complexe. Les images en couleur présentent généralement une plus forte redondance et l'on peut souvent atteindre un rapport 2 entre l'image d'origine et l'image comprimée sans perte.

Mais on peut également rechercher les redondances entre pixels en codant des groupes de pixels (plages ou blocs). Il est possible alors d'exploiter la corrélation spatiale ou chromatique qui sont souvent fortes. Cela se fait au coût d'une plus grande complexité du codage, qui s'accompagne de temps importants, tant au codage qu'au décodage, et à la nécessité de disposer de mémoires importantes pour garder les mots de code. Ces tables doivent aussi être transmises ou conservées et leur coût fait chuter le bénéfice de la compression si les images ne sont pas de très grande taille.

Les algorithmes employés pour le codage sans perte sont tout d'abord ceux dont disposent tous les informaticiens, ce sont ensuite des formats dédiés à l'image ou plus généralement au graphisme.

1.4.1. Les algorithmes généraux de codage sans perte

Ce sont des techniques de codage qui exploitent la redondance de séquences de bits, sans considérer l'origine de ces séquences. Elles sont disponibles de façon banalisée dans les systèmes d'exploitation des ordinateurs [GUI 03a] :

- le codage de Huffman, code à longueur variable, associe des mots de code très courts à des messages très fréquents (le message est ici la valeur d'un pixel ou d'un ensemble de pixels). Le code de Huffman est optimal pour des messages indépendants les uns des autres. Lorsque les messages sont corrélés, Huffman ne tire profit de cette corrélation qu'en associant les messages dans des super-messages. Il n'y a donc pas de modélisation des dépendances et ce n'est pas très efficace pour les images. C'est par contre très efficace pour le codage des graphiques et des dessins et ce codage permet alors des gains importants pour une complexité informatique modique. On n'est pas surpris de voir donc le codage de Huffman par plages (*run-length*) ou par zones dans de nombreuses normes (dont TIFF) ;

- le codage arithmétique ne sépare pas l'image en pixels mais code tout le flot de bits de l'image par une représentation arithmétique unique adaptée aux divers états du signal. Le codage arithmétique n'est pas employé en traitement d'image car il n'y a pas de bon modèle d'apparition des symboles ;

– $\sum_k p(k) \log p(k)$ où $p(k)$ représente la probabilité d'un niveau de gris k et qu'elle est donc limitée par le logarithme du nombre de niveaux de gris de représentation de l'image, donc dans le cas d'une image en noir et blanc, par 8 bit/pixel.

- le codage de Lempel-Ziv (codes LZ77 et LZ78) utilise une fenêtre glissante balayant l'image et un dictionnaire qui garde trace des chaînes élémentaires la première fois qu'il les rencontre. Il remplace les occurrences ultérieures par l'adresse de la chaîne dans le dictionnaire. Le codage LZ est maintenant utilisé en routine pour coder les très gros fichiers à forte redondance comme les fichiers TIFF ;

- des codages hybrides utilisent des références arrière à la façon de Lempel-Ziv, conjuguées avec des représentations plus courtes pour les séquences plus fréquentes, représentations construites à l'aide d'arbres de préfixes. De tels codes sont devenus très populaires sous des noms génériques : Deflate, zip, etc. ;

- des codages plus spécifiques à l'image ont été développés, tirant profit explicitement des dépendances bidimensionnelles et hiérarchiques des images, on les trouvera détaillés dans [PRO 03]. Ils ont conduit à des standards comme JBIG (très adapté à la transmission progressive de documents en télécopie) et à des codeurs comme Ebcot qui sert à la norme JPEG 2000 et sera examiné à la section 1.6.2.

Tous ces codages sont asymptotiquement très proches de la limite entropique de Shannon sur des séquences aléatoires. Ils font payer le poids de leur en-tête pour des messages courts, mais les images ne sont généralement pas des messages courts. S'ils ne sont pas très efficaces sur les images, c'est que la redondance statistique y est en général assez faible dans une représentation par flot de pixels. Elle est cachée dans la structure même de l'image que les codeurs hiérarchiques et les codeurs de la famille JPEG exploitent mieux.

1.4.2. Les codages sans perte de JPEG

Des algorithmes spécifiques pour le codage d'images ont donc été développés qui s'appuient sur les propriétés particulières bidimensionnelles de l'image. Ils ont un certain succès dans des domaines comme l'imagerie médicales où l'on ne tolère pas les dégradations. Ils sont beaucoup utilisés dans le domaine des arts graphiques où l'on manipule conjointement images par balayage, textes et dessins [GUI 03b].

Les approches les plus souvent employées s'appuient sur la prédiction d'un pixel X à la position (i, j) par ses voisins immédiats déjà transmis, donc connus du récepteur, en particulier ceux que l'on désigne par $A(i, j - 1)$, $B(i - 1, j - 1)$, $C(i, j - 1)$ et $D(i, j + 1)$ (figure 1.3, à gauche). La différence entre la valeur prédite et la valeur vraie est souvent très faible et suit une loi de distribution souvent gaussienne que l'on peut coder avec profit avec un codage de Huffman.

Ainsi, la norme JPEG sans perte de 1993 (JPEG *lossless*), propose-t-elle un algorithme de prédiction de X conditionnel à la présence d'un contour horizontal ou vertical. Huit cas de figure sont répertoriés qui conduisent à huit décisions différentes.

Le codage du résidu se fait par Huffman. Des taux de compression pouvant aller jusqu'à trois sont obtenus en imagerie médicale, mais ils sont souvent plus faibles en imagerie naturelle où JPEG sans perte n'a jamais eu de grand succès.

Un second codeur sans perte pour JPEG a ensuite été proposé : JPEG-LS qui s'appuie sur un schéma type de prédiction, LOCO-1, qui dénombre les configurations à partir d'un prédicteur dénommé *median edge detector*. 365 configurations sont recensées et la prédiction est faite sous l'hypothèse de loi de distribution de Laplace de l'erreur. Un codage de Golomb-Rice est alors utilisé pour comprimer la séquence transmise (c'est un codage adapté à des messages pour lesquels les probabilités des faibles valeurs sont très fortes). Il est optimal pour les séquences géométriques. Il a des performances supérieures à JPEG lossless et peut atteindre 3 sur des images naturelles (voir [GUI 03b] pour une comparaison des performances de ces codeurs). Il a fait l'objet d'une normalisation ISO-14495-1.

Avec la norme JPEG 2000, un codage sans perte a ensuite été développé (noté parfois JPEG 2000_R pour réversible), qui utilise un codage par ondelettes bi-orthogonales 5/3 (ondelettes de Le Gall avec un filtre passe-bas à cinq coefficients rationnels et un filtre passe-haut à trois coefficients rationnels). Ce troisième codage, JPEG sans perte, a le mérite d'être scalable¹¹, progressif¹² et compatible avec les facilités offertes par JPEG 2000 pour manipuler les images (voir section 1.6.2). A la différence de JPEG-LS, il utilise le même principe de codage avec ou sans perte mais il est plus lent et de mise en œuvre plus complexe. Il n'a pas de meilleures performances de compression que JPEG-LS.

1.5. Les formats d'image issus des arts graphiques

1.5.1. Le format PNG

Le format PNG (pour *portable networks graphics*) a été conçu comme codage de graphiques en mode balayage (et non vectoriel) et compression sans perte. Il a été développé en concurrence du format GIF (*graphics interchange format*) de CompuServe qui n'était pas d'usage libre pendant de nombreuses années (les brevets de GIF sont maintenant dans le domaine public).

11. La scalabilité est la propriété d'un codage de pouvoir s'adapter au canal qu'il utilise ou au terminal d'affichage en limitant les données transmises lorsque le canal est étroit ou les capacités d'affichage faibles.

12. Un codeur est progressif s'il reconstruit, dans un premier temps, toute l'image de façon grossière, puis affine la qualité de cette image au cours de transmissions successives, permettant à l'utilisateur de « voir apparaître » très tôt l'image dans sa totalité avant que l'ensemble ne soit transmis.

Le format PNG supporte complètement les images en couleur codées sur 24 bits RVB (au contraire de GIF qui ne code les images que sur 8 bits), ainsi que les graphiques codés par des palettes sur 24 ou 32 bits. Il supporte également les informations de transparence associées à une image (le canal α , qui varie de 0 à 1 : 0 = opaque, 1 = totalement transparent). Mais PNG ne supporte pas les graphiques animés (au contraire de GIF). Pour les images, PNG se place résolument après les étapes de correction (démosaïcage, balance des blancs, etc.) et celles d'amélioration (filtrage, réhaussement, etc.). Les informations du fichier EXIF et des traitements ne sont pas transmises et sont donc perdues.

PNG a été normalisé par l'ISO (ISO – 15948 : 2004).

Après un en-tête de signalisation, PNG comporte un certain nombre de fragments (*chunks*), certains obligatoires, d'autres facultatifs, permettant de lire l'image. Les fragments sont identifiés par un code à quatre symboles et protégés par un code correcteur d'erreur. Sont obligatoires :

- le fragment IHDR comportant la largeur de l'image, son nombre de lignes et le nombre de bits par pixel ;
- le fragment PLTE comportant la palette de couleurs lorsque l'image est graphique ;
- le fragment IDAT contenant l'image elle-même, en un ou plusieurs morceaux ;
- le fragment IEND portant la marque de fin.

