DIRO
IFT 6150

TRAITEMENT D'IMAGES

COMPRESSION D'IMAGES

Max Mignotte

Département d'Informatique et de Recherche Opérationnelle.
Http : //www.iro.umontreal.ca/~mignotte/ift6150
E-mail : mignotte@iro.umontreal.ca

COMPRESSION D'IMAGES

SOMMAIRE

COMPRESSION D'IMAGES
INTRODUCTION (1)

But de la compression

Réduire le volume des données (i.e., le nb. de bits) nécessaire pour représenter et coder les caractéristiques d'une image (i.e., élimine la redondance d'information)

- ▶ Réduction du coût de stockage
- ▶ Transmission rapide des données

Taux de compression

$$C = \frac{\text{Nb. de bits après compression}}{\text{Nb. de bits avant compression}}$$

Type de compression

- 2 grandes familles ; avec ou sans perte

3 types de redondances

- Redondance de codage (compression RLE, etc.)
- Redondance spatiale (codage d'Huffman, etc.)
- Redondance visuelle (jpeg, fractale, etc.)

Mesure de la qualité de la compression d'une image

$$\text{SNR} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f^2(x,y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \Big(f(x,y) - f_c(x,y)\Big)^2}$$

COMPRESSION D'IMAGES
COMPRESSION RLE

Compression RLE

0	0	0	11	32	67	96	128	128	128	96	96	37

37	37	37	37	0	0	0	0	0	0	2	2

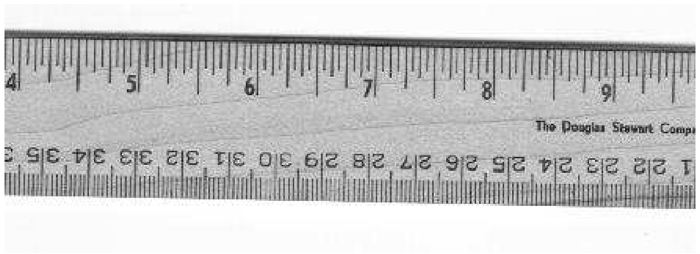
3	0	1	11	1	32	1	67	1	96	3	128	2	96	5	37	6	0	2	2

Principe

- RLE : Run Length Encoding
- Création d'une nouvelle séquence dans laquelle le deuxième élément correspond au niveau de gris et le premier élément correspond au nombre de pixels consécutif possédant ce niveaux de gris
- On code séparemment le niveaux de gris et l'ocurence de chaque pixel

▼

Fort taux de compression pour des images possédant de nb. zones de régions homogènes



COMPRESSION D'IMAGES
COMPRESSION D'HUFFMAN (1)

Principe

Utilisation d'un codage de longueur variable qui assigne le plus petit code (en longueur) pour les niveaux de gris d'occurence élevé

Original source		Source reduction			
N de G	Probability	1	2	3	4
1	0.4	0.4	0.4	0.4	0.6
5	0.3	0.3	0.3	0.3	0.4
0	0.1	0.1	0.2	0.3	
3	0.1	0.1	0.1		
2	0.06	0.1			
4	0.04				

- Calcul de l'histogramme
- Constitution d'une table ou les niveaux de gris sont arrangés par ordre décroissant de probabilité
- Réduire le nombre de probabilité en combinant les deux plus faibles probabilités pour former une nouvelle probabilité
- Assigner la valeur du code pour chaque niv. de gris

Original source			Source reduction							
N de G	Prob.	Code	1		2		3		4	
1	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
5	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
0	0.1	011	0.1	011	0.2	010	0.3	01		
3	0.1	0100	0.1	0100	0.1	011				
2	0.06	01010	0.1	0101						
4	0.04	01011								

COMPRESSION D'IMAGES
COMPRESSION D'HUFFMAN (2)

Original source			Source reduction							
N de G	Prob.	Code	1		2		3		4	
1	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
5	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
0	0.1	011	0.1	011	0.2	010	0.3	01		
3	0.1	0100	0.1	0100	0.1	011				
2	0.06	01010	0.1	0101						
4	0.04	01011								

