

M10: Traitement des éléments multimédia

A.U: 2022/2023

Filière: DWM/S3



Objectifs du cours

- ❖ *Reconnaitre les éléments constitutifs d'une chaîne multimédia.*
- ❖ *Comprendre les principales étapes de la numérisation des données multimédia.*
- ❖ *Maîtriser les concepts de base liés à l'imagerie numérique.*
- ❖ *Traiter et améliorer la qualité des images*
- ❖ *Développer une méthodologie de traitement numérique d'images.*
- ❖ *Définir les notions de base de l'enregistrement sonore.*
- ❖ *Adopter une démarche pour le traitement et montage vidéo.*
- ❖ *Introduire les principaux algorithmes de compression des données multimédia*

Technologies & Logiciels

Logiciels libres et freewares

Manipulation des images

- The GIMP (*multi-plateforme*), ImageMagick (*multi-plateforme*), InkScape (*multi-plateforme*)

Traitement et montage vidéo

- Virtual Dub (Windows), VideoLANMovieCreator(*multi-plateforme*), Avidemux(*multi-plateforme*)

Logiciels professionnels

Traitement des images

- **Adobe Photoshop CS4/CS5** (Windows et Mac OS X), Adobe Illustrator CS4/CS5 (Windows et Mac OS X), PhotoFiltre Studio (*multi-plateforme*).

Traitement et montage vidéo

- **Adobe Première CS4/CS5**..., Final CutPro (Apple)

Langages

- Matlab, Java, C++, **Python**, etc.

Mini-Projet & Liste des exposés

Mini-Projet: Développement d'une application de manipulation et de traitement d'images en Python

Liste des exposés

1. Procédure de création d'une vidéo
2. Acquisition et traitement du son numérique
3. Notion de filmage
4. Montage Vidéo
5. Conception et scénarisation multimédia
6. Monter une animation 2D.
7. Montage vidéo multi-caméras
8. Réalisation d'un site web pour la filière DWM

PLAN

- *Introduction*
- *EM1: Techniques de traitement d'images*
- *EM2: Traitement et montage vidéo*
- *Conclusion*
- *Références*

Introduction

- Qu'est-ce que le multimédia?*
- Technologies matérielles & logicielles*
- Métiers & Applications*
- Projet Multimédia*

Introduction

➊ Qu'est-ce que le multimédia?

- ❖ Un média est un **mode de représentation** de l'information clairement caractérisable
- ❖ Il peut être

Discret (indépendant du temps)

- ✓ Graphique
- ✓ Image fixe

Continu (dépendant du temps)

- ✓ Son
- ✓ Images animées

- ❖ Multimédia = multi + média

Introduction

➊ Qu'est-ce que le multimédia?

Le terme multimédia désigne:

- « un système intégrant divers média et autorisant leur utilisation interactive ». (*The Design Of an Extensible Multimedia Library for an OODBMS*)

Selon la norme AFNOR(Association française de normalisation) le terme MULTIMEDIA peut désigner :

- Un **assemblage** des **technologies destinées** à gérer les données, le son et l'image sur un **même support**.
- **Un ensemble de techniques permettant d'utiliser des informations de type texte, image fixe, image animée et son sur un même support.**
- Le **multimédia** est le **carrefour de plusieurs domaines**:
 - ✓ Audiovisuel, Informatique, Télécommunications, Art graphique, publicité, électronique...

Introduction

Technologies matérielles

✓ **Numérisation**

Son, Vidéo et Image

✓ Accroissement des **capacités de stockage et des débits de transfert**

✓ **Dispositifs** de compression et de décompression

✓ Accroissement **des puissances de calcul**

Technologies logicielles

✓ Représentation de **données complexes** (Hypertexte, réseau sémantiques)

✓ Programmation **orientée objet**

✓ **Synthèse** du son et de l'image

Introduction

Métiers

- ✓ Journaliste
- ✓ Graphiste
- ✓ Webdesigner
- ✓ Animateur 3d
- ✓ Monteur vidéo
- ✓ Infographiste
- ✓ Art numérique
- ✓ Développement
- ✓ Jeux vidéo
- ✓ Métavers ...

Introduction

Applications multimédia

✓ **Applications « grand-public »** (loisir, éducation, art et culture,...)

- Magazine, journaux en ligne
- Musée interactif
- Encyclopédies électroniques
- Livres électroniques
- TV et cinéma interactifs
- Vidéo à la demande
- Réalité virtuelle
- Auto-apprentissage

Introduction

Applications multimédia

✓ **Applications professionnelles** (commerce, promotion, formation)

- Formation à distance
- Vidéoconférences (visioconférence)
- Catalogues interactifs
- Commerce électronique
- Domaine médical (télédiagnostic, dossiers)
- Construction (architecte, simulation)
- Réservation en ligne

Introduction

Projet Multimédia

- ❖ Comme tout projet informatique, **un projet multimédia suit un cycle de vie allant de l'idée jusqu'au développement.**
- ❖ Etapes d'un projet multimédia.

1. Recherche et analyse

- Analyse du public cible pour déterminer leur niveau d'expertise et besoins et identifier le contexte d'utilisation et les contraintes.
- Recherche de contenu à utilisés par la suite.

2. Conception

- Décrire le synopsis du projet.

3. Scénarisation (storyboarding)

- Description détaillée des différents éléments multimédia et de leur timing.

Introduction

Projet Multimédia

4. Prototypage

- Réaliser un premier **prototype** ou **maquette** du produit et le faire **évaluer** par les utilisateurs potentiels.

5. Développement

- Consiste à **créer les éléments multimédia**, les intégrer et développer le reste des modules.

6. Test et validation

- Tester, valider, diffuser le produit puis le maintenir.

PLAN

- *Introduction*
- ***EM1: Techniques de traitement d'images***
- ***EM2: Traitement et montage vidéo***
- *Conclusion*
- *Références*

EM1: Techniques de traitement d'images

- *Concepts et définitions*
- *Chapitre 1: Images Numériques*
- *Chapitre 2: Amélioration des images*
- *Chapitre 3: Restauration d'images*
- *Chapitre 4: Compression d'images*
- *Conclusion*

Concepts et définitions

□ Notion de l'image

- ✓ Une image est **un signal 2D (x,y)**
- ✓ Souvent, cette image représente une **réalité 3D (x,y,z)**

- ❖ Vision **mathématique**:
 - ✓ Une image est une **matrice de nombres représentant un signal**
 - ✓ Plusieurs **outils permettent de manipuler ce signal**

- ❖ Vision **humaine**:
 - ✓ Une image contient plusieurs **informations sémantiques**
 - ✓ Il faut **interpréter le contenu** au-delà de la valeur des nombres

- ❖ Vision **informatique**:
 - ✓ Toute image (**dessin, icône, photographie...**) **acquise, créée, traitée ou stockée sous forme binaire.**