A l'aide des fragments facultatifs, on peut préciser de nombreux compléments d'usage concernant le fond de l'image, l'espace de couleur utilisé, des corrections sur le blanc, le gamma, le facteur d'homothétie lors de l'affichage et même des informations en cas de visualisation stéréoscopique. PNG a été progressivement étendu jusqu'à permettre des opérations de gestion de couleur (*color management*) en post-production, c'est-à-dire le contrôle de la chaîne de reproduction de l'image jusqu'au produit final.

PNG est donc très utilisé pour les arts graphiques. Il permet de travailler de façon très transparente et économique avec des graphiques ou textes binaires, des graphiques définis jusqu'à 16 bits par canal et quatre canaux (images true color + canal α). Ce n'est pas alors ses capacités de compression qui sont ses atouts premiers. Néanmoins, une image PNG est comprimée par la conjonction d'un codage prédictif (exploitant particulièrement la redondance sur les aplats) et un codeur Deflate. Le codeur prédictif est choisi par l'utilisateur parmi cinq (les valeurs possibles sont : 0 (pas de prédiction), A , B , $(A + B)/2$, A ou B ou C selon la valeur la plus proche de $A + B - C$, (voir le schéma de la figure 1.3).

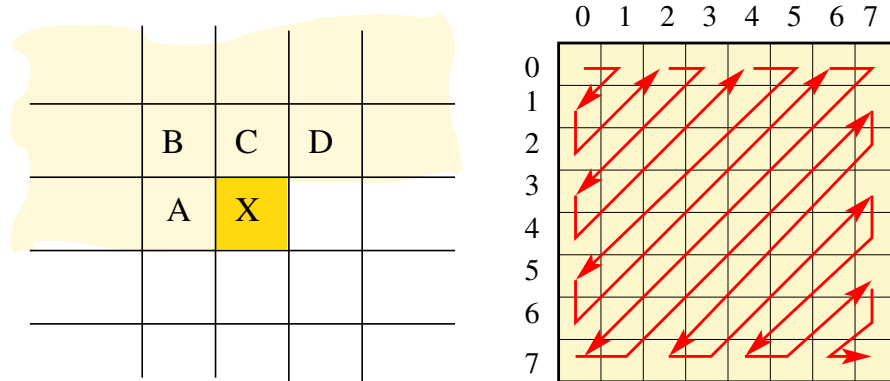


Figure 1.3 – A gauche, lors d'un codage prédictif, le pixel X est codé à partir de ses voisins déjà transmis : A, B, C, D. A droite, ordre de balayage en zig-zag d'une fenêtre de DCT lors d'un codage JPEG

1.5.2. Le format TIFF

Nous avons eu l'occasion de rencontrer plus haut le format TIFF à travers son extension EP permettant d'accueillir un format natif. Revenons maintenant sur ce format. Le format TIFF (*tagged image file format*), maintenant propriété d'Adobe est un format spécifique pour documents scannés dédiés aux arts graphiques qui a progressivement évolué des documents binaires aux documents couleur en haute définition. Conçu comme un conteneur capable de conserver des documents de nature diverse (graphique vectoriel, texte, image par balayage codée avec ou sans perte, etc.), c'est par excellence un format d'archivage et d'échange qui permet d'assurer la qualité des documents source utilisés conjointement dans la construction d'un document final.

Il est constitué de blocs de codes, identifiés par des étiquettes (*tags* qui ont donné leur nom au format) chaque bloc spécifiant le contenu des données. Les étiquettes sont codées sur 16 bits. A partir de 32 000, elles sont attribuées à des entreprises ou organisations pour des besoins collectifs. au-delà de 65 000 et jusqu'à 65 535, les étiquettes peuvent être spécifiées par l'utilisateur à titre privé pour ses applications. TIFF a donc donné naissance à de nombreuses variantes parfois peu compatibles. En 2015, la version de base 6.0, établie il y a plus de vingt ans, indique les points essentiels permettant d'assurer une compatibilité entre systèmes. TIFF permet de contenir des structures très variées dans une seule image : des sous-fichiers, des bandes ou des tuiles qui, chacune, dispose de caractéristiques spécifiques qui expliquent leur coexistence.

Les algorithmes relisant le format TIFF doivent pouvoir décoder le codage sans perte de Huffman par plages ainsi que le codage JPEG par DCT, mais éventuellement

d'autres schémas de compression (comme LZ et JPEG ondelettes) car des drapeaux et des étiquettes spécifiques sont consacrées à ces compressions.

Les images couleur reconnues par TIFF sont de type RVB, YCbCr ou CIE la^*b^* , codées sur 24 bits ou de type CMYK codées sur 32 bits (voir le chapitre ??). Des extensions ont été faites pour des images à plus de 8 bits par canal, ainsi que pour les très grosses images.

En règle général, le fichier TIFF associé à une image est assez volumineux. Dans le cas d'une image de n pixels en sRVB (trois octets par pixel), le fichier TIFF qui utilise exactement un octet par canal, aura pour taille $3n$ octets sans compression, probablement la moitié environ après compression sans perte, mais c'est une décision de l'utilisateur ou non d'y faire appel et cela ralentit son emploi.

1.5.3. Le format GIF

Indiquons ici quelques caractéristiques du format GIF sans perte. C'est également un format propriétaire, maintenu par CompuServe, principalement destiné au graphique, mais permettant cependant de transporter des images sous certaines conditions.

GIF est un codage sur 8 bits de graphiques codés à travers une table de couleurs (ou palette) utilisant le codage de Lempel-Ziv suivant un codage par plage. Une image en noir et blanc est donc transmise sans beaucoup d'altération par une table transparente (quoique quelques niveaux de gris soient souvent réservés pour une représentation graphique minimale), mais sans grande compression. Une image en couleur devra être sévèrement quantifiée (il n'y a plus alors copie sans perte), ou devra être partagée en morceaux, chacune disposant de sa propre table de couleur limitée à 256. Les performances de compression sont alors plus faibles. Une autre solution utilise la technique du tramage (*dithering*) qui remplace un petit bloc uniforme de pixels par un mélange de pixels issus de la palette.

1.6. Les formats de compression d'images avec pertes

Nous avons eu l'occasion de dire que le gain en volume lors d'une compression sans perte d'images est assez limité. Elle permet typiquement de gagner au mieux un facteur 2 avec une image en noir et blanc, un facteur 3 avec une image couleur et un facteur de 5 à 10 pour une image codée sur 3×16 bits.

On s'est cependant très vite rendu compte qu'il était possible de modifier légèrement l'image par un codage qui devenait irréversible, sans altérer notablement son apparence. Pour cela, on joue sur la tolérance du système visuel à des infidélités de

reproduction chaque fois que le signal est assez complexe (on exploite alors l'effet de masquage), c'est-à-dire dans les zones texturées et près des contours.

De cette remarque sont nés les premiers codages avec perte, utilisant les schémas que nous avons déjà signalés : codages prédictifs ou codages par longueurs de plage ou de zones, mais pour lesquels on s'autorise une tolérance pour approcher le signal par des chaînes plus longues. Malgré la sophistication de ces approches, elles n'ont pas permis d'obtenir des taux suffisants et des méthodes nouvelles sont apparues où l'on tire profit de la forte redondance spatiale des images sur des blocs relativement larges en projetant les images sur des bases de fonctions adaptées pour regrouper l'énergie en un petit nombre de coefficients. Ces bases sont d'une part la transformation de Fourier (et surtout sa variante la DCT) et la transformation en ondelettes [BAR 03b].

Dans tout codage avec perte, l'utilisateur dispose d'un paramètre α lui permettant d'imposer le compromis qualité/compression qu'il souhaite atteindre. Ce compromis est très approximatif et dépend beaucoup de l'image qu'il veut comprimer. Il existe des critères objectifs de l'effet d'un codage que nous avons examinés en détail au chapitre ?? . Si l'on désigne par $I(i, j)$ l'image avant codage et par $I'(i, j)$ l'image après codage, les plus courants sont :

– le taux de compression :

$$\tau = \frac{\text{nombre de bits de } I}{\text{nombre de bits de } I'}$$

– l'erreur quadratique moyenne de codage :

$$e = 1/N \sum_{i,j \in I} (I(i, j) - I'(i, j))^2$$

– le rapport signal à bruit :

$$SNR = 10 \log_{10} \left[\sum_{i,j \in I} (I(i, j) - I'(i, j))^2 / I(i, j)^2 \right]$$

– et le rapport signal bruit au pic :

$$PSNR = 10 \log_{10} \left[\sum_{i,j \in I} (I(i, j) - I'(i, j))^2 / 255^2 \right]$$

1.6.1. JPEG

C'est à partir des années 1980 qu'un groupe de chercheurs s'est penché sur l'élaboration d'un standard efficace de compression avec perte. Ses travaux ont abouti dans

les années 1990. Le format JPEG (pour *joint photographic expert group*) a été normalisé en 1992 sous le nom ISO/CEI 10918-1. C'est une norme qui laisse de nombreux espaces de liberté pour les développeurs de logiciel et il existe donc des variantes du logiciel de codage. Seule contrainte, tous les logiciels de décodage doivent pouvoir relire un ensemble d'images de test déposées [GUI 03b].