Nb. Moyen de bits utilisé pour le codage

$$\text{Nb Moyen de bits} = (1 \times 0.4) + (2 \times 0.3) + (3 \times 0.1) + (4 \times 0.1) + (5 \times 0.06) + (5 \times 0.04) = 2.2 \text{ bits}$$

Borne supérieure

$$\boxed{\text{Nb Moyen de bits} > \text{Entropy} = -\sum_{i=0}^{G_{\max}} h_i \log_2(h_i)}$$

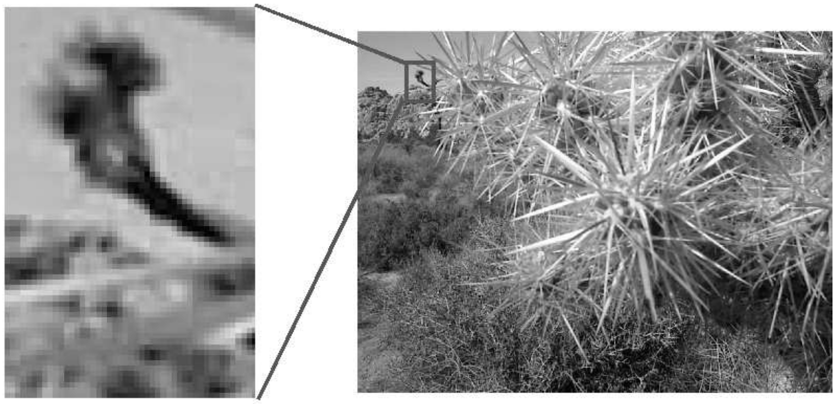
- h_i : probabilité d'avoir le niveaux de gris $= i$
- $G_{\max}$: Nb de niveaux de gris maximal

Exemple

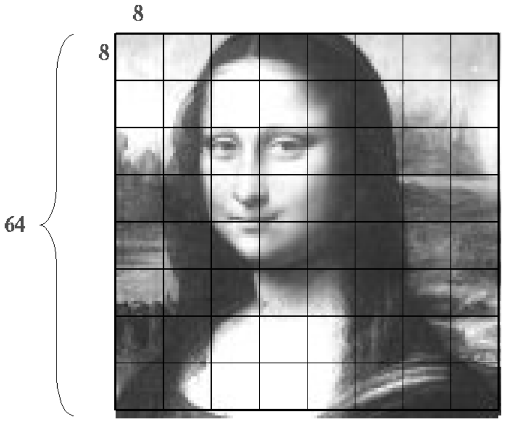
$$\text{Nb Moyen de bits} > 0.52877 + 0.52109 + 0.33219 + 0.33219 + 0.243533 + 0.18575 = 2.1435$$

COMPRESSION D'IMAGES
COMPRESSION JPEG (1)

- Forte compression mais avec perte (25 : 1)
- Basée sur la Transformée Cosinus Discrète (DCT) rapide (FDCT) 2D de sous images de tailles 8×8 pixels



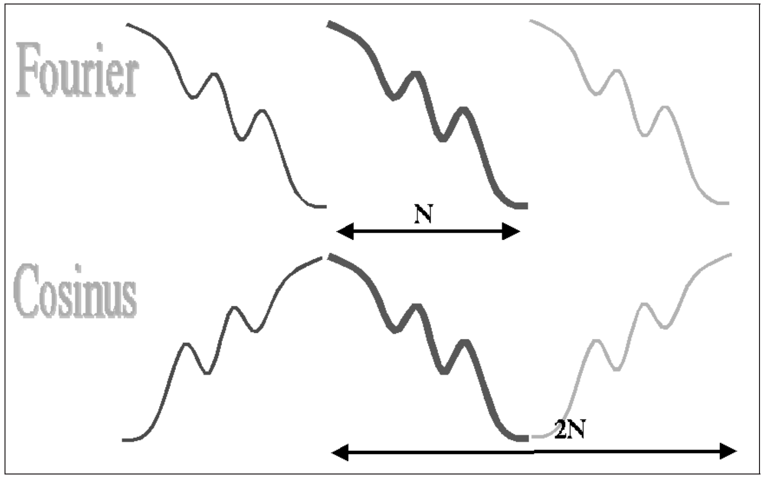
Division de l'image en sous images 8×8



COMPRESSION D'IMAGES
COMPRESSION JPEG (2)

Pourquoi la Transformée de Cosinus Discrète ?