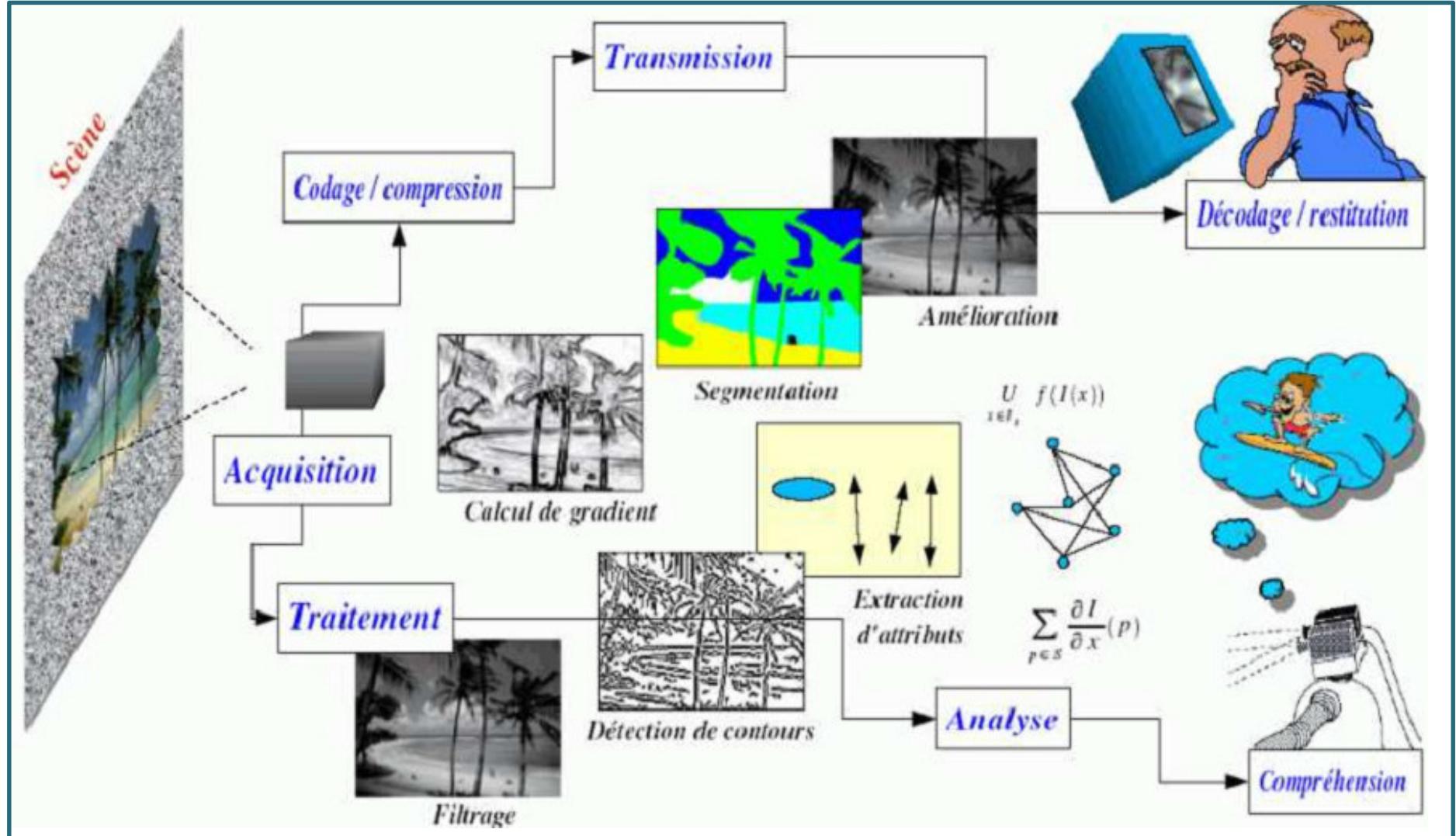
Concepts et définitions

Notion de traitement d'images

Traitement d'images: discipline de l'**informatique** et des **mathématiques appliquées** qui étudie les **images numériques** et leurs **transformations**, dans le but **d'améliorer** leur **qualité** ou d'en **extraire** des information.