1.6.1.1. Les principes du codage JPEG

Le codage JPEG est constitué de six étapes :

- un changement d'espace de couleur pour ramener le signal, généralement présenté dans l'espace RVB, dans un espace luminance/chrominance (généralement YCrCb),
- un sous-échantillonnage des composantes de chrominance par un facteur variable. Il peut atteindre 4 en x et en y , mais est le plus souvent de 2 (cette étape tire profit de la plus faible discrimination de l'œil aux fluctuations de couleur qu'aux fluctuations de gris) ;
- un découpage de l'image en blocs de taille fixe : 8×8 pixels pour les images de luminance, 16×16 pixels pour les images de chrominance ;
- la transformation de ces blocs par un opérateur qui « concentre » l'énergie du bloc dans un petit nombre de coefficients, en l'occurrence la transformation en cosinus discrets DCT (*discrete cosine transform*) ;
- la quantification de ces coefficients de façon à réduire leur entropie ;
- la transmission de ces coefficients réordonnés par un balayage en zig-zag de la fenêtre, puis un codage de type Lempel-Ziv de la chaîne de bits.

Le décodage inversera chaque étape hormis celles qui ont entraîné des pertes irréversibles d'information : les deuxième et cinquième étapes.

La transformée en cosinus discret, définie par la matrice du tableau 1.1, est une variante de la transformée de Fourier, mieux adaptée aux signaux réels et positifs comme les images [AHM 74]. C'est une transformation bidimensionnelle, exacte et inversible d'un signal scalaire qui crée un signal réel ayant autant d'échantillons que le signal d'entrée.

Soit $J(i, j)$ une fenêtre de $N \times N$ pixels (par exemple $N = 8$) du canal à transmettre (canal de luminance ou de chrominance de l'image d'origine). La DCT K_J de J est donnée par :

$$K_J(k, l) = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)J(i, j) \cos \left[\frac{(2i+1)k\pi}{2N} \right] \cos \left[\frac{(2j+1)l\pi}{2N} \right] \quad (1.1)$$

avec $C(0) = \sqrt{2}/2$ et $C(i, i \neq 0) = 1$.

	0	1	2	3	4	5	6	7
0	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
1	0,981	0,831	0,556	0,195	-0,195	-0,556	-0,831	-0,981
2	0,924	0,383	-0,383	-0,924	-0,924	-0,383	0,383	0,924
3	0,831	-0,195	-0,981	-0,556	0,556	0,981	0,195	-0,831
4	0,707	-0,707	-0,707	0,707	0,707	-0,707	-0,707	0,707
5	0,556	-0,981	0,195	0,831	-0,831	-0,195	0,981	-0,556
6	0,383	-0,924	0,924	-0,383	-0,383	0,924	-0,924	0,383
7	0,195	-0,556	0,831	-0,981	0,981	-0,831	0,556	-0,195

Tableau 1.1 – Coefficients des filtres DCT pour $N = 8$, i en ligne et k en colonne (équation (1.1)).

La transformation inverse :

$$J(i, j) = \frac{2}{N} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} C(k)C(l)K_J(k, l) \cos \left[\frac{(2i+1)k\pi}{2N} \right] \cos \left[\frac{(2j+1)l\pi}{2N} \right] \quad (1.2)$$

permet de retrouver exactement $J(i, j)$ tant que l'on n'a pas procédé à la quantification de $K_J(k, l)$.

Les variables k et l sont des fréquences spatiales comme pour une transformée de Fourier. Le couple $(k = 0, l = 0)$ porte la fréquence nulle (c'est-à-dire, au facteur N près, la valeur moyenne de $J(i, j)$ sur la fenêtre). Lorsque k croît, les fréquences horizontales croissent. Lorsque l croît, ce sont les fréquences verticales et toute valeur de couple (k, l) où $k \neq 0$, $l \neq 0$ correspond à une fréquence inclinée par rapport aux axes $0x$ et $0y$.

Le terme $K_J(0, 0)$, appelé composante continue, est toujours beaucoup plus grand que les autres. Il est traité à part dans la DCT, souvent transmis sans altération de façon à donner une information exacte, mais imprécise, de la valeur du bloc.

Les autres termes feront l'objet au contraire d'une quantification d'autant plus importante que l'on souhaitera des taux de compression élevés. On définit un quantificateur $q(k, l)$ et on procède à l'approximation du coefficient $K_J(k, l)$ par $K'_J(k, l)$ tel que :

$$K'_J(k, l) = \left\lfloor \frac{K_J(k, l) + \lfloor q(k, l)/2 \rfloor}{q(k, l)} \right\rfloor \quad (1.3)$$

où $\lfloor x \rfloor$ désigne l'arrondi entier par défaut de x .

L'ensemble des coefficients $q(k, l)$ forme ainsi une matrice de quantification \mathcal{Q} qui est appliquée à toute l'image. Elle est transmise avec les coefficients pour permettre



Figure 1.4 – Deux imagettes issues de la photo du centre (coin supérieur gauche) dont les transformées DCT, après décimation, sont données dans les tableaux 1.1(a) et 1.1(b) (les niveaux de gris des imagettes ont été amplifiés pour faciliter la lecture)

(a) Imagette 1 de la figure 1.4								(b) Imagette 2 de la figure 1.4									
	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7
0	1450	65	0	0	0	0	0	0	0	980	100	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	60	200	0	0	0	0
2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
5	131	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0

Tableau 1.2 – Coefficients DCT de la sous-imagette 1 et 2 de la figure 1.4, après mise à zéro de tous les coefficients inférieurs à 30 en valeur absolue. On note l'importance du terme $J(0,0)$ correspondant à la valeur moyenne du bloc, on note aussi l'ampleur des fréquences spatiales verticales dans 1 et horizontales dans 2 correspondant aux périodicités apparentes dans ces images.

de reconstruire l'image. Les divers codeurs se distinguent par la stratégie d'emploi du sous-échantillonnage de chrominance d'une part et le choix des matrices de quantification d'autre part.

On peut montrer que les termes de haute fréquence contiennent généralement très peu d'énergie et que ces termes sont moins importants pour la qualité perçue de l'image. On fait en sorte qu'ils soient ramenés à zéro par la quantification (voir figure 1.4 et tableau 1.1).

On procède alors à la transmission des coefficients $K'_j(k, l)$ par un balayage en zig-zag qui place en premier, juste après la composante continue les fréquences dans un ordre bidimensionnel (autant qu'on le peut) croissant (voir figure 1.3, à droite). Le train de coefficients ne comporte rapidement plus que des zéros ce qu'un codage de Liv-Zempel ou de Huffman par plages comprimeront très bien.



Figure 1.5 – Agrandissement d'une zone d'image codée par JPEG. L'image en haut à gauche est l'originale non codée. Les facteurs de compression (pour l'image entière) valent ensuite 5,4 (en haut à droite), 12,1 et 20 (ligne du centre), 30 et 40 (ligne du bas)

Lors de la décompression des images JPEG, on procède à l'inversion de toutes les étapes précédentes. La quantification des coefficients peut faire apparaître des défauts de bloc, caractéristiques de JPEG. Certains algorithmes de reconstruction comportent donc une étape de filtrage passe-bas qui gomme ces défauts. Le sous-échantillonnage

des composantes chromatiques peut aussi nécessiter une étape d'interpolation qui pourra à son tour dégrader l'image.

1.6.1.2. *JPEG en pratique*

La compression par JPEG est aujourd'hui la façon ordinaire de transmettre des images de qualité acceptable sous un volume lui aussi acceptable. Pour des images en couleur, on a l'habitude de considérer qu'un taux de compression de 10 s'accompagne d'une très faible dégradation, pratiquement imperceptible à l'œil, d'autant plus que les images sont très grandes et observées sous un faible grandissement.

Des taux de compression de 100 sont couramment utilisés, en particulier sur le web ou pour des échanges par mail. La qualité des images en est très fortement affectée et il n'est guère possible alors d'en faire un autre usage qu'une visualisation rapide sans être gêné par les artéfacts.

Notons que tous les codeurs disposent d'un paramètre souvent unique qui contrôle le taux de compression. Cet indicateur n'est cependant formellement lié ni au gain lors de la compression, ni à la qualité de l'image. Un même indice pour des images différentes pourra donner des résultats très différents. Cependant l'indice, lorsqu'il décroît et pour une même image s'accompagne toujours de dégradations plus fortes et de taux de compression supérieurs. Cet indice vaut généralement 100 pour un codage sans perte et décroît jusqu'à la valeur minimale de 1.

Rappelons que JPEG sans perte n'utilise cependant pas du tout le principe général de la DCT pour coder l'image puisqu'il fait un codage prédictif.

Il est possible d'utiliser JPEG de façon progressive lors d'une transmission, les termes de très basse fréquence sont transmis les premiers et l'image est raffinée ensuite lors de la transmission des fréquences plus élevées.

L'image constituée des seules composantes continues, donc 64 fois plus petite que l'image originale, est souvent utilisée pour des fonctions d'affichage ou de recherche rapide dans les bases de données.

