

Le format JPEG

Principe du format JFIF/JPEG

Le format d'un fichier JFIF/JPEG est basé sur des sections. Chaque section permet de représenter une partie du format. Afin de se repérer dans le flux JPEG, on utilise des marqueurs, ayant la forme `0xff??`, avec le `??` qui permet de distinguer les marqueurs entre eux. **La norme impose que les marqueurs soient toujours alignés dans le flux sur un multiple d'octets.**

Chaque section d'un flux JPEG a un rôle spécifique, et la plupart sont indispensables pour permettre le décodage de l'image. Nous vous donnons dans la suite de cette annexe une liste des marqueurs JPEG que vous pouvez rencontrer.

Petit point sur les indices et identifiants

Afin de faire les associations entre éléments, le JPEG utilise différents types d'indices. On en distingue trois :

- les **identifiants** des composantes de couleur, qu'on notera i_C ;
- les **indices** de table de Huffman, qui sont en fait la concaténation de deux indices $i_{AC/DC}, i_H$;
- et les **indices** de table de quantification, qu'on notera i_Q .

L'identifiant d'une composante est un entier entre 0 et 255. Les indices des tables sont eux des "vrais" indices : 0, 1, 2, etc. Une table de Huffman se repère par le type de coefficients qu'elle code, à savoir les constantes DC ou les coefficients fréquentiels AC, et par l'indice de la table dans ce type, i_H .

Afin de pouvoir décoder chaque composante de l'image, l'en-tête JPEG donne les informations nécessaires pour :

- associer une table de quantification i_Q à chaque i_C ;
- associer une table de Huffman $i_{AC/DC}, i_H$ pour chaque couple $(i_{AC/DC}, i_C)$.

Sections JPEG

Le format général d'une section JPEG est le suivant :

Offset	Taille (octets)	Description
0x00	2	Marqueur pour identifier la section
0x02	2	Longueur de la section en nombre d'octets, y compris les 2 octets coda
0x04	?	Données associées (dépendent de la section)

La longueur de la section indique combien d'octets on doit lire *au décodage*, une fois qu'on a lu le marqueur de section, pour lire l'intégralité de la section. Elle tient donc compte des deux octets permettant de coder cette longueur.

Marqueurs de début et de fin d'image

Toute image JPEG débute par un marqueur SOI (*Start of Image*) 0xffd8 et termine par un marqueur EOI (*End of Image*) 0xffd9. Ces deux marqueurs font exception dans le JPEG puisqu'ils ne suivent pas le format classique décrit ci-dessus : ils sont utilisés sans aucune autre information et servent de repères.

Bien qu'il soit possible qu'un fichier contienne plusieurs images (format de vidéo MJPEG, pour *Motion JPEG*), nous nous limiterons dans ce projet au cas d'une seule image par fichier.

APPx - *Application data*

Cette section permet d'enregistrer des informations propres à chaque *application*, application signifiant ici format d'encapsulation. Dans notre cas, on ne s'intéressera qu'au marqueur APP0, qui sert pour l'encapsulation JFIF. On ne s'intéresse pas aux différentes informations dans ce marqueur. Les seules choses qui nous intéressent sont la séquence des 4 premiers octets de la section, qui doit contenir la phrase JFIF, et le numéro de version JFIF X.Y codé sur deux octets (un pour X, un pour Y).

Offset	Taille (octets)	Description
0x00	2	Marqueur APP0 (0xffe0)
0x02	2	Longueur de la section
0x04	5	'J' 'F' 'I' 'F' '\0'
0x09	1	Version JFIF (1.1) : doit valoir 1
0x0A	1	Version JFIF (1.1) : doit valoir 1
0x0B	7	Données spécifiques au JFIF, non traitées : tout mettre à 0

COM - Commentaire

Afin de rajouter des informations textuelles supplémentaires, il est possible d'ajouter des sections de commentaires dans le fichier (par exemple, on trouve parfois le nom de l'encodeur). Notez que cela nuit à l'objectif de compression, les commentaires étant finalement des informations inutiles.

Offset	Taille (octets)	Description
0x00	2	Marqueur COM (0xfffe)
0x02	2	Longueur de la section
0x04	?	Données

DQT - *Define Quantization Table*

Cette section permet de définir une ou plusieurs tables de quantification. Il y a généralement plusieurs tables de quantification dans un fichier JPEG (souvent 2, au maximum 4). Ces tables sont repérées à l'aide de l'indice i_Q défini plus haut. C'est ce même indice, défini dans une section DQT, qui est utilisé dans la section SOF pour l'association avec une composante.