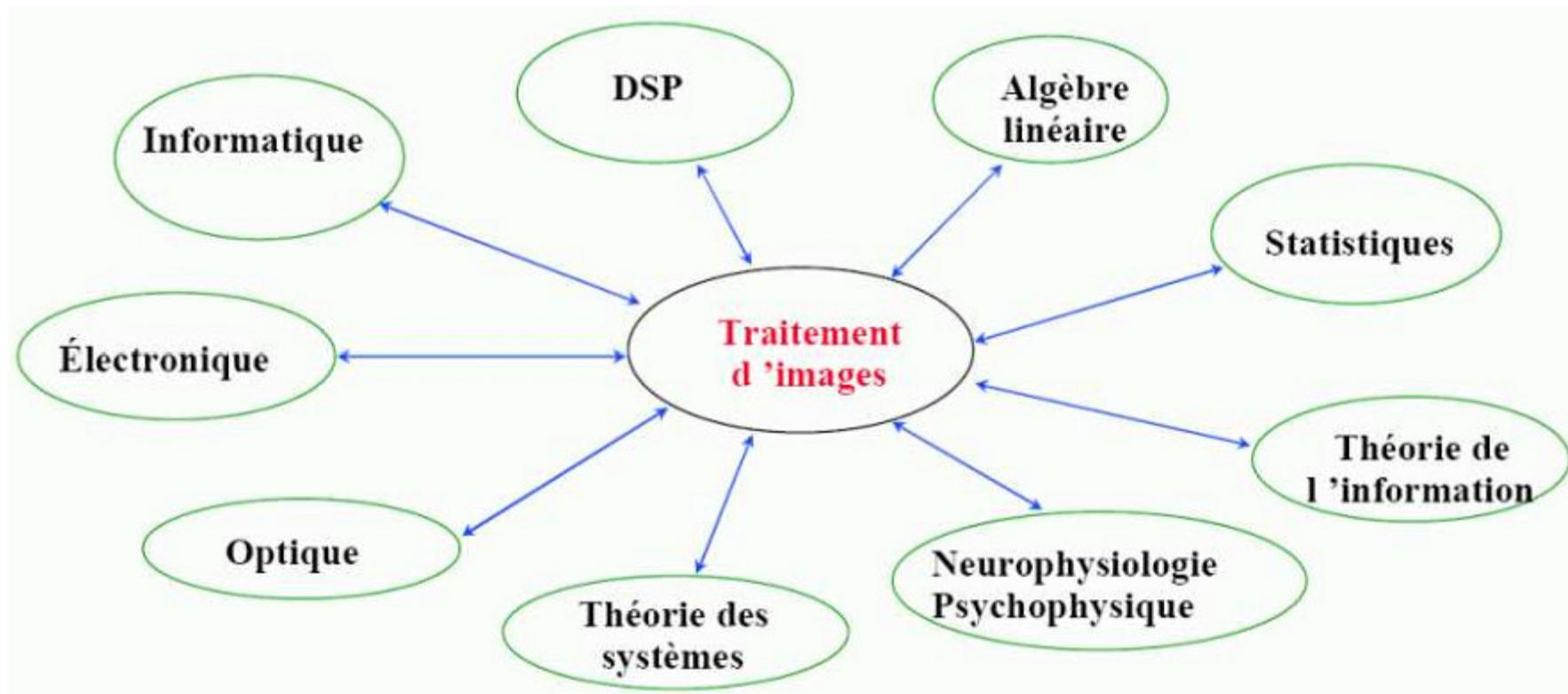
Concepts et définitions

□ Vision globale



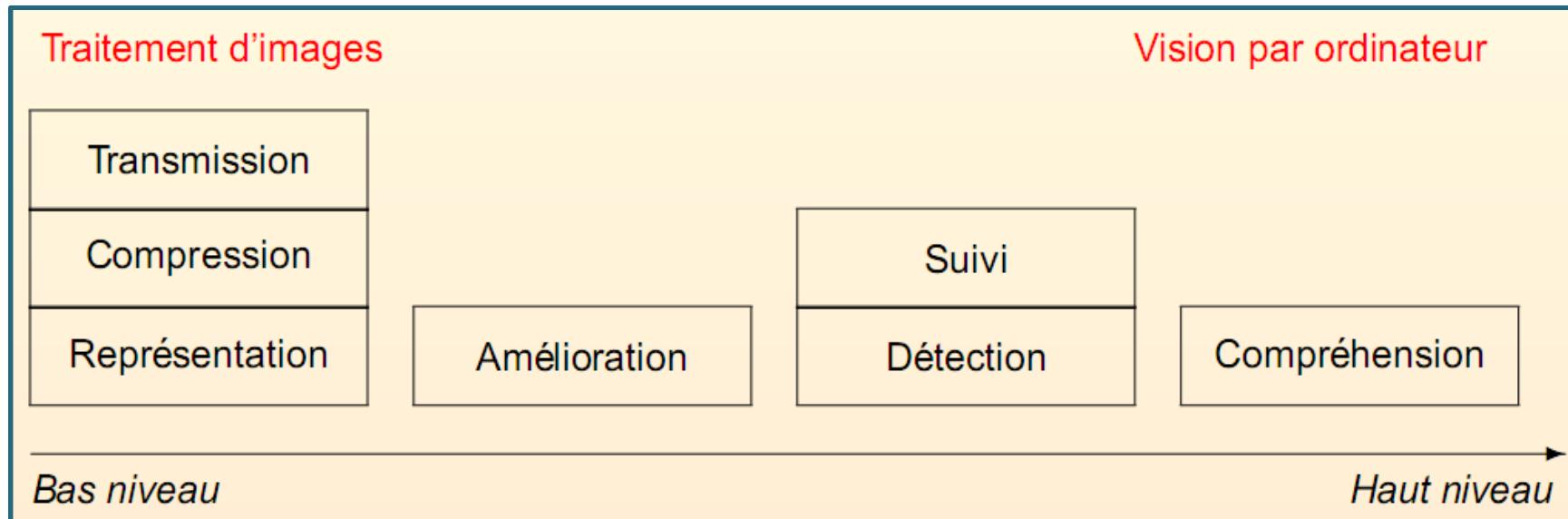
Concepts et définitions

Sciences connectées au traitement d'images



Concepts et définitions

Du traitement d'images à la vision par ordinateur

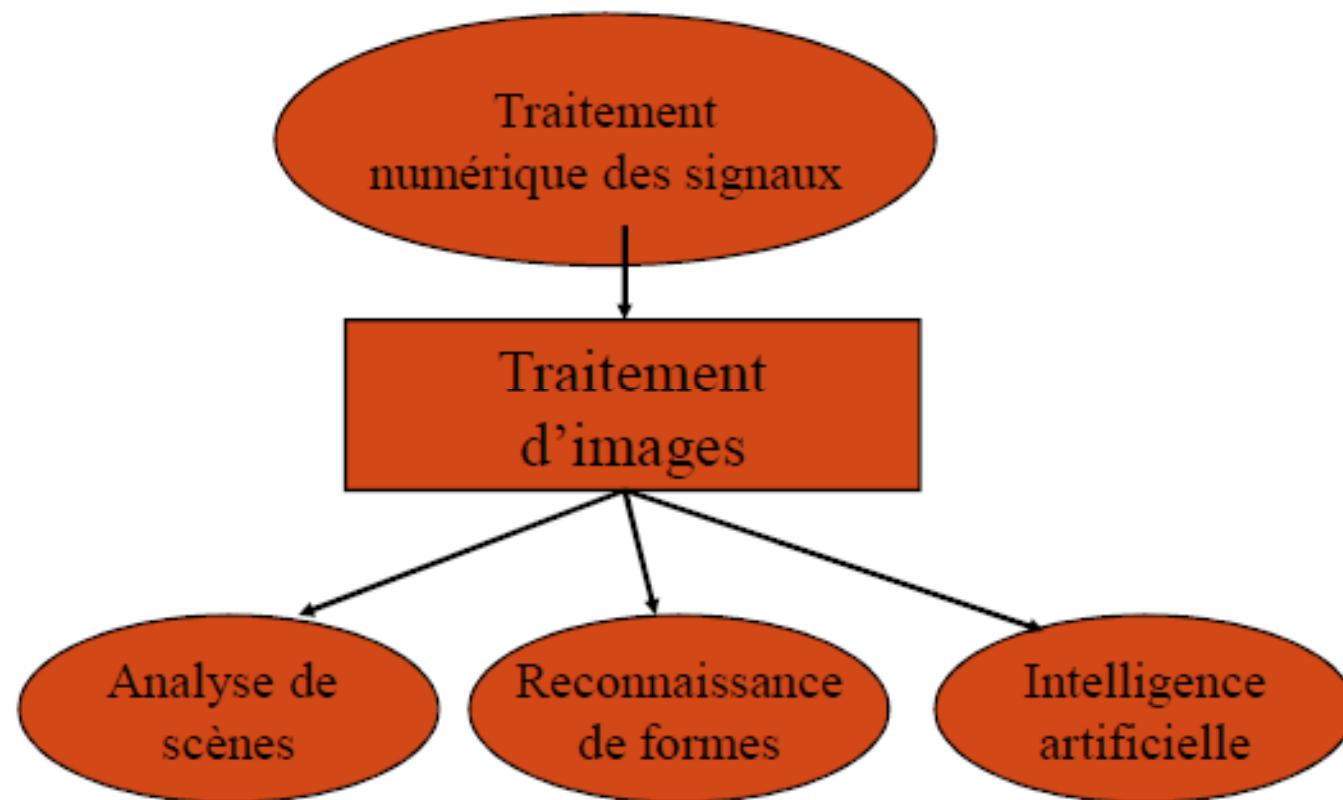


● Pourquoi le traitement d'images ?

- ✓ Le **futur est au multimédia** : les images sont partout!
- ✓ Les **applications sont multiples**

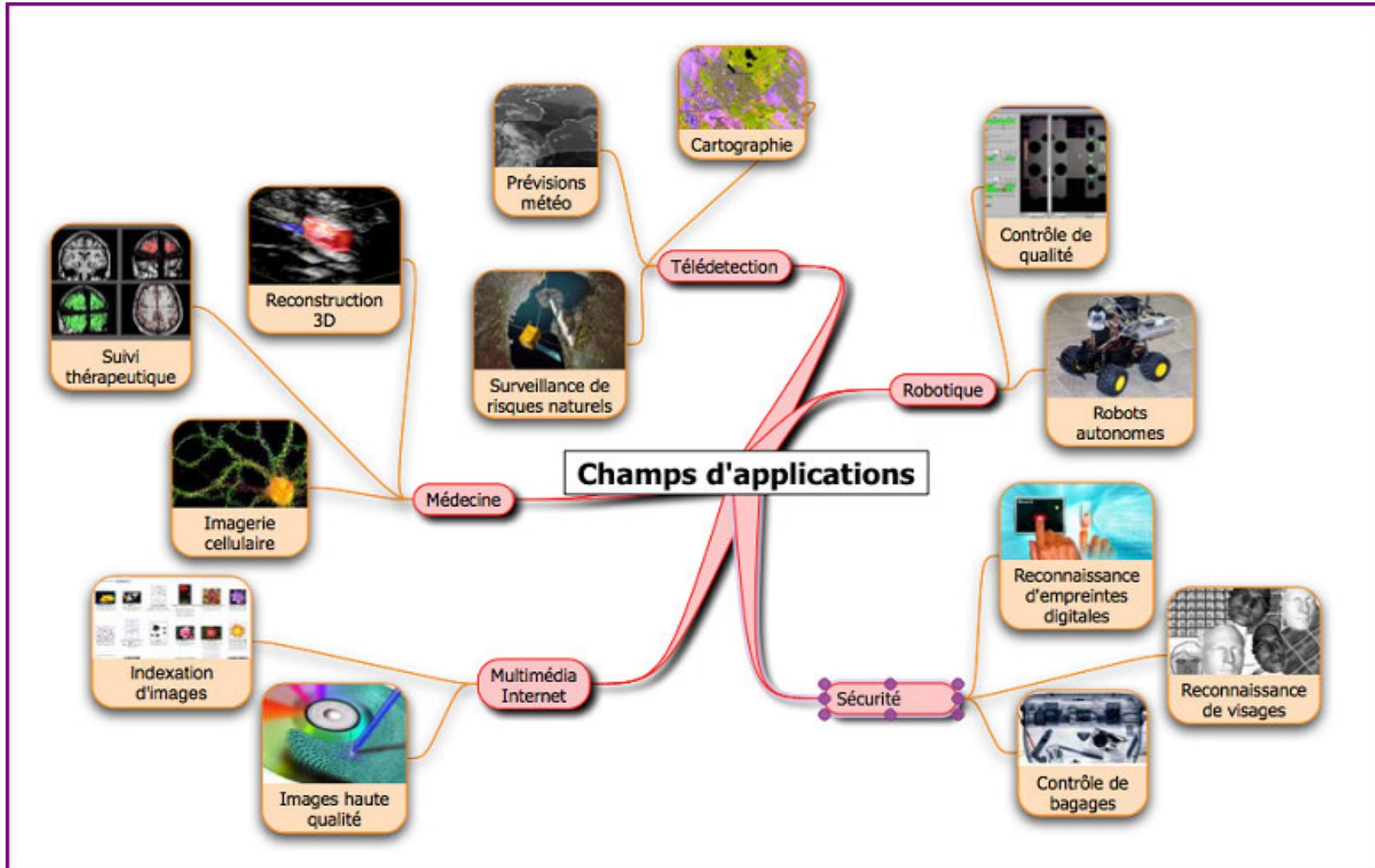
Concepts et définitions

Position du traitement d'images :



Concepts et définitions

□ Champs d'application



→ des milliers d'applications potentielles

Concepts et définitions

Exemples de traitement d'images

- **Amélioration**: augmenter la qualité de la perception visuelle qu'on a d'une image
- **Restauration**: compenser les dégradations (bruit, flou, ...)
- **Compression**: stocker et transférer efficacement
- **Segmentation** : délimiter les “objets”
- **Reconstruction 3D** : obtenir un volume à partir de plans (images)
- **Représentation** : modéliser
 - Bas niveau : texture, couleur, forme, etc.
 - Haut niveau : caractéristiques (features), apprentissage statistiques, graphes
- **Analyse** : convertir en informations
- **Reconnaissance / Compréhension** : identifier le contenu