1.6.2. *JPEG 2000*

JPEG 2000 est une norme reconnue par l'ISO sous le nom : ISO/CEI 15444-1. C'est une norme assez générale d'échange d'images avec des fonctionnalités variées (archivage, internet, vidéo). Nous nous attacherons surtout aux aspects liés au format d'image.

JPEG 2000 exploite la propriété des décompositions en ondelettes de mieux compresser les images que la DCT, c'est-à-dire de fournir, pour un même taux de compression, une image de meilleure qualité, ou, pour une qualité fixée un taux de compression plus fort [BAR 03a].

Elle permet aussi d'assurer des fonctions de scalabilité.

JPEG 2000 utilise également un codeur arithmétique très évolué EBCOT (*embedded block coding with optimized truncation*) qui tire profit de la structure en arbre bidimensionnel de la décomposition en ondelettes [TAU 00].

Enfin, JPEG 2000 offre une grande possibilité de décrire à un niveau supérieur de la pyramide le contenu des images pour faciliter, d'une part l'utilisation intelligente des collections d'images, d'autre part leur transmission optimisée dans les réseaux de nouvelle génération. Elle exploite pour cela les possibilités de transmission progressive données par la scalabilité des ondelettes à travers une structure de transmission appelée *codestream* ou la possibilité de définir des régions d'intérêt (ROI *region of interest*) qui seront transmises en priorité et avec une meilleure qualité. Elle prévoit explicitement la présence de textes ou de graphiques dans une structure appelée *compound*. Elle est faite enfin pour pouvoir s'appliquer à des images très grandes (jusqu'à 2^{32} pixels par ligne !) ou dotées de très nombreux canaux (jusqu'à 2^{14}) et avec des pixels à très forte dynamique (jusqu'à 38 bits par échantillon).

Comme beaucoup d'autres normes, JPEG 2000 est ouverte et ne précise pas explicitement le mode de compression, laissant au développeur la possibilité d'apporter des améliorations au cours du temps. On impose au codeur de produire un train binaire conforme à la norme, mais on ne lui impose pas la stratégie pour le faire. Au décodeur, n'est imposée que la capacité à décoder des fichiers types qui sont joints à la norme.

JPEG 2000 est une norme de transmission très adaptée aux besoins professionnels. Elle a plus de mal à trouver sa place dans le grand public, non pas tant en raison de ses besoins en puissance de calcul supérieurs à ceux de JPEG et des logiciels de décodage assez lourds, qu'en raison de l'inertie d'un marché grand public qui ne recherche pas en priorité un rapport qualité/débit élevé.

1.6.2.1. Principe de JPEG 2000

Comme JPEG, JPEG 2000 peut transformer les images en couleur de l'espace RVB dans un espace luminance/chrominance (YUV ou $YC_R C_B$). Pour cela, deux transformations sont disponibles, l'une réversible, l'autre irréversible (mais plus efficace en termes de compacité de représentation). Les composantes chromatiques sont ensuite traitées séparément.

JPEG 2000 travaille ensuite, soit sur l'image complète, soit sur des tuiles (*tiles*) qui sont des fragments d'image tous de même taille (à l'exception de ceux des bords).

Ces tuiles sont décomposées de façon hiérarchique par un banc de filtres en ondelettes bidimensionnelles et bi-orthogonales [BAR 03a].

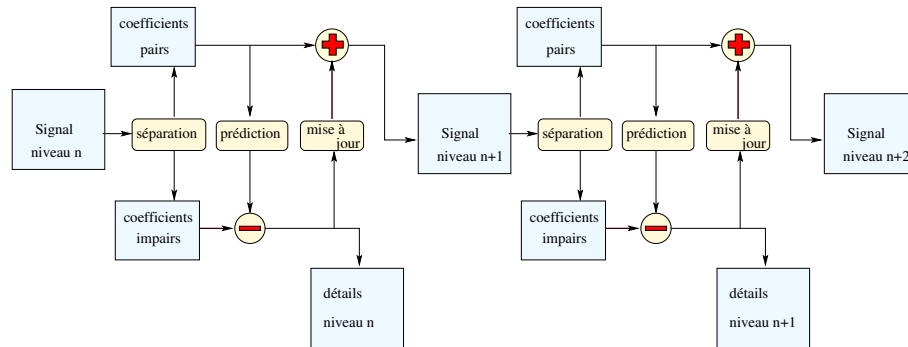


Figure 1.6 – Schéma de lifting entre trois niveaux de n , $n + 1$ et $n + 2$ décompositions en ondelettes : on sépare les échantillons pairs et impairs. On définit les détails du niveau n comme la différence entre les échantillons impairs et leur prédiction faite à partir des échantillons pairs. On prend en compte le risque de repliement de spectre en filtrant le signal de détail et en le réinjectant dans les échantillons pairs qui forment alors le signal du niveau $n + 1$ auquel on peut de nouveau appliquer le schéma de lifting, et ainsi de suite jusqu'à la profondeur souhaitée.

Deux types de filtres sont autorisés par la norme¹³ :

- les filtres à coefficients rationnels (5,3) de Le Gall que nous avons évoqués à la section 1.4 ; ils ne travaillent que sur des entiers sans arrondis et permettent un codage avec ou sans perte, selon que l'on transmet tous les coefficients non nuls ou pas (selon le schéma EBCOT) ;

- les filtres à coefficients réels (9,7) de Daubechies¹⁴ (voir tableau 1.3), qui ne permettent qu'un codage avec perte, car les calculs se font en nombres réels avec une précision finie. Ces ondelettes donnent de meilleures performances que les ondelettes rationnelles pour le codage avec perte.

Ces filtres sont des filtres bi-orthogonaux¹⁵ qui sont définis par un filtre passe-bas servant à calculer des images à résolution de plus en plus faible depuis le niveau original de la tuile et des filtres passe-haut qui permettent de définir à chaque niveau

13. Ces filtres n'ont malheureusement pas de forme analytique simple. Ils s'expriment soit par la liste de leurs coefficients soit par leur équation dans l'espace polyphase où ils sont définis.

14. Ou plus exactement, ondelettes de Cohen-Daubechies-Feauveau [COH 92].

15. Les filtres sont orthogonaux si le produit scalaire de deux filtres de rang différent est nul. Si l'on ne peut tenir cette contrainte très difficile à réaliser, on peut imposer la bi-orthogonalité, c'est-à-dire l'orthogonalité entre les filtres passe-bas et passe-haut.

le contenu des détails, caractéristiques du niveau considéré. Ces détails sont portés par des coefficients de la décomposition en ondelettes : $a_n(i, j)$.

Le filtrage peut se faire de deux façons : soit comme d'habitude en traitement du signal, par convolution, soit, et c'est l'un des mérites de l'approche hiérarchique par ondelettes, par application itérative de deux filtres l'un passe-bas, l'autre passe-haut, ces filtres étant identiques à tous les étages de la pyramide. C'est ce que l'on appelle le schéma de *lifting*. Dans ce schéma (voir figure 1.6), on exprime les coefficients impairs d'une décomposition à partir de la valeur des coefficients pairs affectée d'un terme correctif. Ce terme correctif constitue l'image de détails propre au niveau considéré [SWE 96].

La transformée en ondelettes de base est monodimensionnelle. On l'applique donc une fois en lignes et une fois en colonnes sur le domaine d'intérêt. Il en résulte quatre images (voir la figure 1.7) :

- une image de hautes fréquences en lignes et en colonnes ;
- une image de hautes fréquences en lignes et de basses fréquences en colonnes ;
- une image de hautes fréquences en colonnes et de basses fréquences en lignes ;
- une image de basses fréquences en lignes et en colonnes.

On montre que, pour des bases d'ondelettes bien choisies, ces coefficients présentent des propriétés statistiques très intéressantes : ils sont très concentrés et peuvent être très bien modélisés à l'aide d'une loi gaussienne généralisée qui permet donc de les quantifier de façon très efficace :

$$p(a_n) = \alpha \exp(-|bx|^d) \quad (1.4)$$

Les tables de quantifications sont définies pour chaque niveau de résolution. Elles sont calculées pour fournir une grande diversité de compression.

Ce sont ces coefficients quantifiés, transformés en plans de bits, regroupés par sous-bande qui sont présentés au codeur. Comme nous l'avons évoqué plus haut, le codeur adopté, EBCOT, utilise un schéma très performant de codage contextuel dans l'arbre des coefficients d'ondelettes prédisant un bit à partir du bit du niveau de résolution supérieure et de ses voisins immédiats, en commençant par les bits de poids fort. Le codeur travaille sur des sous-blocs dont la taille est laissée libre mais assez réduite (maximum : 64×64).