Un fichier JPEG peut contenir soit une seule section DQT avec plusieurs tables, soit plusieurs sections DQT avec une table à chaque fois. C'est la longueur d'une section qui permet de déterminer combien de tables elle contient. Si une table de quantification a le même indice i_Q qu'une table de quantification précédemment lue, alors cette table est redéfinie avec les nouvelles données lues.

Offset	Taille (octets)	Description
0x00	2	Marqueur DQT (0xffdb)
0x02	2	Longueur de la section
...	4 bits	Précision (0 : 8 bits, 1 : 16 bits)
	4 bits	Indice i_Q de la table de quantification
...	64	Valeurs de la table de quantification, stockées au format zig-zag (cf. sec

SOFx - *Start Of Frame*

Le marqueur SOF définit le début effectif d'une image, et donne les informations générales rattachées à cette image. Il existe plusieurs marqueurs SOF selon le type d'encodage JPEG utilisé. Dans le cadre de ce projet, nous ne nous intéressons qu'au JPEG *baseline sequential*, *DCT*, *Huffman*, *8 bits*, soit SOF0 (0xffc0). Pour information, les autres types sont récapitulés en section 6.3.

Les informations générales associées à une image sont la précision des données (le nombre de bits codant chaque coefficient, toujours 8 dans notre cas), les dimensions de l'image, et le nombre de composantes de couleur utilisées (1 en niveaux de gris, 3 en YCbCr).

Pour chacune de ces composantes sont définis :

- un identifiant i_C entre 0 et 255, qui sera référencé dans la section SOS ;
- les facteurs d'échantillonnage horizontal et vertical. Comme décrit section 2.5.1, ces facteurs h x v indiquent le *nombre de blocs par MCU* codant la composante (dans le cas de Y, il s'agit donc de la taille de la MCU) ;
- l'indice i_Q de la table de quantification associée à la composante.

Dans cette section SOF (et ce n'est garanti qu'ici), l'ordre des composantes est toujours le même : d'abord Y, puis Cb puis Cr. Les identifiants sont normalement fixés à 1, 2 et 3. Cependant, certains encodeurs ne suivent pas cette obligation et donc un décodeur doit savoir gérer des identifiants quelconques entre 0 et 255.

Finalement, une section SOF suit le format :

Offset	Taille (octets)	Description
0x00	2	Marqueur SOFx : 0xffc0 pour le SOF0
0x02	2	Longueur de la section
0x04	1	Précision en bits par composante, toujours 8 pour le baseline
0x05	2	Hauteur en pixels de l'image
0x07	2	Largeur en pixels de l'image
0x09	1	Nombre de composantes N (Ex : 3 pour le YCbCr, 1 pour les niveaux de
0x0a	3N	N fois : <ul style="list-style-type: none">• 1 octet : Identifiant de composante i_C, de 0 à 255• 4 bits : Facteur d'échantillonnage (<i>sampling factor</i>) horizontal, de 1 à 4• 4 bits : Facteur d'échantillonnage (<i>sampling factor</i>) vertical, de 1 à 4• 1 octet : Table de quantification i_Q associée, de 0 à 3

DHT - *Define Huffman Table*

La section DHT permet de définir une (ou plusieurs) table(s) de Huffman, selon le format décrit en section 2.8.1. Pour chaque table sont aussi définis ses indices de repérage $i_{AC/DC}$ et i_H .

Comme pour DQT, une section DHT peut contenir une ou plusieurs tables. Dans ce dernier cas, il y a en fait répétition des 3 dernières cases du tableau suivant. La longueur en octets de la section représente la taille nécessaire pour stocker toutes les tables contenues. Au décodage, pour déterminer si une section contient plusieurs tables, il faut donc regarder combien d'octets ont été lus pour construire une table. S'il en reste, la section contient encore (au moins) une autre table. A l'encodage, libre à vous, lorsque plusieurs tables de Huffman sont utilisées, de les définir dans autant de sections DHT, ou de les rassembler dans une seule section.

Dans un fichier, il ne peut pas y avoir plus de 4 tables de Huffman par type AC ou DC (sinon, le flux JPEG est corrompu).

Offset	Taille (octets)	Description
0x00	2	Marqueur DHT (0xffc4)
0x02	2	Longueur de la section
0x04		Informations sur la table de Huffman :
	3 bits	☐ non utilisés, doit valoir 0 (sinon erreur)
	1 bit	☐ type (0=DC, 1=AC)
	4 bits	☐ indice (0..3, ou erreur)
0x05	16	Nombres de symboles avec des codes de longueur 1 à 16. La somme de
0x15	?	Table contenant les symboles, triés par longueur (cf 2.8.1)

SOS - *Start Of Scan*

Une section SOS contient des données brutes encodant l'image. Un fichier séquentiel ne contient qu'une seule section SOS, mais un fichier progressif peut en contenir plusieurs.