Techniques de traitement d'images

- *Concepts et définitions*
- *Chapitre 1: Images Numériques*
- *Chapitre 2: Amélioration des images*
- *Chapitre 3: Restauration d'images*
- *Chapitre 4: Compression d'images*
- *Conclusion*

Chapitre 1: Images numériques

- ❖ Types d'images numériques
- ❖ Images matricielles ou bitmap
- ❖ Caractéristiques de l'image bitmap
- ❖ Codage d'image bitmap
- ❖ Images vectorielles
- ❖ Formats d'images

Images numériques

□ Types d'images numériques

Deux types d'images numériques sont distingués:

- Images **matricielles** ou mode point ou **bitmap**
- Images **vectorielles**

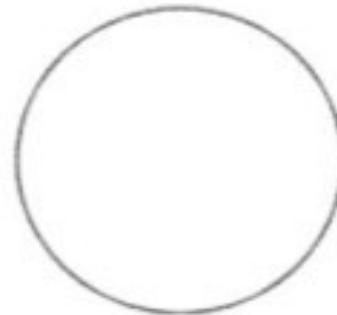


image vectorielle

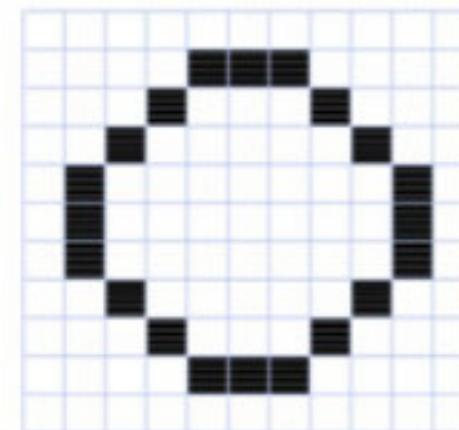
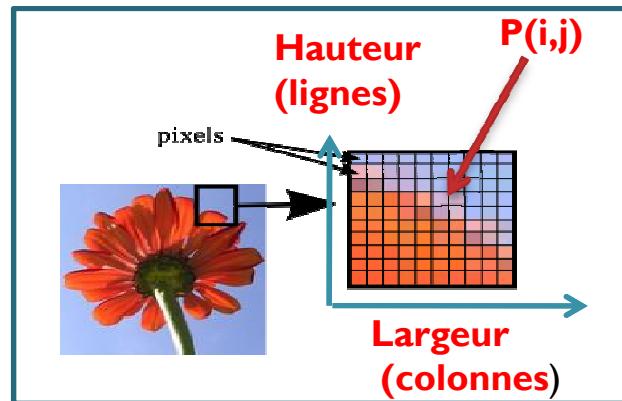


image bitmap

Images numériques

❑ Définition d'image matricielle

- ✓ Image représentée par un **tableau de pixels en 2 dimensions (matrice de pixels)**



- ✓ Le **pixel** (abréviation venant de l'anglais : picture element) est **l'élément de base** d'une image, c'est-à-dire un **point**. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image.

Remarque:

- ✓ Comme pour une matrice, **chaque pixel est spécifié par ses « coordonnées » dans l'image matricielle.**

Images numériques

□ Caractéristiques d'un pixel

- ✓ Ses **coordonnées** (dans la matrice / tableau 2D).
- ✓ Sa **forme** (carrée / rectangulaire).
- ✓ Sa **taille** (largeur × hauteur).
- ✓ Sa **structure** (couleur, profondeur).

Images numériques

□ Caractéristiques de l'image matricielle(bitmap)

Une image bitmap possède:

- **Une définition:** le nombre de points (pixels) constituant une image: C'est la multiplication de **nombre de colonnes** et **le nombre de lignes** de l'image.
- ✓ Une image possédant 10 colonnes et 11 lignes aura une définition de 10×11 .

	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										

- ✓ **Nombre total des pixels:** C'est le produit $L \times H$ qui donne le nombre total de pixels. Il est généralement exprimé en **Megapixels** (million de pixels)

- Par exemple, le standard des appareils photo numériques font des photos ayant comme définition 4000x3000 pixels environ soit un nombre de: **12.10^6 pixels = 12 MPx.**

Images numériques

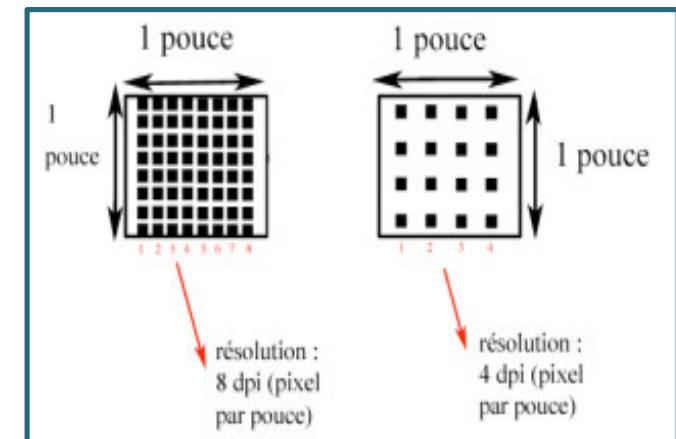
□ Caractéristiques de l'image matricielle(bitmap)

Une image bitmap possède également:

➤ **Une résolution:** Il s'agit du nombre de pixels par unité de longueur

- ✓ L'unité de longueur étant souvent le **pouce** (1 pouce = 2,54 cm),
- ✓ L'unité de la résolution est le **point par pouce (ppp)** (ou ppi en anglais)
- ✓ La **résolution** permet ainsi d'établir le rapport entre la **définition en pixels** d'une image et la **dimension réelle** de sa représentation sur un support physique (affichage écran, impression papier...)
- ✓ La **taille de l'image = Largeur * Hauteur**

$$\text{Largeur} = \frac{m \text{ colonnes}}{\text{Résolution}}$$
$$\text{hauteur} = \frac{n \text{ lignes}}{\text{Résolution}}$$



Images numériques

□ Caractéristiques de l'image matricielle(bitmap)

Exemple

L'image suivante possède une définition de **1900x1174** et une résolution de **72 ppp**. La taille de l'image est :

$$\text{Largeur} = \frac{1900}{72} = 26,4 \text{ pouces} = 67 \text{ cm}$$

$$\text{Hauteur} = \frac{1174}{72} = 16,3 \text{ pouces} = 41,4 \text{ cm}$$

La même image avec une définition différente :



$40 \times 25 = 1000$ pixels



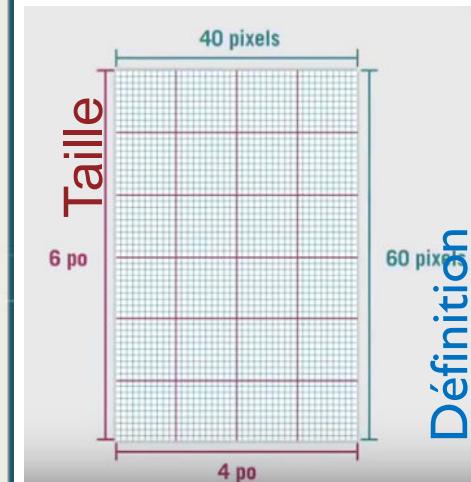
$160 \times 100 = 16000$ pixels



$80 \times 50 = 4000$ pixels



$400 \times 250 = 100000$ pixels



Images numériques

□ Caractéristiques de l'image matricielle(bitmap)

Exercice 1

1.Déterminer le nombre de pixel d'une image 800×400 . **Déterminer** en (cm) la largeur et la hauteur de cette image sachant qu'elle présente une résolution de 72 PPP.

2.Calculer la résolution d'une image bitmap carrée de côté 10 cm et de définition 800×800 .