La puissance du codage repose sur l'importance de l'analyse hiérarchique fournie par l'analyse en ondelettes. On s'intéresse à des voisinages en 6- ou 27-connexité dans la pyramide tridimensionnelle des niveaux de résolution. Les pixels majoritaires (identifiés par le bit du niveau supérieur) ne sont pas codés, les autres sont codés par leur signe et leur écart au niveau supérieur s'il est significatif. Enfin, on examine les

Ordre du filtre	0	± 1	± 2	± 3	± 4
analyse : filtre passe-bas	0,602949	0,266864	-0,078223	-0,016864	0,026749
analyse : filtre passe-haut	0,557543	0,295636	-0,028772	-0,045636	0
synthèse : filtre passe-bas	0,602949	-0,266864	-0,078223	0,016864	0,026749
synthèse : filtre passe-haut	0,557543	-0,295636	-0,028772	0,045636	0

Tableau 1.3 – Coefficients des filtres des ondelettes de Daubechies 9/7 pour l'analyse et la reconstruction de l'image (d'après [BAR 03a]). Notez que l'on trouve parfois des formules différant d'un facteur $\sqrt{2}$. Les filtres d'analyse et de reconstruction doivent être symétriques par rapport à l'ordre 0 de façon à ne pas déplacer les contours des images.

situations où il n'y a pas de configuration dominante. L'arbre ainsi créé est codé par un code algébrique.

1.6.2.2. Les performances de JPEG 2000

JPEG 2000 permet des transmissions d'images fixes avec des taux de compression supérieurs à JPEG de l'ordre de 20 % à qualité fixée. Les défauts de JPEG 2000 sont subjectivement moins gênants que ceux de JPEG car il n'y a pas d'effet de bloc systématique. Les grandes images sont particulièrement bien codées par JPEG, ainsi que les images contenant peu de très hautes fréquences. JPEG 2000 donne des résultats de qualité satisfaisante à très fort taux de compression (de l'ordre de 100). Il apparaît également comme une norme bien adaptée pour les images de télédétection pour sa capacité à traiter de grandes images dans un mode compatible avec le tuilage nécessaire à une navigation à haute résolution et avec la possibilité de spécifier des zones d'intérêt qui recevront une attention plus élevée.

1.7. Les formats tuilés

Disons un mot, avant de quitter le domaine du codage des images fixes, du codage des très larges images. Ce n'est pas encore vraiment le cas des images photographiques, mais avec des capteurs qui dépassent maintenant 50 mégapixels, il se peut que nous retrouvions ces formats au sein de certains appareils d'ici quelques années.

Les formats tuilés concernent donc les très grands fichiers d'images (plus de 10 mégapixels et jusqu'à plusieurs gigapixels) et proposent des méthodes rapides pour les consulter. Ils utilisent une décomposition de l'image en blocs (par défaut 64×64 pixels dans le format FlashPix, le plus répandu, et 256×256 dans le format IVUE), chaque bloc étant appelé tuile. Ils utilisent également une description hiérarchique pyramidale, binaire de l'image qui conduit de l'image pleine résolution (constituée

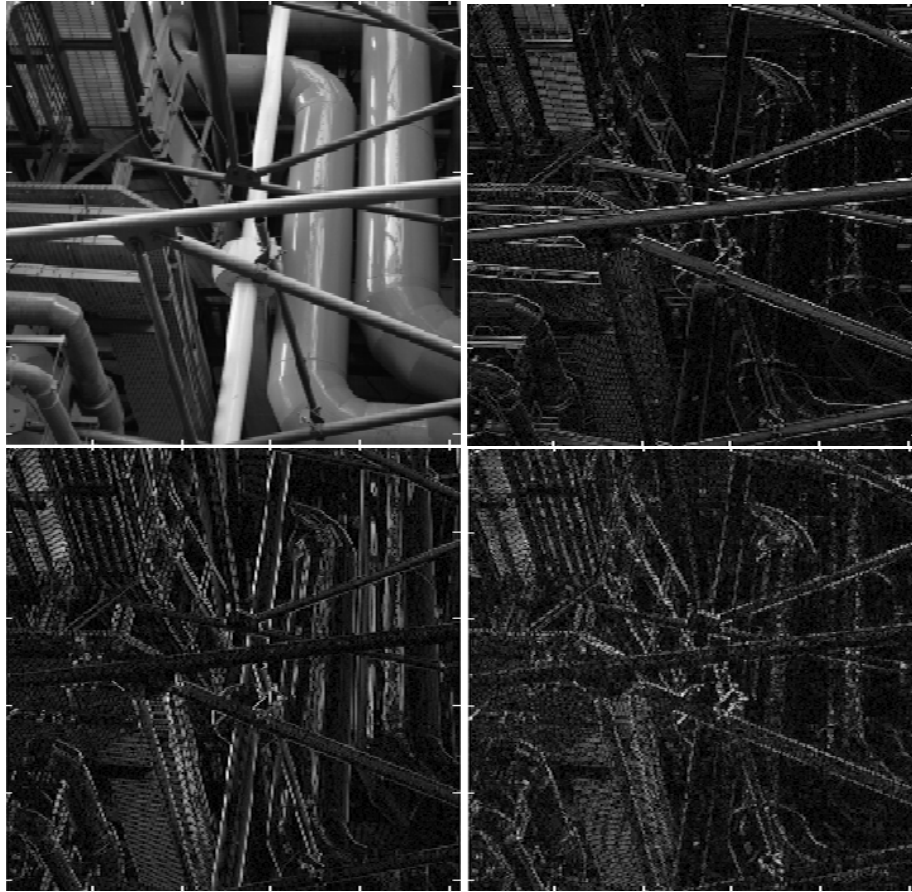


Figure 1.7 – Codage d'une image par JPEG 2000. En haut à gauche, image de niveau 2, entourée des trois images de détails du niveau 1 : en haut à droite, détails résultant d'un filtre passe-bas horizontal et d'un filtre passe-haut vertical, en bas à gauche, détails résultant d'un filtre transposé : passe-haut horizontal et passe bas vertical, en bas à droite, résultat d'un filtre passe-haut vertical et passe-haut horizontal. Ces quatre images, proprement combinées, redonneront l'image d'origine. (La base d'ondelettes utilisée est la base de Daubechies).

de plusieurs centaines de tuiles) à une image unique, très grossière, contenue sur une seule tuile. Le pixel d'un niveau de la pyramide est obtenu en moyennant les quatre pixels qu'il représente au niveau inférieur. Une structure d'arbre permet de déterminer rapidement le niveau de représentation nécessaire pour une demande de l'opérateur (par exemple pour afficher une zone donnée sur un écran de taille donnée). L'usage de cet arbre et l'exploitation de structures de représentation précalculées expliquent

le gain très important obtenu lors de la manipulation de l'image, aussi bien en entrées/sorties qu'en traitement, par exemple lors d'un défilement sur l'écran.

Ce gain se paie par un surcoût de stockage puisque l'on est amené à conserver l'image originale (partagée en petites tuiles) ainsi que toutes les images de résolution 1/2, 1/4, 1/8, ce qui, aux erreurs de troncatures près, se traduit par un stockage total de $\sum_0^\infty 1/4^n$ fois la taille de l'image, soit un surcoût de 33%.

Le format FlashPix autorise des tuiles de tout type : huit ou seize bits, un, trois ou quatre canaux, codées avec ou sans perte. La description de la représentation par FlashPix (taille et nombre de tuiles) a une place au sein du fichier EXIF.

Pour les très grands ensembles de données utilisés en astronomie et en télédétection, dont le référentiel est la sphère et non le plan, des modèles particuliers de représentations tuilées de la sphère sont employés. Le plus connu, HEALPix (*hierarchical equal area isolatitude projection of a sphere*) [HEA 14], décompose la sphère par projection sur douze carrés du plan répartis en quinconce et utilisant des projections différentes selon la position des zones concernées. Ces carrés sont orientés à $\pi/2$ sur l'équateur et centrés soit en $(0, k\pi/2)$ soit en $(k\pi/4, \pm\pi/4)$. Ces carrés sont ensuite subdivisés en 4, dans une approche hiérarchique classique.

Une projection tuilée [KUN 01], plus intuitive, s'appuie sur une décomposition en triangles découpés le long de grands cercles de la sphère : la projection HTM (*hierarchical triangular mesh*) ; une autre exploite simplement la structure hiérarchique quadratique. C'est le format Q3C (*QuadTree Cube*) [KOP 06].

Ces représentations tuilées sphériques prennent tout leur intérêt dans les assemblages de photographies et les panoramas à très large champ (voir section ??).

1.8. Le codage vidéo

Le codage vidéo déborde très largement du cadre de cet ouvrage, néanmoins de plus en plus nombreux sont les appareils de photo qui permettent d'enregistrer soit des images fixes, soit des séquences animées. Ils comportent alors obligatoirement un codeur vidéo qui comprime le flot d'images. Ces codeurs exploitent, comme les codeurs d'images fixes, la redondance au sein de chaque image (c'est ce que l'on appelle le codage intra-image), mais peuvent également réduire la redondance entre images successives (c'est le codage inter images).

1.8.1. Vidéo et normalisation

Les travaux de normalisation ont débuté au sein de l'UIT¹⁶, avec des recommandations appelées H.120, puis H.261, H.263 et H.264. La norme H.263 la première fait intervenir un mode inter images ; H.264 sera la base de MPEG-4.

Ces recommandations ont été reprises dans un cadre moins dédié aux télécommunications pour des applications d'archivage, de jeux vidéo, d'usage privatif. C'est le groupe de travail MPEG (*moving picture expert group*), groupe commun à l'ISO et à la CEI, qui s'est chargé de construire des solutions de consensus aptes à passer en normalisation [NIC 03].