- La TF suppose que l'image à transformer est périodique ▶ crée des discontinuités ▶ hautes fréquences qui ne sont pas présent dans l'image
- la TCD suppose que l'image à transformer est paire ▶ aucune discontinuités



Pour une image de taille $N \times N$

$$C(u,\nu) = k_1(u)k_2(\nu) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\Big(\pi u \frac{x+\frac{1}{2}}{N}\Big) \cos\Big(\pi \nu \frac{y+\frac{1}{2}}{N}\Big)$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{\nu=0}^{N-1} C(u,\nu) \cos\Big(\pi x \frac{u+\frac{1}{2}}{N}\Big) \cos\Big(\pi y \frac{\nu+\frac{1}{2}}{N}\Big)$$

COMPRESSION D'IMAGES
COMPRESSION JPEG (3)

Transformée de Cosinus Discrète et TCD inverse

$$C(u,\nu) = k_1(u)k_2(\nu)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\Big(\pi u\frac{x+\frac{1}{2}}{N}\Big)\cos\Big(\pi\nu\frac{y+\frac{1}{2}}{N}\Big)$$

$$k_1(u) = \begin{cases}\sqrt{\frac{1}{N}} & \text{si } u=0\\ \sqrt{\frac{2}{N}} & \text{sinon}\end{cases} \qquad k_2(\nu) = \begin{cases}\sqrt{\frac{1}{N}} & \text{si } \nu=0\\ \sqrt{\frac{2}{N}} & \text{sinon}\end{cases}$$

Symétrie miroire

$$C(-u,\nu) = C(u,-\nu) = C(-u,-\nu) = C(u,\nu)$$
$$\text{car } \cos(x) = \cos(-x)$$

Périodicité

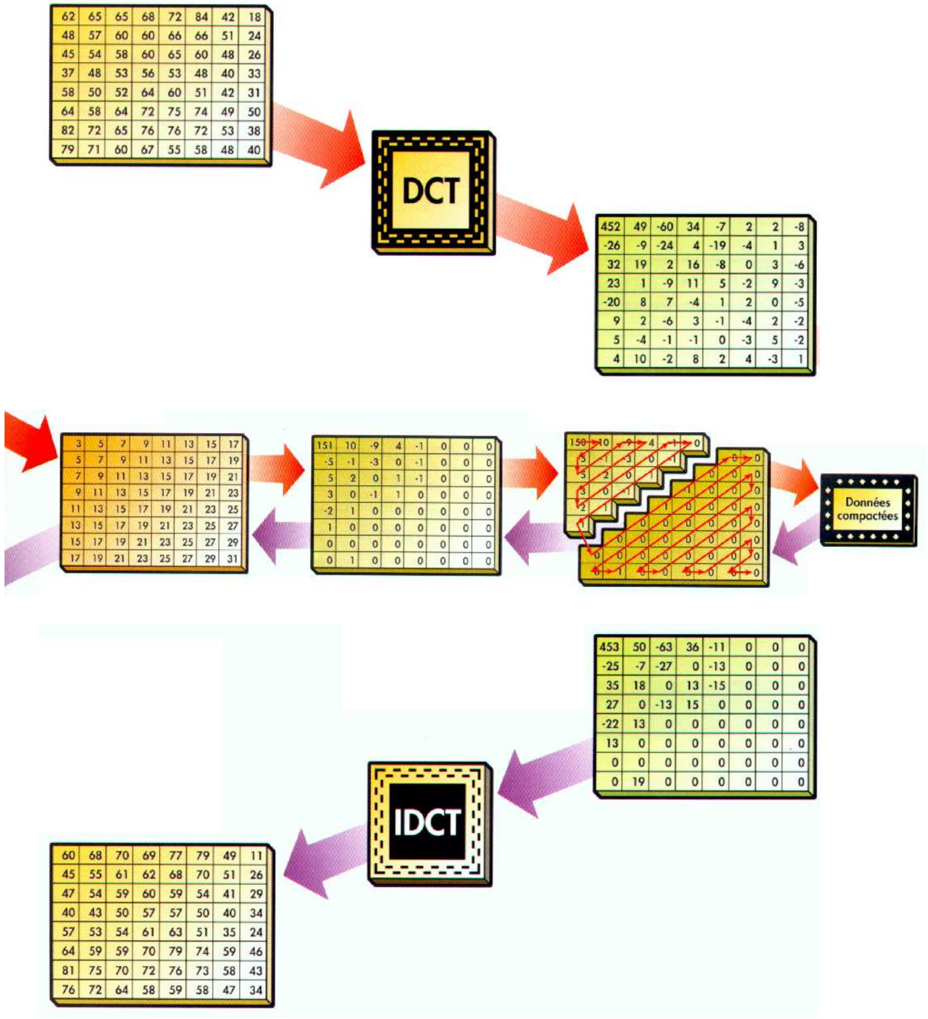
$$C(u+2N,\nu) = C(u,\nu+2N) = C(u+2N,\nu+2N) = C(u,\nu)$$