Réponse

I. Une image 800×400 est constituée de 320 000 pixels.

$$\text{Largeur} = \frac{800}{72} \times 2,54 = 28,2\text{cm}$$

$$\text{Hauteur} = \frac{400}{72} \times 2,54 = 14,1\text{cm}$$

2. La résolution d'une image bitmap carrée de côté 10 cm (**3,94 pouces**) et de définition 800×800 est de:

$$\text{Résolution} = \frac{800}{3,94} = 204 \text{ dpi}$$

Images numériques

❑ Caractéristiques de l'image matricielle(bitmap)

Enfin une image bitmap possède:

Un codage: Il s'agit du **nombre de couleurs que peut prendre un pixel**

- ✓ Ce codage s'exprime en **bit**.
- ✓ Les codages les plus fréquents sont:
 - **1 bit**
 - **8 bits**
 - **24 bits**
- ✓ Le **poids** d'une image en octet
$$\text{Poids (octet)} = \text{Nombre total de pixels} \times \text{codage couleurs (octet)}$$
- ✓ **La profondeur de bit** (ou **profondeur de couleur**) est définie par le **nombre de bits utilisés pour représenter chaque pixel**

Images numériques

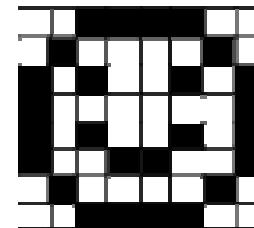
□ Codage d'une image bitmap

Codage 1 bit par pixel: image noir et blanc

- ✓ Chaque pixel peut prendre 2 valeurs de couleurs: **0** ou **1**
- ✓ L'image est en **noir (0)** et **blanc (1)**

1	1	0	0	0	0	1	1
1	0	1	1	1	1	0	1
0	1	0	1	1	0	1	0
0	1	1	1	1	1	1	0
0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0
1	0	1	1	1	1	0	1
1	1	0	0	0	0	1	1

Matrice



image



- ✓ Ex. quel est le poids en (Octet) de l'image d'une définition de 640×480 codée sur 1 bit (noir et blanc) ?

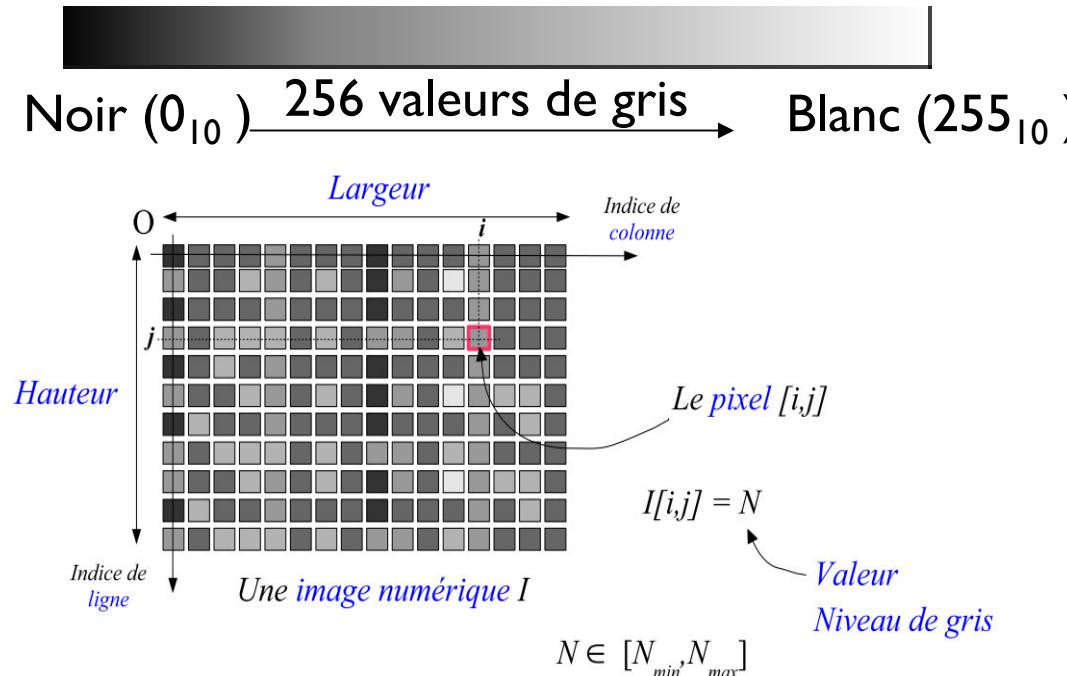
Poids= $640 \times 480 \times 1 \text{ bit} = 307200 \times (1/8) = 38400 \text{ Octet} = 37,5 \text{ Ko}$

Images numériques

□ Codage d'une image bitmap

Codage 8 bits par pixel: image en niveaux de gris

Chaque pixel peut prendre $2^8 = 256$ valeurs de gris



✓ Ex. quel est le poids en (Octet)d'une image d'une définition de 640×480 codée sur 8 bit (niveaux de gris) ?

✓ Poids= $640 \times 480 \times 8$ bit = $307200 \times (8/8)$ = 307200 Octet = 300 Ko

Images numériques

□ Codage d'une image bitmap

Codage 24 bits par pixel en RVB: Image en couleur

- Il existe plusieurs modes de codage des couleurs d'une image numérique, le plus utilisé pour le maniement des images est l'espace colorimétrique Rouge, Vert, Bleu (RVB ou RVG : Red Green Blue) par synthèse additive.
- Le codage de la couleur est réalisé sur 3 octets (24 bits) dont les valeurs codent la couleur dans l'espace RVB.
- Chaque octet représente la valeur d'une composante couleur par un entier de 0 à 255.
- Le nombre de couleurs différentes est de $256 \times 256 \times 256 = 16,8$ Millions.

Images numériques

□ Codage d'une image bitmap

Codage 24 bits par pixel en RVB: Image en couleur

- Une **image numérique RVB** est représentée par **3 tableaux à 2 dimensions (3 plans couleur)** dont la **taille dépend du nombre de pixels contenus dans l'image.**
- Chaque **plan est codé comme une image en niveaux de gris**, avec des valeurs allant de 0 à 255.

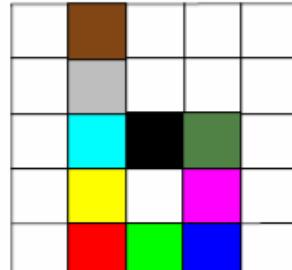
	255 130 255 255 255	255 70 255 255 255	255 20 255 255 255
255 190 255 255 255	255 190 255 255 255	255 190 255 255 255	255 190 255 255 255
255 0 0 80 255	255 255 0 130 255	255 255 0 70 255	255 255 0 255 255
255 255 255 255 255	255 255 255 0 255	255 0 255 0 255	255 0 0 255 255
255 255 0 0 255	255 0 255 0 255	255 0 0 255 255	255 0 0 255 255

Image couleur Niveau de Rouge Niveau de Vert Niveau de Bleu

Représentation matricielle d'une image couleur.

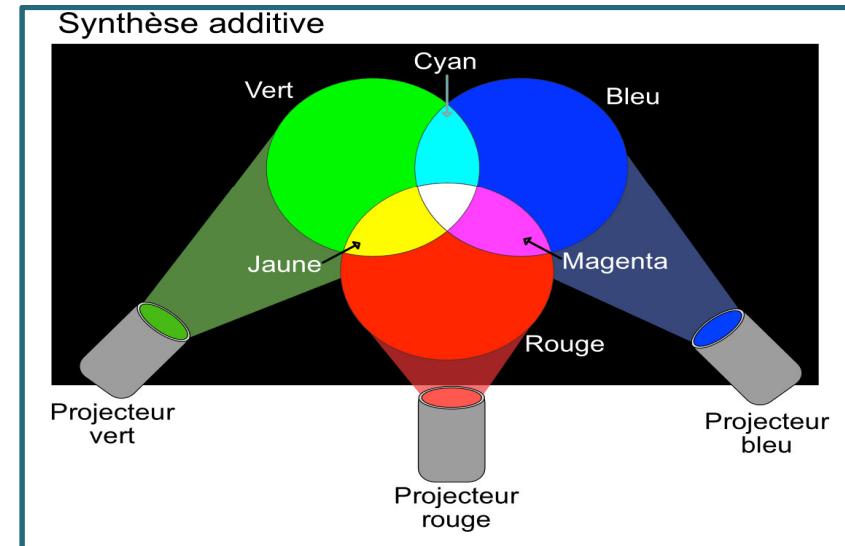
Images numériques

□ Codage d'une image bitmap

Codage 24 bits par pixel en RVB: Image en couleur

- La **couleur du pixel** sera le **résultat de la synthèse additive** des 3 couleurs primaires (**RVB**), chacune étant codée sur 8 bits

R	V	B	Couleur
0	0	0	Noir
0	0	1	Nuance de Noir
0	0	255	Bleu
0	255	0	Vert
255	0	0	Rouge
128	128	128	Gris
255	128	0	Orange
128	0	128	Violet
255	255	255	Blanc



