Image Numérique

Chap III

Les images traités

Types :

- Image Vectorielle : ensemble de formes géométriques (cercle, rectangle, segment, etc ...) qui sont représentées par des formules mathématiques.
- Image Bitmap : image matricielle / pixellisée. Un ensemble de pts appelés pixels sous la forme d'une matrice

Avantages VS inConvenients

Img Vectorielle

(+)

+ souplesse des modifications de l'image (échelle, rotation ...)
+ Intégration aux travaux de schématique : DAO, CAO
+ Moins gourmand en espace mémoire que le bitmap

(-)

+ Complexité d'outils de manipulation
+ Ressources de traitement importante
+ Certains manipulation sont difficiles (le changement de couleur d'une figure)

Img bitmap

(+)

+ Adéquation aux images complexes.
+ Adéquation au traitement d'image
+ Convertir un fichier bitmap d'un format à un autre
+ Utilisé pour tout matériel d'affichage moderne

(-)

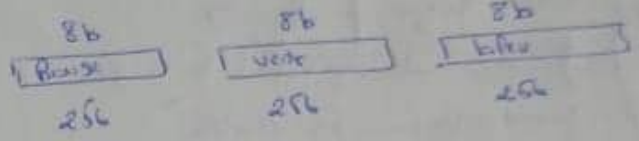
+ Résolution fixe : la taille d'img ne peut pas être modifié sans perdre la qualité.
+ Modification spatiale difficile (changement d'échelle, translation ...)

Scanned with ACE Scanner

Codage de la couleur

+ La valeur stockée dans une case est codé sur certain nombre de bits détermine la couleur de pixel ⇒ Profondeur de codage

1) bitmap (Noir et Blanc) : chaque case 1 bit
2) bitmap 256 sur 8 bits : $2^8 = 256$ couleur différents
3) Image en gris : représente 256 couleurs de gris.
4) Couleur vraies :



Taille d'img en octet ⇒ L x H x (Profondeur codage / 8)

1 bit ⇒ 2 colors
8 bit ⇒ $2^8 = 256$
16 bits ⇒ 2^{16} ....

1 octet = 8 bits.
1 Koctet = 1024 oct = 8192 bits.
1 To = 1024 Kot = 1048576 oct
1 Go = 1024 To = 1048576 Ko.

A apprendre

Comparaison entre eux

(.bmp, .jpeg, ...) bitmap — Vs — vectorielle (.svg, ...)

bitmap	vectorielle
+ taille d'une image est très importante (volumineuse)	+ taille d'une image est faible (peu volumineuse)
+ les transformations géométriques sont dégradantes pour les image bitmap.	+ Les transformations géométriques sont non dégradantes pour l'image vectorielle.
+ Elle permet de représenter les images réelles (photos)	+ Elle permet de représenter les formes « simples » (schémas... logo...)

* Pixel + C'est une abréviation de Picture Element.
  + le plus petit élément constitutif d'une img numérique.

* Image : nombre de pixels constituant l'image / tableau de deux dimension

Taille d'image : largeur x hauteur

XXX Résolution d'une image : établissement entre le nombre de pixels et la taille réelle d'une img.

- Codage de ...
La valeur ...
... déterminé ...
1)
2)

+ + Stockage d'img

opérations de base :
- lecture : mémoire → fichier
- écriture : fichier → mémoire

+ l'image peut stocké :
  + un fichier : forme texte
  + un mémoire : forme vecteur de pt

Les nécessaires :
- nbres lignes / colonnes
- forme de pixels (bit, gris - - -)
- compression éventuelle

2 formats

Simples

fichier texte comportant une entête contenant les dimensions et les pixels d'une img
(PPM .. PGM ..)

(RLG) Join

Compressés (réduire la taille du fichier)

GIF

+ Compression par LZW
+ 256 couleurs au max
+ Animation (plusieurs images dans fichier)
+ n'y a pas de perte d'information

TIFF

Tagged Img File format
+ milliers de couleurs

PCX

+ compressé par RLE
+ Palette de couleur (256)

Scanned with ACE Scanner

La compression d'une image

Déf. réduire la taille physique d'une image enregistrée sur un support de stockage, en essayant de garder une apparence acceptable

2 types

Compression sans perte

image → (compression) → image compressée
image compressée → (décompression) → image

+ Possibilité de reconstruire l'image originale à partir de l'img compressé (décompression)

Compression avec perte

Img → (compression) → Img compressé → (décompression) → Img restitué

Rq : L'image reconstituée n'est pas identique à l'image originale.

Avantages de compression d'une image et aux formats non compressés

+ Gain en espace de stock → disque mémoire.

+ Gain en temps sur les lectures/écritures

- Moins de données pour lire/écrire sur disque
- Compression + écriture plus rapide qu'une écriture des données brute
- Lecture + décompression plus rapide.

+ Gain sur les temps de transmissions.

- Moins de données pour transmettre
- Compression + envoi + décompression } Plus rapide.

