



## PPM



Taille mémoire : 1,3 MB (800 × 530)

## Compression

Taux de compression  $\rho$  :

- $T_i$  : taille initiale
- $T_f$  : taille finale

$$\rho = \frac{T_i - T_f}{T_i}$$

## Compression

**Compression sans perte :**

- réduire la taille mémoire
- ne pas altérer l'image
- utilise les redondances de l'images

**Compression avec perte :**

- réduire très efficacement la taille mémoire
- enlever les informations les moins importantes

## GIF

**Graphics Interchange Format :**

- 1987, CompuServe
- compression sans perte
- gère la transparence (booléenne)
- ne supporte que 256 couleurs parmi les 16 millions habituelles
- gère les animations

## GIF



## Palette :

- génère une palette de 256 valeurs RGB
- transparence (booléenne)
- chaque pixel de l'image fait référence à cette palette
- mal adapté pour les dégradé et les images naturelles
- bien adapté pour les images synthétiques (logo, ...)

## GIF

## Compression :

- LZW :Lempel-Ziv-Welch
- 1978-84
- sans pertes
- regroupe les redondances de l'image

## LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire
				1 a
				2 b
				3 c

Code :

## LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire
	a			1 a
				2 b
				3 c

Code :

# LZW

Texte : a **b** a b c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
				3	c
				4	ab

Code : 1

# LZW

Texte : a b a **b** c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
				5	ba

Code : 1 2

# LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
				4	ab
				5	ba

Code : 1 2

# LZW

Texte : a b a b **c** a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
ab	c	abc = 6	4	5	ba
				6	abc

Code : 1 2 4

# LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
ab	c	abc = 6	4	5	ba
c	a	ca = 7	3	6	abc
				7	ca

Code : 1 2 4 3

# LZW

Texte : a b a b c a **b** a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
ab	c	abc = 6	4	5	ba
c	a	ca = 7	3	6	abc
a	b			7	ca

Code : 1 2 4 3

# LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
ab	c	abc = 6	4	5	ba
c	a	ca = 7	3	6	abc
a	b			7	ca
ab	a	aba = 8	4	8	aba

Code : 1 2 4 3 4

# LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire	
␣	a			1	a
␣a	b	ab = 4	1	2	b
b	a	ba = 5	2	3	c
a	b			4	ab
ab	c	abc = 6	4	5	ba
c	a	ca = 7	3	6	abc
a	b			7	ca
ab	a	aba = 8	4	8	aba
a	c	ac = 9	1	9	ac

Code : 1 2 4 3 4 1

## LZW

Texte : a b a b c a b a c

dernier	courrant	entrée	code	dictionnaire
	a			1 a
a	b	ab = 4	1	2 b
b	a	ba = 5	2	3 c
a	b			4 ab
ab	c	abc = 6	4	5 ba
c	a	ca = 7	3	6 abc
a	b			7 ca
ab	a	aba = 8	4	8 aba
a	c	ac = 9	1	9 ac
c	EOF		3	

Code : 1 2 4 3 4 1 3

## GIF



Taille mémoire : 114 kB (800 × 530)

## GIF



Taille mémoire : 168 kB (800 × 530)

## JPEG

### Joint Photographic Experts Group :

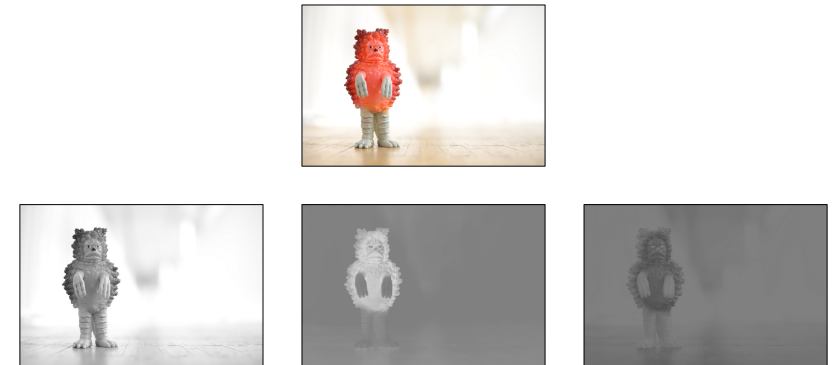
- spécifications 1991
- compression avec perte
- exif