$$\text{car} \quad \cos\Big(\pi(u+2N)\frac{x+\frac{1}{2}}{N}\Big) = \cos\Big(\pi u\frac{x+\frac{1}{2}}{N} + (2x+1)\pi\Big) = \cos\Big(\pi u(x+\frac{\frac{1}{2}}{N})\Big)$$

Remarque

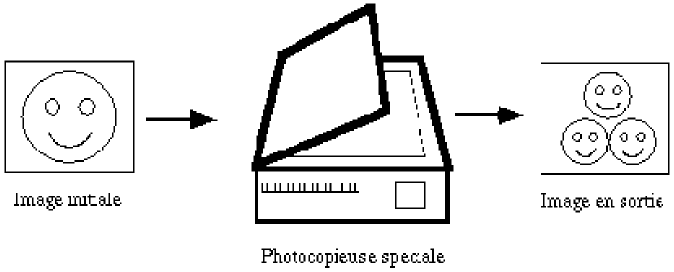
- TCD réelle ▶ Nb. de coefficients moins important que la TF

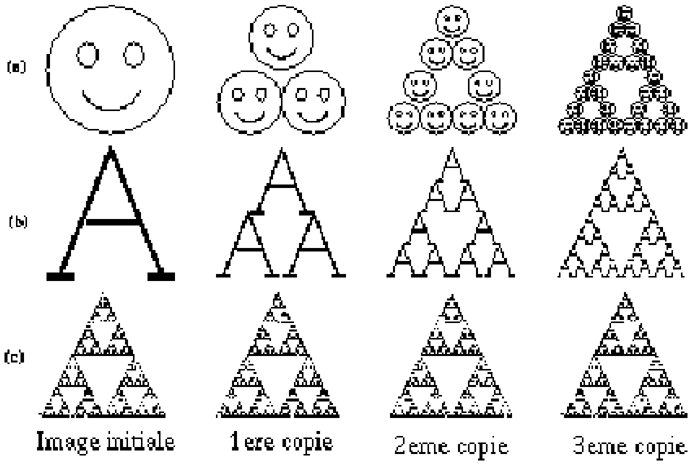
COMPRESSION D'IMAGES
COMPRESSION JPEG (4)



COMPRESSION D'IMAGES
COMPRESSION FRACTALE (1)

Génération d'images fractales





Les trois premières copies générées par la photocopieuse pour différentes images d'entrée

▶ Toute les copies convergent vers la même image finale, que l'on appelle l'attracteur

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (2)