- Pour passer d'une **image couleur** à une **image en niveau de gris**, on utilise la formule suivante:
- Lorsque **R=V=B**, la couleur associé est un **niveau de gris**.

$$G = \frac{R + V + B}{3}$$

Images numériques

□ Codage d'une image bitmap

Les différents systèmes de colorimétrie :

Mode	Nb de bits par pixels	Nombre de couleurs	Remarques
Monochrome ou Noir et Blanc	1	2	Système utilisé pour scanner les textes pour faire de la reconnaissance de texte (OCR)
Niveaux de gris	8	256	Nuance de gris
Mode 4 bits ou 16 couleurs	4	16	Palette de couleurs peu étendue réservée aux dessins simples sans couleurs nuancée
Mode 8 bits ou 256 couleurs	8	256	Palette de 256 couleurs qui permet de conserver une taille raisonnable
Mode 16 bits	16	65536	Palette de 64536 couleurs qui convient pour la plupart des usages
Mode 24 bits ou Couleurs RVB	24	16,7 millions	Mode utilisé par défaut par de nombreux logiciels
Couleurs CMJN	32	4,3 Milliards	4 couleurs primaires : Cyan, Magenta, Jaune et Noir (256 teintes). Utilisé par les imprimantes

Images numériques

Codage d'une image bitmap

Codage 24 bits par pixel en RVB: Image en couleur

Exercice 2

1. Indiquer par combien de bits est codée chacune des 3 couleurs en mode couleurs 24 bits (ou couleurs vraies). Donner la valeur minimale et maximale de chacune des 3 composantes. Déterminer le nombre de nuances de couleurs obtenues avec ce type de codage couleur.
2. Indiquer quelle couleur est obtenue pour une intensité maximale des 3 couleurs RVB?
3. Indiquer quelle couleur est obtenue pour une intensité minimale des 3 couleurs RVB?

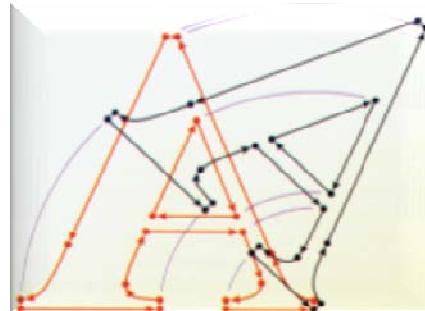
Réponse

1. Chacune des **3 couleurs** est codée sur **8 bits**. Chacune des **3 composantes** présente une valeur comprise entre **0 (Min)** et **255 (Max)**. Avec ce type de codage on obtient $256 \times 256 \times 256 = 16,8$ Millions de couleurs.
2. Les intensités **maximales** des 3 couleurs (RVB) produisent une **lumière blanche**
3. Les intensités **minimales** des 3 couleurs (RVB) produisent une **lumière noire**.

Images numériques

□ Image vectorielle: Définition

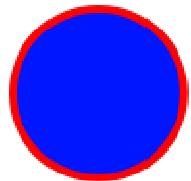
- Image représentée par **des formes géométriques simples** (**arcs de cercle / ellipse, segments de droite, courbes de Bézier...**), auxquelles on peut appliquer différentes transformations : **rotations, écrasement, mise à l'échelle**.



- Chacune des **primitives géométriques** possède quelques attributs (**position, couleur, remplissage**).
- Les **logiciels** permettant de les créer et modifier: **illustrator, Inkscape, Coreldraw,**

Images numériques

□ Image vectorielle: Exemple



En vectoriel, le disque ci-contre n'est pas vu comme un ensemble de points mais comme **un cercle** qui les propriétés suivantes:

- Cercle de couleur **rouge**
- Le cercle est rempli de **bleu**

Le fichier « svg » (créé par Inkscape) correspondant à ce disque est:

```
sodipodi:type="arc" ← L'objet est un arc (fermé donc un cercle)
style="fill:#0000ff;stroke:#ff0000; ← "rempli" de 0000FF: bleu
stroke-width:1;stroke-miterlimit:4; "entouré" de FF0000: rouge: encore ce RVB !
stroke-opacity:1;stroke-dasharray:none; } caractéristiques
stroke-dashoffset:0" de l'objet (couleur du fond,
id="path3839" rempli ou non, fermé ou non)...
sodipodi:cx="222.85715" } coordonnées du
sodipodi:cy="308.07648" } centre
sodipodi:rx="80" } rayon suivant x et y
sodipodi:ry="80"
```

Images numériques

□ Comparaison: image matricielle vs vectorielle

Image matricielle

❖ Avantages :

- ✓ **Aucun calcul** pour l'affichage
- ✓ La **complexité** de l'image n'influe pas sur **sa taille**.

❖ Inconvénients:

- ✓ Peuvent **occuper** beaucoup d'espace mémoire.

Image vectorielle

❖ Avantages :

- ✓ **Peu de place** en mémoire
- ✓ On peut **l'agrandir à l'infini**: la **qualité** n'est pas **modifiée**

❖ Inconvénients:

- ✓ L'image ne doit **pas être trop complexe**

Images numériques

□ Formats des images: Définition

- ✓ Les **images numériques** se retrouvent dans un **format donné**, au même titre que d'autres fichiers informatiques (.docx pour MS Word, .pptx pour MS PowerPoint, .avi pour une vidéo, ...).

- ✓ Une **image numérique** est nécessairement doit être visualisée et utilisée via un **format adéquat**, pour des raisons de poids informatiques notamment.

- ✓ Le **format d'image** correspond à son **mode d'enregistrement** et, le cas échéant, à **son mode de compression**.

Images numériques

□ Formats des images: Formats à privilégier

Nom du format	Type d'image	Points forts	Points faibles	Poids de fichier
JPEG / JPEG 2000 Joint Photographic Experts Group	Matricielle	Excellente compression	Compression Destructrice	Faible
GIF (Graphical Interchange Format)	Matricielle	Possibilité d'animation et de transparence, compression efficace	Limité à 256 couleurs	Faible
PNG (Portable Network Graphics)	Matricielle	Excellente compression sans perte.	Pas très efficace pour les larges photographies	Moyen
BMP (Bitmap)	Matricielle	Format par défaut de Windows	Disponible uniquement sur la plateforme de Microsof	Elevé
TIFF (Tagged Image File Format)	Matricielle	Compression sans perte efficace.	Lourdeur des fichiers non compressés.	Moyen
AI (Adobe Illustrator)	Vectorielle	Reconnu par tous les logiciels graphiques.	Format propriétaire .	Faible
SVG (Scalable Vector Graphics)	Vectorielle	Format XML donc extensible. Très compressible car format texte.	Encore très peu reconnu ,	Faible

Images numériques

❑ Formats des images: Meilleur choix

Critère	Format d'image
Meilleure qualité pour l'archivage	Le TIFF ou le PNG (pas le JPEG à cause de la compression)
Poids de fichier minimum	JPEG
Compatibilité maximum (Windows, Mac, Unix)	Le BMP ou le JPEG
Plus mauvais choix	Le GIF , très limité avec ses 256 couleurs et moins compressé que le JPEG 24 bits !

Images numériques

Exercices

Exercice n°3 :

- I. **Quelle** est la taille en pouces puis en centimètres d'une image numérique dont la résolution est de 200 dpi et les dimensions en pixels sont 2000×3000 ?

10 x 15 pouces ; 25,4 x 38,1 cm

- I. **Calculer** la définition en pixels d'une photographie de 5 pouces sur 7 numérisée à 400 dpi ?

2000 x 2800 pixels

- I. **Donner** la résolution (dpi) d'une image de 8,5 x 11 pouces et qui possède des dimensions en pixels de 2550×3300 ?

300 dpi

4. **Quelle** est la taille en pixels d'une photographie de 20cm x 10cm numérisée à 300 dpi ?
150 dpi ? 100 dpi ? Dans les trois cas, indiquer également la taille d'un pixel. Conclure.

2361 x 1179 pixels ; 1180 x 589 pixels ; 787 x 393 pixels ; 0,085 mm ; 0,17 mm ; 0,254 mm

Images numériques

Exercices

Exercice n°4 :

1. **Calculer** le poids du fichier d'une image de 1000 x 2000 pixels capturée en mode 4 bits (profondeur de bit) ? (en octet puis en Ko)

1000000 octets ; 976 ko

2. **Quelle** est le poids du fichier d'une image de 11 x 8,5 pouces capturée en mode 32 bits (profondeur de bit) à 100 dpi ? (en octet puis en Ko puis en Mo).