La première recommandation MPEG-1 visait des transmissions à bas débit pour les transmissions sur fil avant l'ADSL (par paquets de 64 kbit/s, permettant de monter jusqu'à 1,5 Mbit/s environ). Ce très faible débit permettait la transmission vidéo d'images de taille réduite (352 × 288 pixels par image, à 25 image/s). MPEG-1 a beaucoup été utilisée dans le cadre de l'archivage sur vidéo-CD.

MPEG-2 est une norme adaptée à la télévision. Elle vise des transmissions de très bonne qualité (codage dit de contribution) et convient particulièrement à la transmission hertziennne, sur câble ou sur ADSL. Elle autorise des transmissions à des débits de 2 à 6 Mbit/s pour l'image vidéo ordinaire et des débits de 15 à 20 Mbit/s pour des vidéo en HDTV¹⁷.

MPEG-4 est une norme de transmission multimédia dont une partie traite spécifiquement l'image vidéo. Son domaine de transmission privilégié concerne la transmission à bas débit (autour de 2 Mbit/s). Elle utilise des mécanismes de compression plus élaborés que ceux de MPEG-2 et pouvant s'adapter à de très fortes compressions (avec bien-sûr alors une perte de qualité importante). Elle tire profit des propriétés de scalabilité présentes dans la recommandation H.263. Elle propose de nombreuses passerelles avec le monde de l'informatique de façon, en particulier, à assurer la gestion des droits de propriété.

MPEG-7, MPEG-21 et MPEG-x (x prenant les valeurs de A à V) sont des développements qui visent les applications multimédias des séquences vidéo et définissent des fonctionnalités spécifiques à l'archivage, la recherche dans les bases de données, la production (en particulier par intégration de sources variées), l'interaction, la diffusion et la protection des contenus.

16. UIT = Union internationale des télécommunications) ici le groupe intéressé est l'UIT-T, chargé de la normalisation.

17. HDTV, ou TVHD = télévision haute définition = 1 920 × 1 080 pixels par image à 50 images par seconde.

1.8.2. Codage MPEG

1.8.2.1. Les mécanismes mis en œuvre dans MPEG

Les normes de codage vidéo de MPEG-2 et de MPEG-4 sont aujourd'hui présentes dans de nombreux appareils photographiques. Elles tirent profit des architectures dédiées au traitement vectoriel, présentes au sein de l'appareil, pour accomplir des calculs très lourds qui exploitent beaucoup d'éléments de ces normes [NIC 03, PER 02].

Examinons tout d'abord l'implémentation de MPEG-2. L'image en couleur, généralement acquise dans un espace RVB, est convertie dans un espace {luminance, chrominance} et les canaux de chrominance sont généralement sous-échantillonnés d'un facteur 2 en lignes et en colonnes, comme cela a été fait avec JPEG.

Le codage intra-image se fait par subdivision de l'image en groupes de blocs (GOB), construits à partir de macroblocs MB. Un macrobloc est lui-même constitué de quatre blocs de 8×8 pixels de luminance et généralement un bloc de chrominance (codage à chrominance sous-échantillonnée dit 4 : 2 : 0) ou quatre blocs (cas du codage 4 : 4 : 4 sans sous-échantillonnage). Le train binaire sera donc constitué d'une couche image (I), constituée des couches de groupes de blocs (GOB), elles-mêmes constituées de la couche de macroblocs (MOB). Cette couche de macroblocs comportera typiquement les quatre blocs de luminance suivis de deux blocs de chrominance, l'un pour la dominante R, l'autre pour la dominante B.

Les blocs sont codés par transformation en cosinus discret (DCT), quantifiée et suivie d'un codage entropique (codeur par longueur de plage, ou mieux, codage arithmétique) (figure 1.8).

Le codage inter images se fait selon deux modes :

- le mode prédictif (mode P) : une image du flot est prise comme référence et transmise avec un codage intra-image (image I). La nouvelle image (P) à transmettre à un temps postérieur est comparée à I de façon à déterminer le déplacement que chaque pixel a subi. C'est une étape de prédiction de mouvement qui fonctionne par corrélation de blocs (*block matching*) et détermine un vecteur de déplacement (Δx en x et Δy en y) pour chaque macrobloc. La différence entre le macrobloc de l'image P et celui de l'image I, déplacé de $\{\Delta x, \Delta y\}$, constitue un macrobloc d'erreur qui est codé par DCT. La transmission se fera par l'envoi de l'adresse du macrobloc de référence dans I, du vecteur de déplacement $\{\Delta x, \Delta y\}$, puis du bloc d'erreur quantifié et compressé entropiquement ;

- le mode interpolé bidirectionnellement (mode B) : ce mode permet d'interpoler le flot d'images en utilisant à la fois une image antérieure I et une image postérieure P pour reconstruire des images intermédiaires notées B (voir la figure 1.9) à l'aide d'une très petite surcharge de transmission (on n'ajoute que les éléments indispensables au

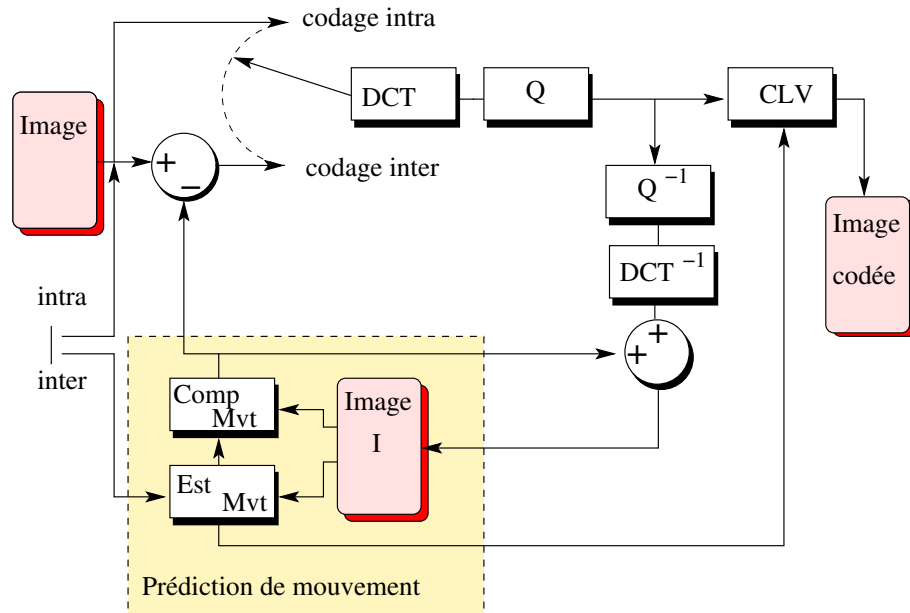


Figure 1.8 – Schéma de principe de codage d'une image dans un flot H.262. Deux modes différents sont possibles : en mode intra-image ou en mode prédictif. DCT est la transformation en cosinus d'un bloc 8×8 . DCT^{-1} est la transformée inverse. Q est un quantificateur et Q^{-1} le déquantifieur. LVC est un codeur par plages à longueur variable. La boîte inférieure gauche est le prédicteur de mouvement. La détection de mouvement est suivie d'une compensation de mouvement permettant de mesurer l'erreur de prédiction qui sera codée dans son bloc par DCT . La décision de basculer en inter ou en intra-image est prise en fonction de la position de l'image dans le flot.

positionnement du macrobloc de B s'il ne se déduit pas par simple interpolation de celui de P). Ce mode de fonctionnement présente néanmoins l'inconvénient d'introduire un délai dans la reconstruction de l'image. Il est surtout utilisé à ce titre aux archivages qui n'imposent pas de temps réel, ou à ceux qui tolèrent une très faible latence.

Ce mode de codage fait apparaître d'importants effets de blocs pour des taux de compression élevés. De nombreuses améliorations ont été proposées pour les éviter : multiplication des vecteurs de déplacement pour un macrobloc, superposition de blocs lors de la recherche du déplacement, filtrage à postériori, etc.