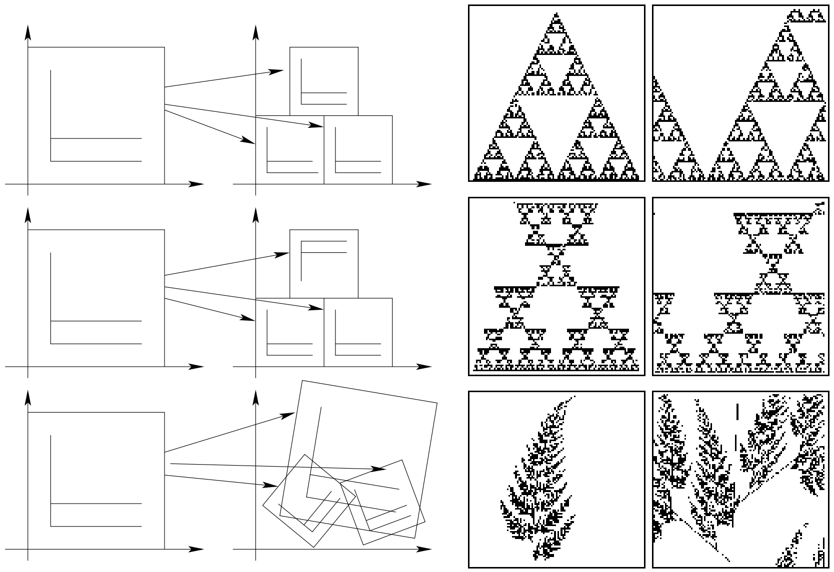
Résultat final ▶ dépend uniquement de la manière dont l'image d'entrée est transformée

▼

Résultat final peut être décrit par un ensemble de transformations affines du type

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

Condition Nécessaire : transformation contractante (i.e., 1 transformation donnée appliquée à 2 points de l'image initiale les rapproche l'un de l'autre dans la copie)



Des transformations, leurs attracteurs et un agrandissement de ceux-ci

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (3)

Chaque image est formée de copies transformées (et réduites) d'elle-même et donc doit avoir des détails à toute les échelles ▶ image fractale

Principe de la compression fractale

Consiste à stocker les paramètres de la transformation donnant l'image finale considérée comme étant un attracteur (M. Barnsley)

Compression fractale d'images réelles

L'image d'un visage n'est pas fractales ou pas exactement auto-similaire, par contre ...

Exemple de régions qui sont similaires à différentes échelles

▼

L'image est formée de copies convenablement transformée de parties d'elle-même

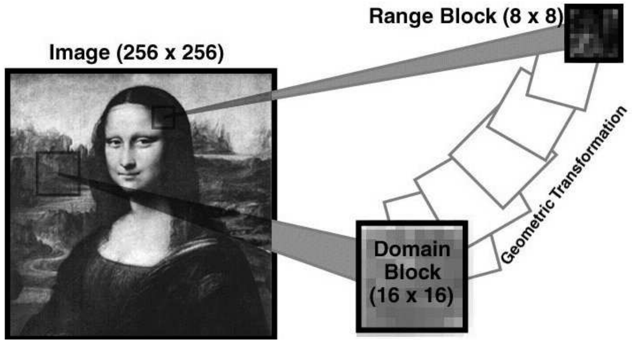
▼

Compression d'une image quelconque : stockage des déformations permettant de générer cette image

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (4)

Algorithme

- Pour chaque sous-image R (8×8 pixels)
  - Pour chaque transformation isométrique
    - Trouver une sous-image D (se recouvrant éventuellement) pouvant se superposer et les paramètres s et o tq. D et $sT(R) + o$ soit le plus proche possible (par ex. au sens des moindres carrés)
- Écrire D, T, s, et o dans le fichier compressé



Transformations isométriques

0^o, 90^o, 180^o, 270^o + flips verticaux ▶ 8 Transformations

Exemple de flips

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (5)

Compression fractale

- Il existe 1024 carrés 8×8 qui sont des sous-parties de l'image ne se recouvrant pas
- On code la position (x, y) de la meilleure correspondance D (2×8 bits)
- On code la transformation T avec 3 bits (8 transformations possibles)
- s : 5 bits, o : 7 bits
- $1024 \times ((2 \times 8) + 3 + 5 + 7) = 1024 \times 31$ bits $= 31744$ bits vs $256 \times 256 \times 8$ bits $= 524288$ bits (16.5 : 1)

Décompression fractale

- On démarre avec une image quelconque
- On prend chaque D (position dans le fichier) et on le transforme (T, S, o) en la sous image R correspondante
- Répéter jusqu'à convergence ▶ attracteur

Note

Compression avec perte

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (6)