3740000 octets ; 3652 ko ; 3,5 Mo

3. **Donner** le poids en Mo du fichier d'une image A4 en RVB à 150 dpi ? à 300dpi ? à 600dpi ?

6,25 Mo ; 25 Mo ; 100 Mo)

Images numériques

Exercices

Exercice °5: Nombre de pixels d'une image

Soit une image de 50,0 cm x 70,0 cm dont la résolution est 400 dpi.

I.Calculer le nombre de pixels par centimètre.

$$\text{Nombre de pixels par centimètre} = \frac{400}{2,54} = 157,480315 \frac{\text{px}}{\text{cm}}$$

2. Calculer le nombre de pixels représentant la hauteur de cette image.

$$\text{Nombre de pixels représentant } L = 157,480315 \times 70,0 = 11\,023,622 \text{ px.}$$

3. Calculer le nombre de pixels représentant la largeur de cette image.

$$\text{Nombre de pixels représentant } l = 157,480315 \times 50,0 = 7\,874,01575 \text{ px.}$$

4. Calculer le nombre total de pixels pour cette image.

$$\text{Nombre total de pixels pour l'image} = 11\,023,622 \times 7\,874,01575 = 86\,800\,173,6 \text{ px.}$$

Images numériques

Exercices

Exercice n°6: Mémoire de stockage :

Pour une image de 10,0 cm x 12,0 cm dont la résolution est 300 dpi :

- 1.Calculer le nombre de pixels par centimètre.
- 2.Calculer le nombre de pixels représentant la hauteur de cette image.
- 3.Calculer le nombre de pixels représentant la largeur de cette image.
- 4.Calculer le nombre total de pixels pour cette image.
- 5.Calculer l'espace mémoire nécessaire pour un codage noir et blanc de cette image.
- 6.Calculer l'espace mémoire nécessaire pour un codage en niveau de gris de cette image.
- 7.Calculer l'espace mémoire nécessaire pour un codage 24 couleurs de cette image.

Images numériques

Conclusion

A retenir pour ce chapitre:

Image numérique
Définition, Types, Caractéristiques



Codage d'image
Noir & Blanc , Niveaux de Gris,
RVB



Formats d'image
Nom , Points forts, Points faibles

Techniques de traitement d'images

- *Concepts et définitions*
- *Chapitre 1: Images Numériques*
- *Chapitre 2: Amélioration des images*
- *Chapitre 3: Restauration d'images*
- *Chapitre 4: Compression d'images*
- *Conclusion*

Chapitre 2: Amélioration des images

- Contraste, luminance et contenu de l'image
- Transformations sur les images
- Manipulation d'histogramme

Amélioration des images

● Contraste, luminance et contenu de l'image

□ Qualité d'image est mesurée par:

- Contraste : **qualité** de la dynamique des **intensités** de l'image
- Bruit : **signal** “parasite” dont la distribution dans l'image est **aléatoire** et la plupart du temps **inconnue**
- Déformations géométriques : **défauts** dus à la **différence d'axe entre le capteur d'acquisition et le centre de la scène observée**

Amélioration des images

- Contraste et luminance
- Luminance ou brillance: **moyenne** des niveaux de gris **de l'image I**

$$\text{Moy} = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x, y)$$

- **Contraste:** plusieurs définitions possibles :

- **Ecart-type** des variations de niveaux de gris :

$$C = \sqrt{\frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (I(x, y) - \text{Moy})^2}$$

- **Variations** entre valeurs de niveaux de gris min et max :

$$C = \frac{\max I(x, y) - \min I(x, y)}{\max I(x, y) + \min I(x, y)}$$

Amélioration des images

● Contenu de l'image

- **Texture**: **répartition statistique ou géométrique** des **intensités** dans **l'image**
- **Contour**: **limite** entre deux (ou un groupe de) pixels dont la **différences** de niveau de gris (couleur) est **significative**.
- **Région**: **groupe de pixels** présentant des **caractéristiques similaires** (intensité, mouvement, etc.)
- **Objet**: **région** (groupe de régions) entièrement **délimitée** par un **contour**, possédant une indépendance dans l'image



Amélioration des images

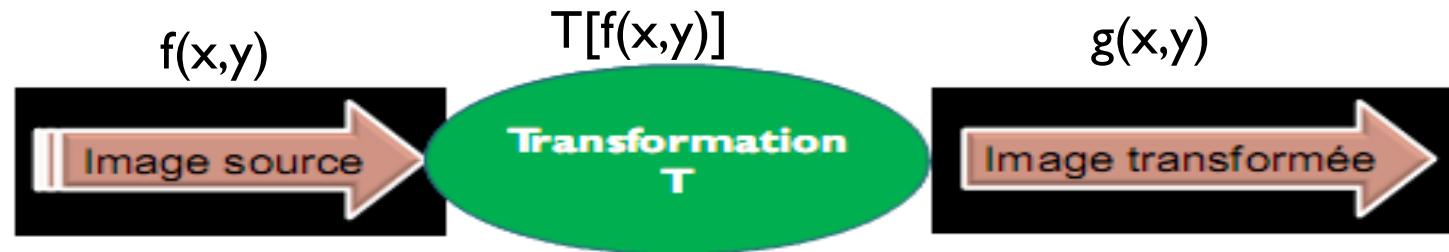
Qu'est-ce que l'amélioration des images

- Opérations d'amélioration: ont pour **but** de **rendre l'image plus lisible**.
- Techniques d'amélioration: cherchent essentiellement à **rehausser les contrastes** afin de rendre les **régions** et les **objets composants l'images bien distinctes et bien séparées les uns des autres**.
- Différentes approches :
 - ✓ **Transformations sur les images**
 - ✓ **Traitements à base d'histogramme**
 - ✓ **Autres méthodes**

Amélioration des images

Transformations sur les images

- Au cours du processus de traitement et d'analyse, l'image subit une **série de transformations.**



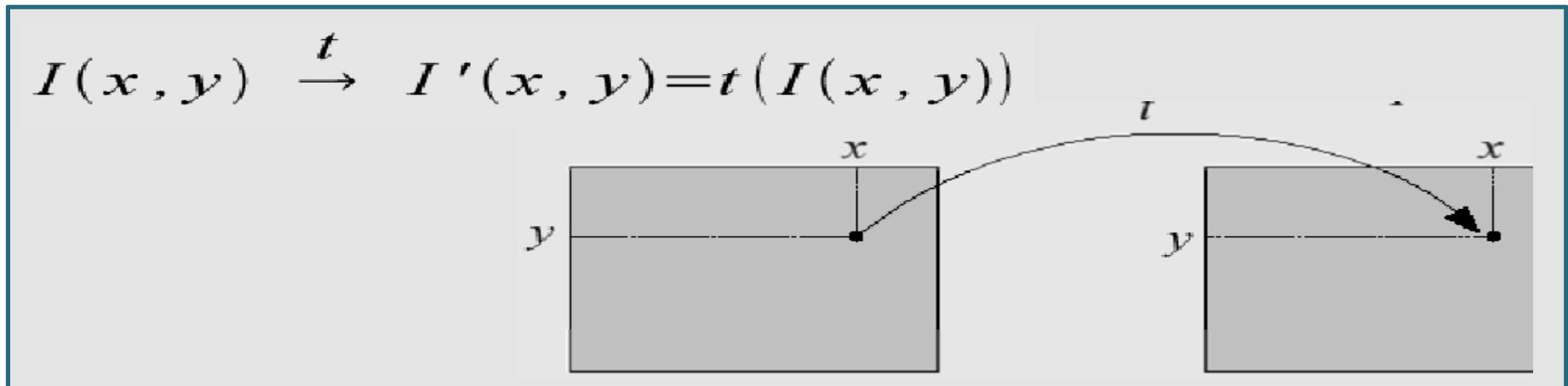
- Une transformation permet de **modifier** la valeur de chaque **pixel** afin d'obtenir une **nouvelle image de même taille** mais ayant des **propriétés plus intéressantes**.
- Les transformations peuvent être classées de la manière suivante:
 - ✓ Transformations **Ponctuelles**.
 - ✓ Transformations **Locales**.
 - ✓ Transformations **Globales**

Amélioration des images

● Transformations sur les images

■ Transformations Ponctuelles:

- ✓ A partir d'une **image source**, la **nouvelle valeur du pixel dépend uniquement de son ancienne valeur.**



- ✓ **Exemples** : correction gamma, inversion, manipulations d'histogramme, ajustement luminosité/contraste, opérations algébriques, opérations logiques et arithmétiques, etc.