## Les étapes

### Conversion RGB → YCbCr :

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{bmatrix} 0.229 & 0.587 & 0.114 \\ -0.1687 & -0.3313 & 0.5 \\ 0.5 & -0.4187 & -0.0813 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix}$$

## Les étapes



### Conversion RGB → YCbCr :

- l'œil est moins sensible à la chrominance

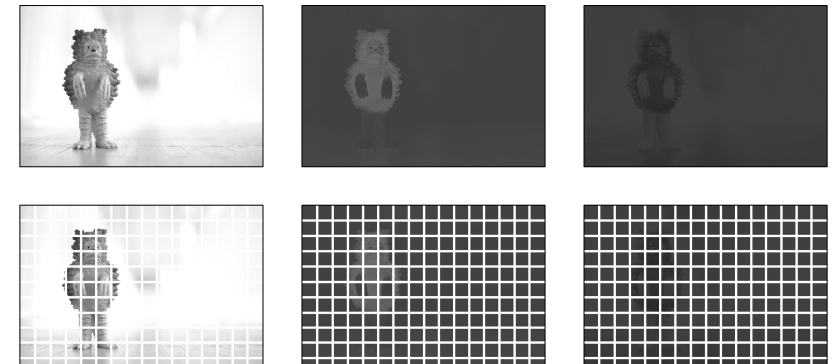
## Les étapes

### Sous échantillonnage de moitié de Cb Cr :



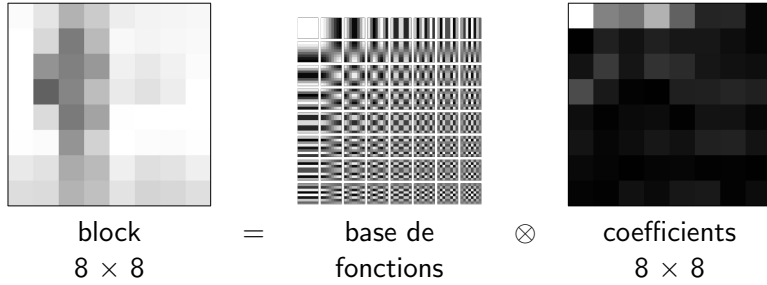
## Les étapes

### Découpage par blocs de 8 × 8 :

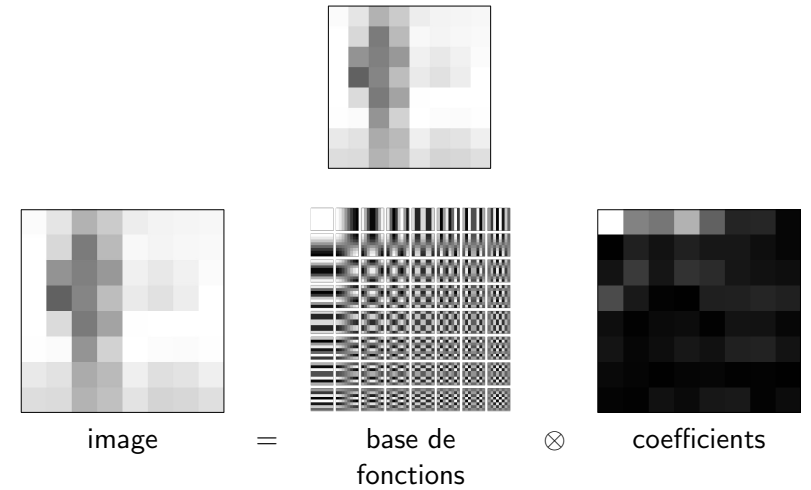


## Les étapes

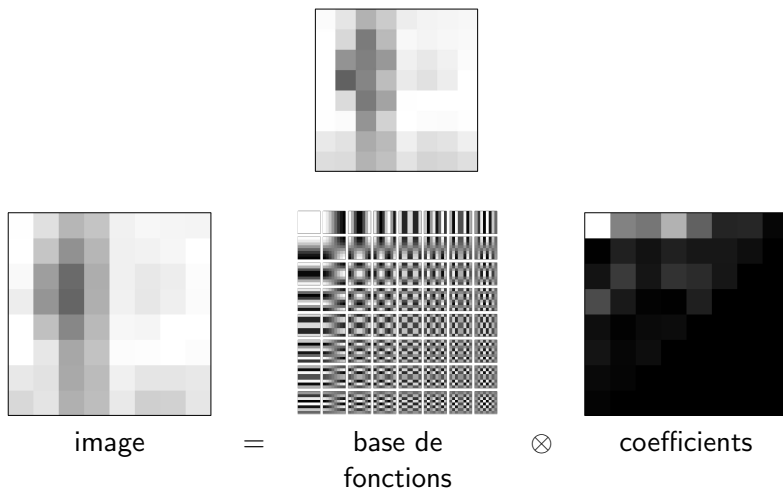
Pour chaque bloc  $8 \times 8$  : DCT