Scanned with ACE Scanner

Chapitre : Introduction à la compression des images

- Image numérique (discrète) : une image constituée d'un nombre fini de points (pixels)
- Profondeur : nombre de bit représentant un pixel de l'image.
- Mode de stockage
  - Bitmap : permet de représenter tout type d'image
  - Vectorielle : stockés sous la forme d'une définition mathématique.

Taille fichier = taille d'img × profondeur img

- Compression de données (réduire l'espace à la représentation d'une certaine quantité d'information)
  - destructive : perte de l'information
  - non destructive : pas de perte de l'information (compactage)

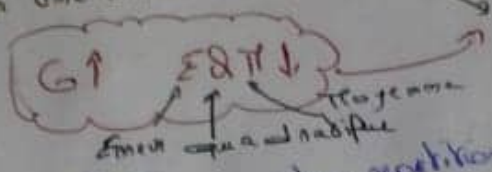
Quotient de compression = $\frac{\text{Taille initiale}}{\text{Taille finale}}$

$\text{Taux} = \frac{1}{(Q)\ \text{Quotient}}$

$\text{Gain} = 1 - \text{Taux} = \frac{TI - TF}{TI}$

Scanned with ACE Scanner

Pour avoir une bonne compression → Un gain de compression maximal
→ erreur quadratique Moyenne minimale



1) RLE : nbre repetition d'apparition une donnée

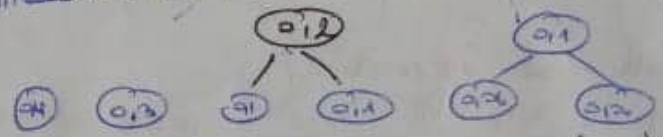
Ex: AABBCCD → A 3B 2C D (sans espaces)

+ Parcours image → Axe x
→ Axe y
Zigzag

2) Le code Huffman :

- Phase 1 : Construction de l'arbre.
  1. classification des symboles
  2. classer les deux faibles probabilités à une supra proba.

  3. Réordonner le table de poids par poids décroissant



  4. Répéter 2/3 pour obtenir un seul arbre.

- Phase 2 : Construction du code à partir de l'arbre obtenu.
  1. من اليمين 1 و من اليسار 0
  2. Lecture du code de haut vers le bas.

3) Méthode par dictionnaire.

1ère passe

1. Listes les symboles (Alphabets) avec fréquence
2. Classement des symbole selon la fréquence.
3. dictionnaire

Scanned with ACE Scanner

2ème passe
1. Lecture symbole par symbole
2. Remplacer chaque symbole par son code binaire
3. Ajout de la longueur du code

⇒ RLE / Code Huffman / méthode par dictionnaire

↳ compression non destructives

* Compression destructives

- Sous échantillonnage -
(ignore certains pixels)
Effet de
⇒ grandes parties d'image sont

- Sous échantillonnage de couleurs -
* Processus qui consiste à réduire la quantité de données d'une image.

JPEG = une méthode de compression

* Chaine de compression :

Transformation → Sous échantillonnage → Découpage en blocs de Pixel → DCT → Quantification → Codage Huffman → Image compressée

compression

décompression

Transformation de couleurs ← Sous échantillonnage ← Assemblage de blocs ← DCT inverse (≠) ← Quantification inverse ← Décodage RLE et décodage

Exemple : Codage Huffman

Exemple de compression d'image au niveau de gris

10	15	15	15	15
10	90	100	100	15
10	90	180	100	15
10	90	180	90	15
10	10	10	10	10

1/ Construire l'arbre de Huffman correspondant.

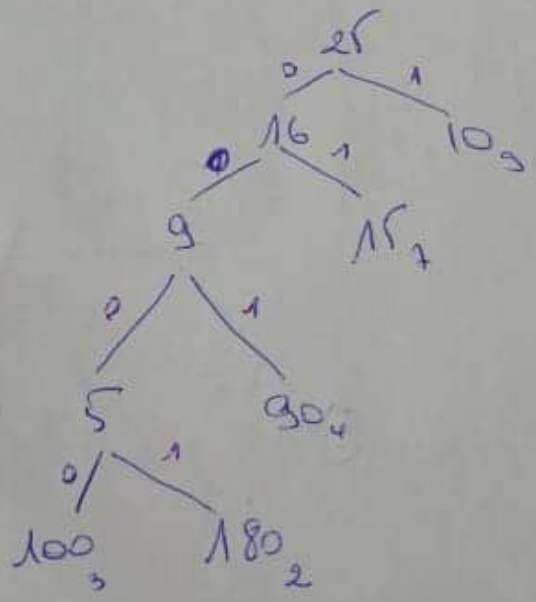
2/ Donner le code binaire obtenu.

*** Réponse ***

1) $10_{(5)} \mid 15_{7} \mid 90_{4} \mid 100_{3} \mid 180_{2}$

nbre de occurrence (fréquence)

	fréq	code bin
10	5	1
15	7	01
90	4	001
100	3	0000
180	2	0001



Scanned with ACE Scanner