L'en-tête d'une section SOS (on parle de *Scan Header*) contient:

- le nombre de composantes du *scan* ;
- les identifiants de(s) composante(s) dans l'ordre où elles apparaissent dans le flux ;
- les associations entre composantes et tables de Huffman ;
- les informations de sélection (fréquences) et d'approximation des coefficients AC pour le mode progressif. Pour plus d'informations rendez-vous en section B.2.3 (page 37) de la norme JPEG (document [itu-t81.pdf](#) distribué avec le sujet). Dans le cadre de ce projet, en mode *baseline sequential*, ces trois octets doivent prendre les valeurs respectives 0, 63 et 0.

Si un *scan* contient plusieurs composantes, elles sont entrelacées MCU par MCU. D'après la norme, l'**ordre** des composantes devrait être le même que dans la section SOF, soit Y, Cb puis Cr. Mais puisque certains encodeurs ne respectent pas cette convention, seul l'identifiant i_C d'une composante permet de l'associer correctement à la bonne composante de couleur. Les composantes apparaissent donc dans le flux dans l'ordre des indices i_C de cette section SOS.

Après le *Scan Header*, les données brutes sont ensuite stockées par blocs 8x8 encodés RLE + Huffman, dans l'ordre des composantes indiqué dans l'en-tête. Le nombre de blocs par composante de MCU dépend des facteurs d'échantillonnage lus en section SOF. Leur ordre est spécifié en 2.5.2.

Offset	Taille (octets)	Description
$0x00$	2	Marqueur SOS
$0x02$	2	Longueur de la section (données brutes non comprises)
$0x04$	1	N = Nombre de composantes. La longueur de la section vaut $2N + 6$
$0x05$	2N	N fois : <ul style="list-style-type: none"> 1 octet : identifiant i_C de la composante 4 bits : indice de la table de Huffman (i_H) pour les coefficients DC (4 bits : indice de la table de Huffman (i_H) pour les coefficients AC (i
...	1	Ss : Premier indice de la sélection spectrale : doit valoir 0 en mode bas
...	1	Se : Dernier indice de la sélection spectrale : doit valoir 63 mode basel
...	1	Approximation successive : <ul style="list-style-type: none"> Ah : 4 bits, poids fort : doit valoir 0 mode baseline Al : 4 bits, poids faible : doit valoir 0 en mode baseline

Récapitulatif des marqueurs

Code	Nom	Description
$0x00$		Byte stuffing (ce n'est pas un marqueur !)
$0x01$	TEM	
$0x02$... $0xbf$	Réservés (not used)	
$0xc0$	SOF0	Baseline DCT (Huffman)
$0xc1$	SOF1	DCT séquentielle étendue (Huffman)

Code	Nom	Description
0xc2	SOF2	DCT Progressive (Huffman)
0xc3	SOF3	DCT spatiale sans perte (Huffman)
0xc4	DHT	Define Huffman Tables
0xc5	SOF5	DCT séquentielle différentielle (Huffman)
0xc6	SOF6	DCT séquentielle progressive (Huffman)
0xc7	SOF7	DCT différentielle spatiale (Huffman)
0xc8	JPG	Réservé pour les extensions du JPG
0xc9	SOF9	DCT séquentielle étendue (arithmétique)
0xca	SOF10	DCT progressive (arithmétique)
0xcb	SOF11	DCT spatiale (sans perte) (arithmétique)
0xcc	DAC	Information de conditionnement arithmétique
0xcd	SOF13	DCT Séquentielle Différentielle (arithmétique)
0xce	SOF14	DCT Différentielle Progressive (arithmétique)
0xcf	SOF15	Progressive sans pertes (arithmétique)
0xd0 ... 0xd7	RST0 ... RST7	Restart Interval Termination
0xd8	SOI	Start Of Image (Début de flux)
0xd9	EOI	End Of Image (Fin du flux)
0xda	SOS	Start Of Scan (Début de l'image compressée)
0xdb	DQT	Define Quantization tables
0xdc	DNL	
0xdd	DRI	Define Restart Interval
0xde	DHP	
0xdf	EXP	
0xe0 ... 0xef	APP0 ... APP15	Marqueur d'application
0xf0 ... 0xfd	JPG0 ... JPG13	
0xfe	COM	Commentaire