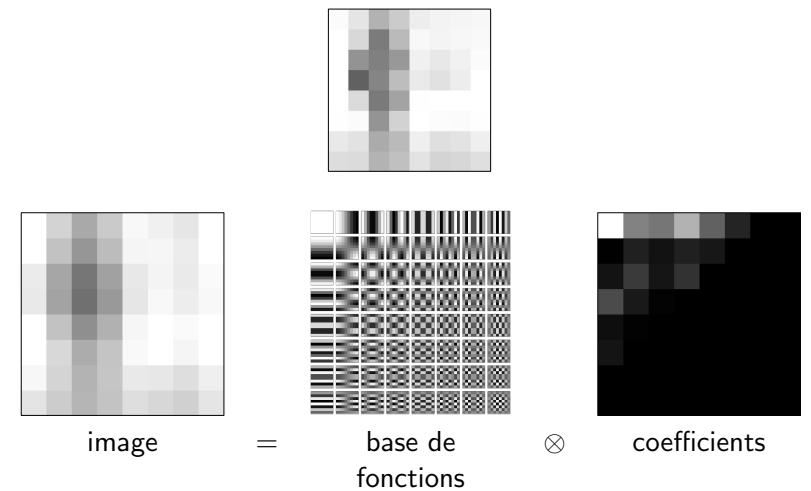
## DCT : les hautes fréquences



## DCT : les hautes fréquences

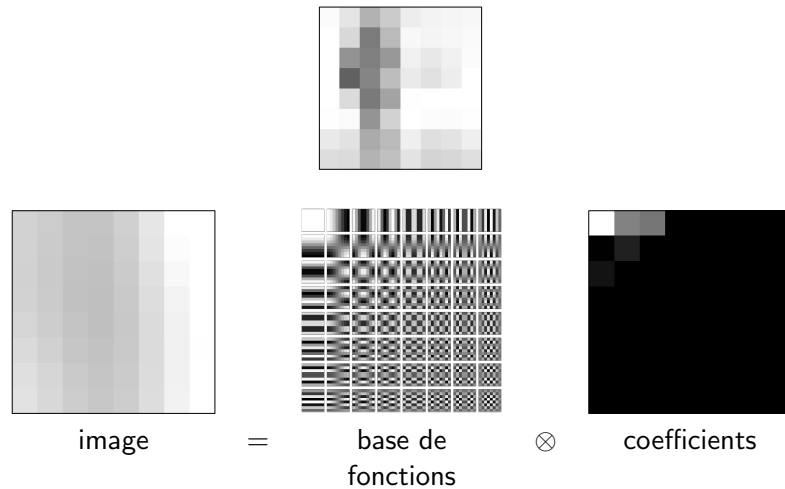


## DCT : les hautes fréquences



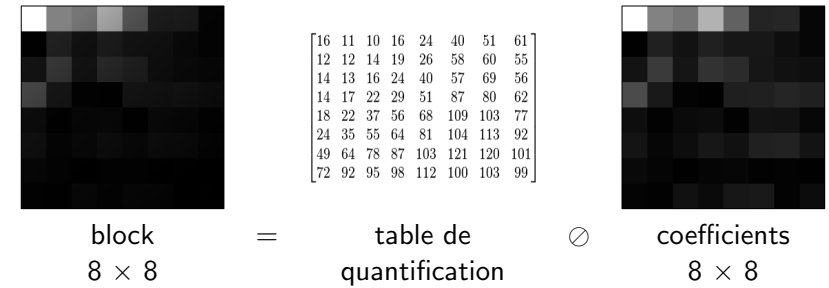


## DCT : les hautes fréquences



## Les étapes

### Quantification



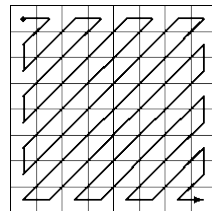
### Taux de compression :

c'est le choix de la table qui définit le taux de compression.

## Les étapes

### RLE et Huffman :

79	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



### Texte à compresser :

79 0 -2 -1 -1 -1 -1 0 0 -1 0 ...

## Les étapes

### Texte à compresser :

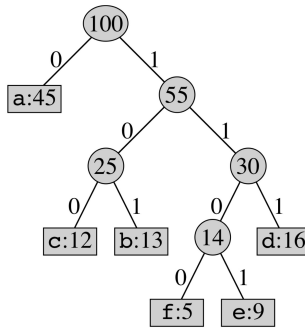