Exemple

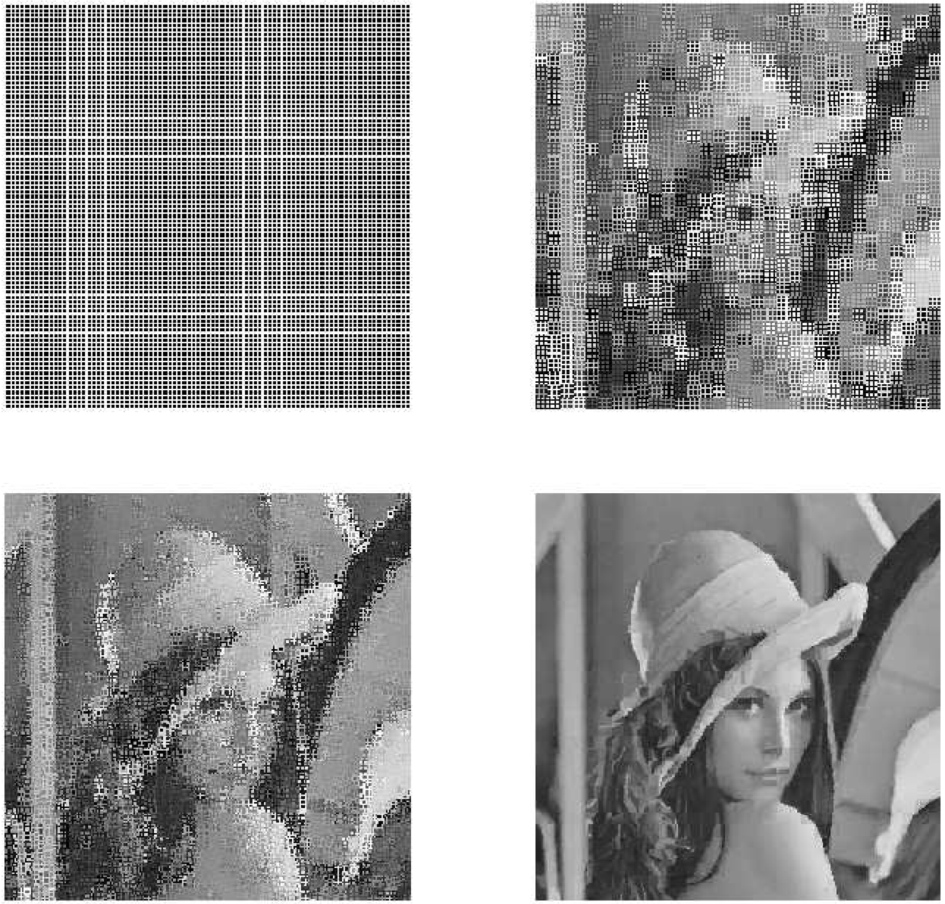


Image originale, première, seconde et dixième itération de la décompression fractale

COMPRESSION D'IMAGES
COMPRESSION FRACTALE (7)

Original

512x512 Boat Image

JPEG

512x512 Boat Image
Compression = 54.3:1 (0.147 bpp)
PSNR = 23.7dB

Fractal Encoding

512x512 Boat Image
Compression = 58.1:1 (0.137 bpp)
PSNR = 27.2dB

Epic Wavelet

512x512 Boat Image
Compression = 58.0:1 (0.138 bpp)
PSNR = 26.4dB

COMPRESSION D'IMAGES
AUTRES FORMAT DE COMPRESSIONS

Format GIF

- Bien adapté au téléchargement
- Image compressée avec l'algorithme LZW (Lempel-Ziv & Welch) sous licence Unisys
- Palette de couleur (et non pas RGB), maximum de 256 couleurs simultanées, généralement suffisant sauf pour les images artistiques

Format PCX (Paintbrush)

- Bien adapté aux dessins
- Image compressée avec l'algorithme RLE (Run Length Encoding)
- Palette de couleur (max de 256 couleurs simultanément)

Format TIFF (Tagged Image File Format)

- Très "ouvert", permet différents formats, différentes options et compressions (différenciable par des "tags") ... ▶ donc pas très standart
- Pixel 24 bits (16.7 millions de couleurs)
- Difficile pour un logiciel de supporter tous les TIFF

Format BMP (BitMap de Microsoft)

- Format brute ("raw") ou binaire
- Image NON compressée
- Palette de 256 couleurs