MPEG-4 se distingue de MPEG-2 par une amélioration de la plupart des fonctionnalités : meilleure prédiction des coefficients DCT , meilleure prédiction de mouvement, resynchronisation du flot des données, codage à longueur variable entropique,

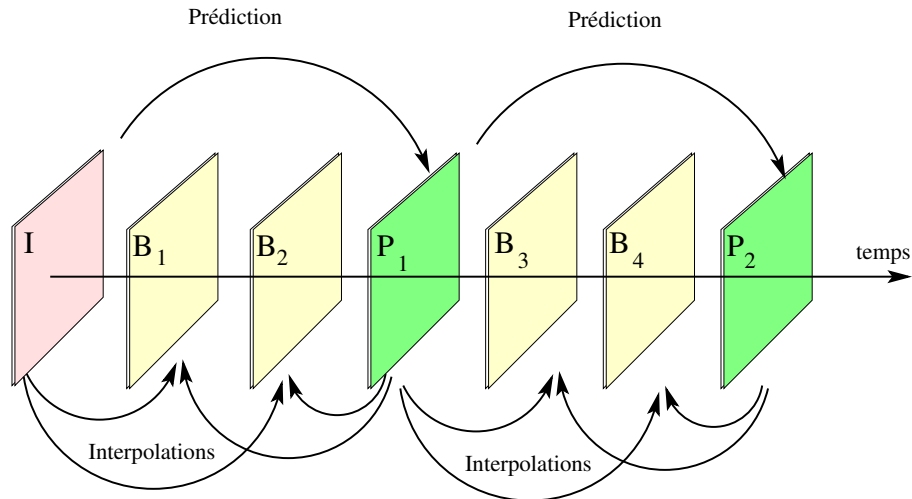


Figure 1.9 – Codage d’un flot d’images par MPEG : I est l’image de référence codée en intra-image. P_1 et P_2 sont les images codées par prédiction de mouvement à partir de I, et pour P_2 de I et P_1 . B_1 et B_2 sont interpolées à partir de I et P_1 , B_3 et B_4 à partir de P_1 et P_2

avec correction d’erreur, etc. Il introduit par exemple dans sa partie 10 un objet nouveau : le VOP (*video object plane*), c’est-à-dire la boîte englobant un objet d’intérêt. Un VOP pourra recevoir un mode particulier de compression, différent du reste de l’image. Un VOP est un nombre entier de macroblochs. Il sera identifié dans le flot des données. Par exemple, un intervenant dans une émission de télévision, parlant sur un fond fixe pourra être situé dans un VOP qui évoluera à haute définition et haute fréquence de rafraîchissement tandis que le fond du studio sera codé à très bas débit. Les VOP, qui sont généralement décrits à la main par l’utilisateur, ne sont pas aujourd’hui considérés par les appareils photographiques (qui exploitent plutôt la partie 2 des recommandations MPEG-4). Néanmoins, certains appareils offrent des fonctionnalités très semblables (par exemple un codage de meilleure qualité dans une zone rectangulaire définie par détection des visages).

La présence de codeurs MPEG-4 au sein des appareils photographiques et leur capacité à encoder des vidéos de grande dimension (format HDTV¹⁸ en 2015 et sur

18. Il faut revenir ici sur le débit introduit par la vidéo en HDTV. Sous la pression des enjeux commerciaux des systèmes d’affichage grand public, le format de la télévision à haute définition (HDTV) est encombré de sous-formats très différents et il n’est pas aisé de savoir à quels chiffres correspondent les performances identifiées par le sigle HDTV affecté à un appareil photographique. Comme nous l’avons dit, la référence en production fixe la résolution de la HDTV à 1 152 lignes de 1 920 pixels et 60 (ou 30) trames par seconde [NIC 03]. Néanmoins,

quelques prototypes, format 4k pendant quelques secondes) est une évolution majeure des appareils photo car c'est cette application qui fixe aujourd'hui la puissance de calcul crête embarquée, qui définit l'architecture du processeur, du processeur de signal, du bus de données et des mémoires tampons.

1.9. Codage comprimé (*Compressed sensing*)

On ne peut terminer un chapitre sur le codage et la compression des images sans aborder même brièvement cette technique qui porte aujourd'hui (2015) beaucoup d'espoirs pour réduire le volume occupé par un signal d'image au-delà des valeurs fournies par JPEG 2000. Néanmoins, le codage comprimé est bien loin de constituer une norme ou un format d'image et ne saurait être recommandé aujourd'hui à qui-conque en concurrence des standards en place [ELD 12].

L'idée sous-jacente au codage comprimé est de reconstruire une image à partir d'un nombre d'échantillons très faible, notablement moins nombreux que ce que recommande le théorème de Shannon¹⁹ [BAR 07, CAN 08, DON 06]. Les pistes proposées consistent à faire quelques mesures pertinentes et très complémentaires permettant de reconstruire l'image par des techniques de reconstruction sophistiquées qui exploitent la parcimonie des signaux d'images, essentiellement par des reconstructions en norme \mathcal{L}_0 ou en norme \mathcal{L}_1 . La parcimonie²⁰ est la propriété d'un signal de pouvoir être décrit dans une base appropriée par un petit nombre de coefficients non nuls seulement. Que les images soient parcimonieuses n'est ni démontré, ni trivial. Il existe de bons arguments et des expériences convaincantes pour soutenir cette hypothèse qui mérite cependant d'être affirmée et définie au cas par cas.

On voit que le codage comprimé place d'entrée de jeu l'acquisition de l'image au cœur du problème de compression (ce que les termes anglo-saxons de *compressed*

les normes d'affichage, précisées par des qualificatifs : HD TNT, HD Ready, HD 1080, sont nettement plus faibles. Le format 4k propose une résolution double en ligne (3 840 pixels d'où le terme 4k) et en colonne (2 304 lignes). C'est le nombre de trames qui sert aujourd'hui à adapter le nombre de pixels traités à la puissance du processeur.

19. Si le codage comprimé permet de reconstruire des fréquences spatiales au-delà de la fréquence de coupure de Nyquist, il n'invalide cependant pas le théorème de Shannon, car la reconstruction obtenue par codage comprimé utilise des hypothèses différentes de celles contenues dans les hypothèses du théorème de Shannon. Shannon pourrait être vu comme un cas particulier de codage comprimé sous l'hypothèse de la parcimonie des signaux sur les bases infinies de Fourier à l'intérieur du domaine fréquentiel $[-B_x, B_x] \times [-B_y, B_y]$. Pour reconstruire tous les signaux de ce type possibles, il est nécessaire de connaître tous les échantillons sur un pas $1/2B_x, 1/2B_y$, mais pour en connaître un en particulier, il est probablement possible d'en mesurer beaucoup moins.

20. Parcimonie se dit *sparsity* en anglais.

sensing font mieux sentir) et que l'on est bien loin du schéma que nous développons dans cet ouvrage d'un capteur qui échantillonne de façon régulière et dense toute image avec le même pas. Néanmoins, cet échantillonnage « à la Shannon » n'est pas à priori incompatible avec un futur codage comprimé. Il laisse cependant au processeur embarqué le soin de calculer la forme compressée qu'un capteur plus malin pourrait peut-être mesurer directement (mais cela reste à montrer pour les photographies de la vie courante).

Si l'on suppose qu'une image se décompose avec un tout petit nombre de coefficients sur une base appropriée, quelle stratégie doit-on adopter pour découvrir quels coefficients sont actifs ? Candès et Tao ont montré [CAN 06] qu'il fallait analyser l'image avec des échantillons de mesure les plus « orthogonaux » possible aux fonctions de bases choisies de façon à ce que même un petit nombre de points d'analyse soit assuré de rendre compte des contributions des fonctions de base. Comme l'on ignore quelles fonctions de base sont actives, Candès a suggéré d'utiliser une famille d'échantillons aléatoires et indépendants [CAN 08]. Par ailleurs, Donoho a donné les conditions pour qu'une reconstruction soit possible en termes de nombre d'échantillons nécessaires en fonction de la parcimonie du signal [DON 09].

Quelle base doit-on choisir ? Nous n'avons pas aujourd'hui de base optimale, mais nous disposons de nombreux candidats : bases de Fourier, DCT, ondelettes (comme celles utilisées dans JPEG), et paquets d'ondelettes utilisés en traitement des images (*curvelets*, *bandelets*, etc.). La reconstruction, si l'on se donne une base, est alors relativement bien maîtrisée. Elle se fait par un gros calcul d'optimisation sous contrainte de parcimonie, soit par exemple en recherchant la fonction de base la plus pertinente parmi celles disponibles, puis en retirant des échantillons les contributions qu'elle explique et en itérant le processus (c'est l'algorithme classique de *matching pursuit* [MAL 93]), soit par optimisation convexe.

Si l'on souhaite transmettre l'image, il n'est pas souhaitable de la reconstruire. On doit s'intéresser à la transmission des échantillons aléatoires. Si l'on veut assurer l'efficacité de la transmission, le caractère aléatoire des échantillons ne rend pas la tâche aisée et même limite sérieusement les possibles performances du codage comprimé, au moins si l'on utilise les formes bien maîtrisées aujourd'hui de la quantification et du codage [GOY 08]. D'autres formes de représentation sont donc probablement nécessaires qu'il importe de développer pour tirer profit du codage comprimé, mais elles demandent encore beaucoup de travaux.

Les études menées aujourd'hui en codage comprimé n'ont pas encore apporté beaucoup de résultats intéressants le domaine de la photographie grand public. Cependant, un très original prototype de caméra à un seul pixel a été proposé, testé [DUA 08], puis commercialisé par la société InView. Il consiste à saisir l'image à l'aide d'une matrice de miroirs individuellement orientables en x et en y de façon à observer à chaque instant une zone différente de la scène. Tous les signaux envoyés

par les miroirs sont alors repris sur un seul capteur, donnant à un instant donné un pixel, mélange complexe de tous ces signaux. Puis la matrice est modifiée selon une loi aléatoire et un nouveau pixel est mesuré. Du flux des pixels successifs, mesuré à la cadence de 1 000 pixels par seconde environ, et à l'aide d'une reconstruction adaptée sur une base d'ondelettes, on peut reconstituer la scène en utilisant environ le dixième des échantillons nécessaires. Cette technologie apparaît particulièrement intéressante dans les longueurs d'onde où les capteurs sont très chers (comme l'infrarouge, domaine où l'entreprise InView a situé son cœur de marché).