79 0 -2 -1 -1 -1 -1 0 0 -1  $\underbrace{0 \dots 0}_{54}$

### RLE: Run Length Encoding

1 79 1 0 1 -2 4 -1 2 0 1 -1 54 0

# Les étapes

## Huffman:



lettres	a	b	c	d	e	f
occurrence (%)	45	13	12	16	9	5

# JPEG

## Principe :

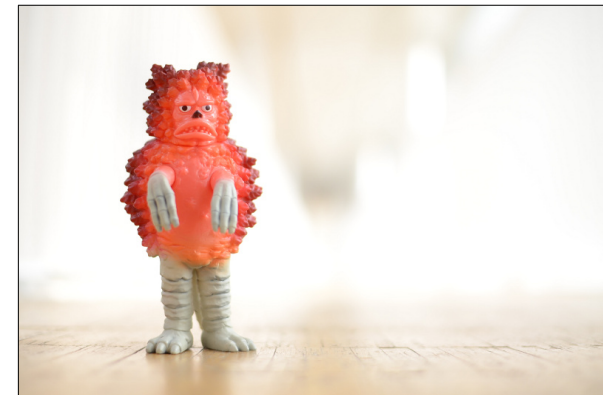
- ❶ conversion RGB → YCbCr
- ❷ sous échantillonnage de moitié de Cb Cr
- ❸ découpage par bloc
- ❹ pour chaque bloc : DCT
- ❺ quantification
- ❻ compression RLE et Huffman

# JPEG



**Compression : 100**  
**Taille mémoire : 223,3 kB (800 × 530)**

# JPEG



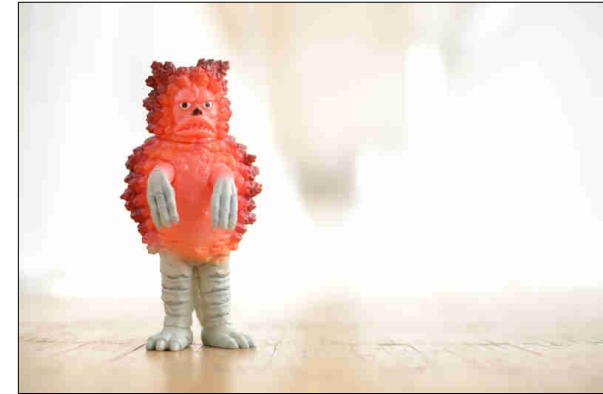
**Compression : 95**  
**Taille mémoire : 77,1 kB (800 × 530)**

# JPEG



**Compression : 70**  
**Taille mémoire : 26,1 kB (800 × 530)**

# JPEG



**Compression : 20**  
**Taille mémoire : 12 kB (800 × 530)**

# JPEG



**Compression : 5**  
**Taille mémoire : 7,1 kB (800 × 530)**

# Fourier vs. DCT

DCT	Fourier
coefficients réels	coefficients complexes
gère bien les discontinuités	gère mal les discontinuités

## JPEG-2000

### Le format :

- avec ou sans perte
- ondelettes
- meilleur qualité et compression que jpeg
- codes correcteurs

## JPEG-2000

### Le pipeline → comme jpeg, sauf :

- DCT ( $8 \times 8$ ) → ondelettes (toute l'image)
- ↔ ondelettes : pas d'effet de blocs
- codage entropique beaucoup plus sophistiqué (EBCOT)

## JPEG-2000



**Compression :** sans perte  
**Taille mémoire :** 42,5 kB ( $800 \times 530$ )

## JPEG-2000

### JPEG vs. JPEG-2000 : (même taille de fichier)



# PNG

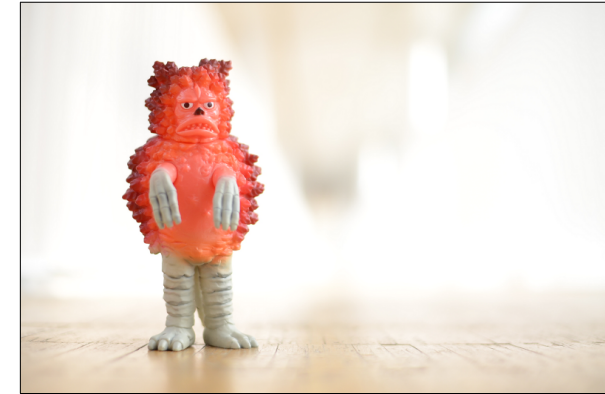
## Portable Network Graphics :

- 1995
- conçu en alternative au gif, alors sous brevet
- gère la transparence et translucidité (RGBA)

## Compression : LZ-77

- compression sans perte
- très similaire à gif
- pas spécifique à l'image

# PNG



**Compression : sans perte**  
**Taille mémoire : 324,5 kB (800 × 530)**

# HEIF

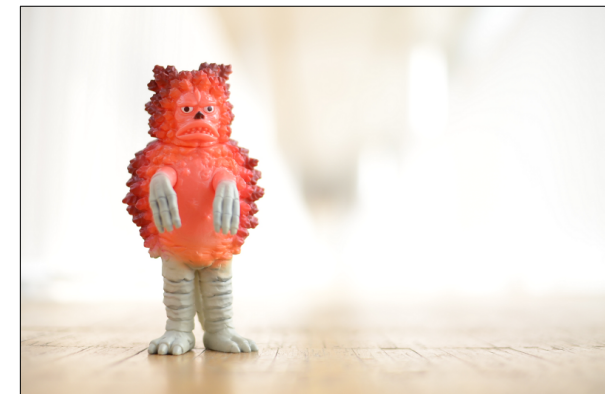
## High Efficiency Image File Format :

- 2020 (iOS-11) → `file.heic`
- conçu en alternative au jpeg2000
- gère les séquences d'images

## Compression :

- conteneur et non méthode de compression à proprement parler.

# HEIF



**Compression : sans perte**  
**Taille mémoire : 53,2 kB (800 × 530)**