Les applications du codage comprimé sont surtout nombreuses dans le champ de l'instrumentation : elles ont conduit à d'intéressantes solutions pour des problèmes complexes de mesure tridimensionnelle en imagerie médicale (tomographie axiale, résonance magnétique [LUS 08]), en microscopie pour la biologie [LEM 12], en holographie pour l'instrumentation physique [DEN 09], etc.

Pour le photographe, le codage comprimé fait clairement partie des enjeux encore très lointains.

Bibliographie

- [ADO 12] ADOBE, Digital Negative (DNG) specification, vers 1.4.0, Adobe Systems Inc., San Jose, Etats-Unis, juin 2012.
- [AHM 74] AHMED N., NATARAJAN T., RAO K. R., « Discrete cosine transform », *IEEE transactions on Computers*, vol. C-23-(1), p. 90-93, jan 1974.
- [BAR 03a] BARLAUD M., ANTONINI M., « Transformées en ondelettes pour la compression d'images », BARLAUD M., LABIT C., Eds., *Compression et codage des images et des vidéos*, Chapitre 3, p. 73-95, Hermès-Lavoisier, Paris, 2003.
- [BAR 03b] BARLAUD M., LABIT C., *Compression et codage des images et des vidéos*, Hermès-Lavoisier, Paris, 2003.
- [BAR 07] BARANIUK C. S., « Compressive Sensing », *IEEE Signal Processing Magazine*, vol. 24, p. 118-121, juillet 2007.
- [CAN 06] CANDÈS E. J., ROMBERG J. K., TAO T., « Stable signal recovery from incomplete and inaccurate measurements », *Communications on Pure and Applied Mathematics*, vol. 58 (8), p. 1207-1222, 2006.
- [CAN 08] CANDÈS E. J., WAKIN M. B., « An introduction to compressive sampling », *IEEE Signal Processing Magazine*, vol. 25 (2), p. 21-30, 2008.
- [COF 14] COFFIN D., « DCRAW », [http : //cybercom.net/~dcoffin/dcraw](http://cybercom.net/~dcoffin/dcraw), 2014.
- [COH 92] COHEN A., DAUBECHIES I., FAUVEAU J.-C., « Biorthogonal bases of compactly supported wavelets », *Communications on Pure and Applied Mathematics*, vol. 45-5, p. 485-560, juin 1992.
- [DEN 09] DENIS L., LORENTZ D., THIÉBAUT E., FOURNIER C., TREDE E., « Inline hologram reconstruction with sparsity constraint », *Optics Letters*, vol. 34(22), p. 3475-3477, 2009.
- [DON 06] DONOHO D. L., « Compressed sensing », *IEEE transactions on Information Theory*, vol. 52 (4), p. 1289-1306, 2006.
- [DON 09] DONOHO D., TANNER J., « Observed universality of phase transitions in high-dimensional geometry with implications for modern data analysis and signal processing », *Philosophical Transactions of the Royal Society, A*, vol. 367, p. 4273-4330, 2009.

- [DUA 08] DUARTE M. F., DAVENPORT M. A., TAKHAR D., LASKA J. N., SUN T., KELLY K. F., BARANIUK R. C., « Single-pixel imaging via compressive sampling », *IEEE Signal Processing Magazine*, vol. 25(2), p. 83-91, mars 2008.
- [ELD 12] ELДАР Y., KUTYNIОK G., *Compressed sensing : Theory and applications*, Cambridge University Press, Cambridge, 2012.
- [GOY 08] GOYAL V. K., FLETCHER A. K., RAGAN S., « Compressive sampling and lossy compression », *IEEE Signal Processing Magazine*, vol. 25(2), p. 48-56, mars 2008.
- [GUI 03a] GUILLEMOT C., PATEUX S., « Eléments de théorie de l'information et de communication », BARLAUD M., LABIT C., Eds., *Compression et codage des images et des vidéos*, Chapitre 1, p. 21-41, Hermès-Lavoisier, Paris, 2003.
- [GUI 03b] GUILLOIS J., CHARRIER M., LAMBERT C., PAUCARD B., « Standards de compression d'images fixes », BARLAUD M., LABIT C., Eds., *Compression et codage des images et des vidéos*, Chapitre 4, p. 97-130, Hermès-Lavoisier, Paris, 2003.
- [HEA 14] HEALPIX, « Hierarchical Equal Area isolatitude projection of a sphere », healpix.sourceforge.net, janvier 2014.
- [HOR 14] HORVÀTH G., « RawTherapee », www.rawtherapee.com, 2014.
- [ISO 10] ISO, « ISO 12234-2 :2001 Electronic still-picture imaging – removable memory – Part 2 : TIFF/EP Image Data Format », Norme ISO, 2010.
- [KOP 06] KOPOSOV S., BARTUNOV O., « Q3CQuad Tree Cube, the new sky-indexing concept for huge astronomical catalogues. », *Astronomical Data Analysis Softwares and Systems*, vol. 351, page735, 2006.
- [KUN 01] KUNSZT P. Z., SZALAY A. S., THAKAR A. R., « The hierarchical triangular mesh. », LEWIS J., Ed., *Mining the sky*, p. 631-637, Springer., Berlin Heidelberg, 2001.
- [LEM 12] LE MONTAGNER Y., ANGELINI E., OLIVO-MARIN J. C., « Video Reconstruction using compressed sensing measurements and 3D total variation regularization for bio-imaging applications », *IEEE International Conference on Image Processing ICIP-12*, Orlando (Florida), p. 917-920, 2012.
- [LUS 08] LUSTIG M., DONOHO D. L., SANTOS J. M., PAULY J. M., « Compressed sensing MRI », *IEEE Signal Processing Magazine*, vol. 25(2), p. 78-82, 2008.
- [MAL 93] MALLAT S., ZHANG Z., « Matching pursuits with time-frequencies dictionaries », *IEEE transactions on Signal Processing*, vol. 12, p. 3397-3415, décembre 1993.
- [NIC 03] NICOLAS H., GUILLEMOT C., « Normes de compression vidéo », BARLAUD M., LABIT C., Eds., *Compression et codage des images et des vidéos*, Chapitre 6, p. 177-208, Hermès-Lavoisier, Paris, 2003.
- [PER 02] PEREIRA F., EBRAHIMI T., *The MPEG-4 book*, Prentice-Hall, Upper Saddle River, Etats-Unis, 2002.
- [PRO 03] PROST R., « Compression sans perte », BARLAUD M., LABIT C., Eds., *Compression et codage des images et des vidéos*, Chapitre 9, p. 285-305, Hermès-Lavoisier, Paris, 2003.

- [SWE 96] SWELDENS W., « The lifting scheme : a custom-design construction of biorthogonal wavelets », *Applied and Computational Harmonic Analysis*, vol. 3-2, p. 186-200, avril 1996.
- [TAU 00] TAUBMAN D., « High performance scalable image compression with EBCOT », *IEEE transactions on Image Processing*, vol. 9-7, 2000.
- [WOO 11] WOODS J. W., *Multidimensional Signal, Image and Video Processing and Coding*, Academic Press, San Diego, Etats Unis, 2nd édition, 2011.

BibT_EX BibT_EX

Index

- alpha (canal), 22
- canal α , 22
- codage
 - JPEG-LS, 21
 - comprimé, 41
 - de Huffman, 15, 20
 - de Lempel-Ziv, 15, 20
 - hiérarchique, 33
 - vidéo, 38
- compressed sensing, 41
- compression sans perte, 19
- comprimé (codage), 41
- DCRAW, 10
- DCT (transformation), 25
- dithering, 25
- DNG (format), 16
- EBCOT (codeur), 31
- Exif (fichier), 17
- FlashPix (format), 35
- format
 - DNG, 16
 - FlashPix, 35
 - GIF, 25
 - HDTV, 41
 - HEALPix, 36
 - HTM, 36
 - JPEG, 26
 - JPEG 2000, 31
 - MPEG, 38
 - PNG, 22
 - Q3C, 36
 - TIFF, 23
 - tuilé, 35
- GIF (format), 25
- graphique (image), 15
- HDTV (format), 41
- HEALPix (format), 36
- HTM (format), 36
- Huffman (codage), 15, 20
- image
 - graphique, 15
 - vidéo, 37
- JBIG, 20
- JPEG
 - (codage), 26
 - 2000 (codage), 31
 - sans-perte (codage), 21
- Lempel-Ziv (codage), 15, 20
- lifting (schéma de), 33
- matching pursuit, 42
- métadonnées, 8, 16
- MPEG (codage), 38
- natif (format), 9
- palette, 25
- parcimonie, 41
- PNG (format), 22
- Q3C (format), 36
- RAW (format), 9
- scalabilité, 22, 31
- schéma de lifting, 33
- sparsity, 41
- standard XMP, 17

TIFF (format), 23
TIFF/EP (format), 15
tramage, 25
tuilé (format), 35
TVHD = Télévision haute définition,
38
UIT = Union Internationale des Télé-
communications, 37
XMP (standard), 17
zig-zag (balayage), 27