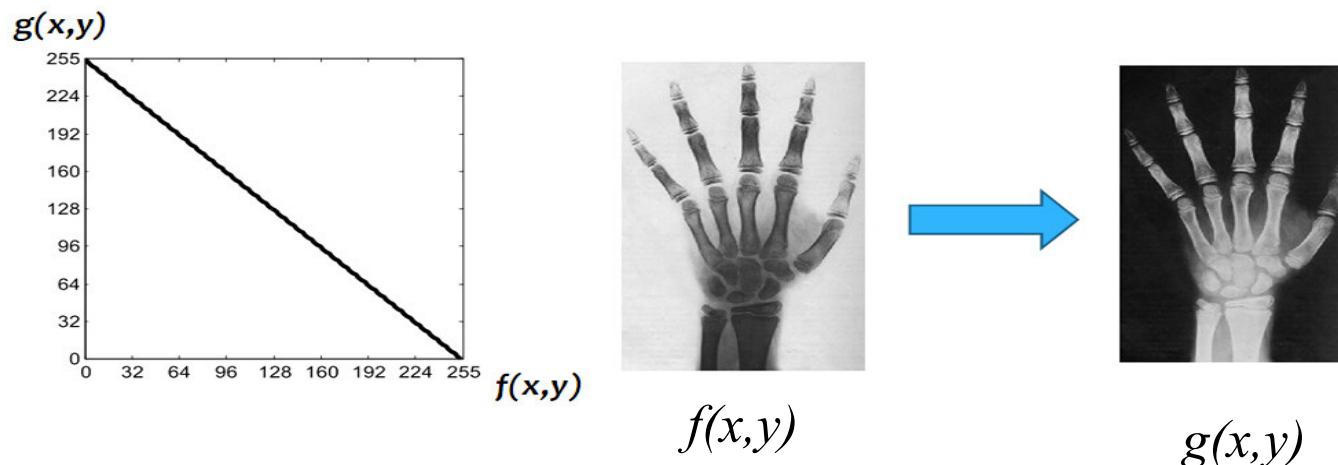
Amélioration des images

Transformations sur les images

a. Transformations Ponctuelles: Inversion Dynamique

- Distingue mieux certains détails en blanc sur fond noir qu'en noir sur fond blanc
- On inverse les extrêmes noir et blanc par la formule suivante:

$$g(x,y) = f_{max} - f(x,y) = 255 - f(x,y)$$



Amélioration des images

Transformations sur les images

b. Transformations Ponctuelles: Correction gamma

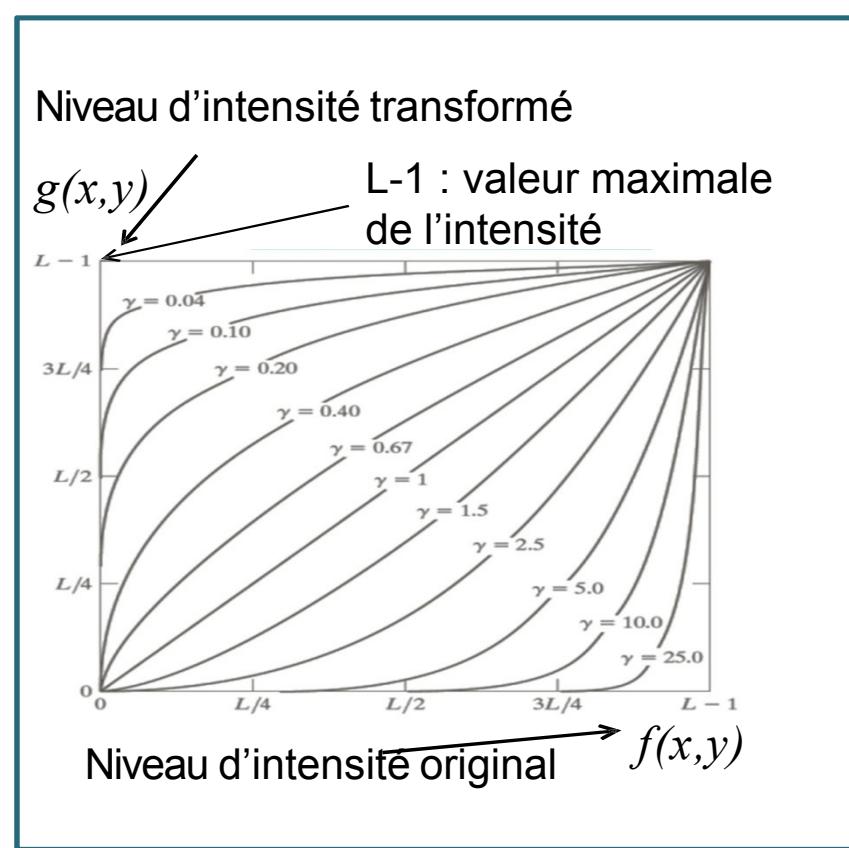
- Applique une transformation **non linéaire** à chaque pixel.
- Le **facteur gamma** va définir la **courbe non linéaire** qui va être appliquée.

$$g(x,y) = \frac{f_{max}}{f(x,y)} f(x,y)^{\gamma}$$

$\gamma < 1$: augmente la plage dynamique des intensités élevées: **dilatation des zones claires**

$\gamma > 1$: augmente la plage dynamique des intensités faibles: **dilatation de zones sombres**

$f_{max} = 255$: pour une Image niveau de gris (8 bits)



Amélioration des images

Transformations sur les images

b. Transformations Ponctuelles: Correction gamma



Image originale

$y = 1.5$



$y = 0.7$



$y = 2$



$y = 0.5$



$y = 0.2$

$y = 3$



Amélioration des images

● Transformations sur les images

c. Transformations Ponctuelles: **Addition d'images**

- L'addition pixel à pixel de deux images f_1 et f_2 est définie par :

$$g(x,y) = \text{Min}(f_1(x,y) + f_2(x,y); 255)$$

- L'addition d'images peut permettre

- de **diminuer le bruit** d'une vue dans une série d'images
- d'**augmenter la luminance** en additionnant une **image avec elle-même**



Amélioration des images

● Transformations sur les images

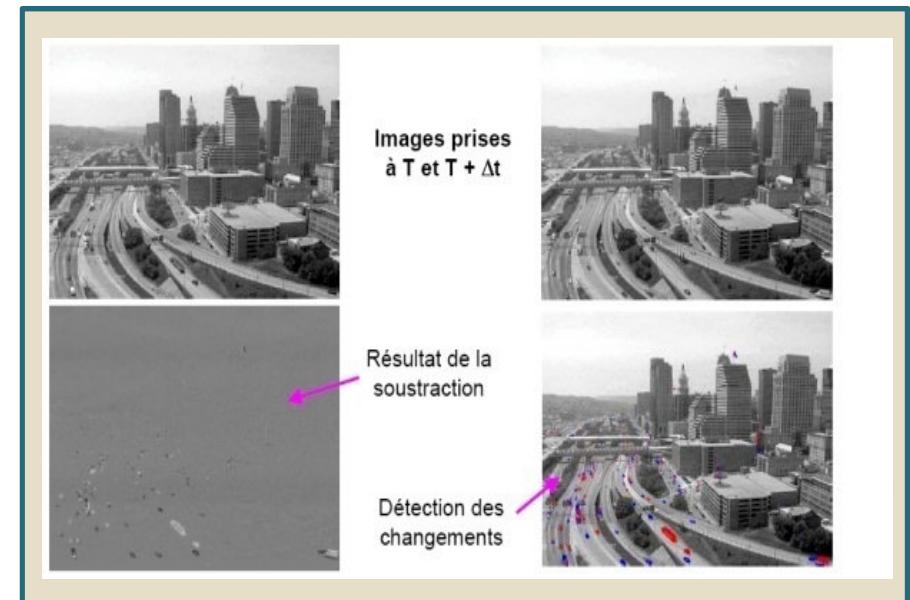
d. Transformations Ponctuelles: Soustraction d'images

- La soustraction pixel à pixel de deux images f_1 et f_2 est définie par

$$g(x,y) = \text{Max}(f_1(x,y) - f_2(x,y); 0)$$

- La soustraction d'images peut permettre

- ✓ la détection de défauts
- ✓ la détection de mouvements



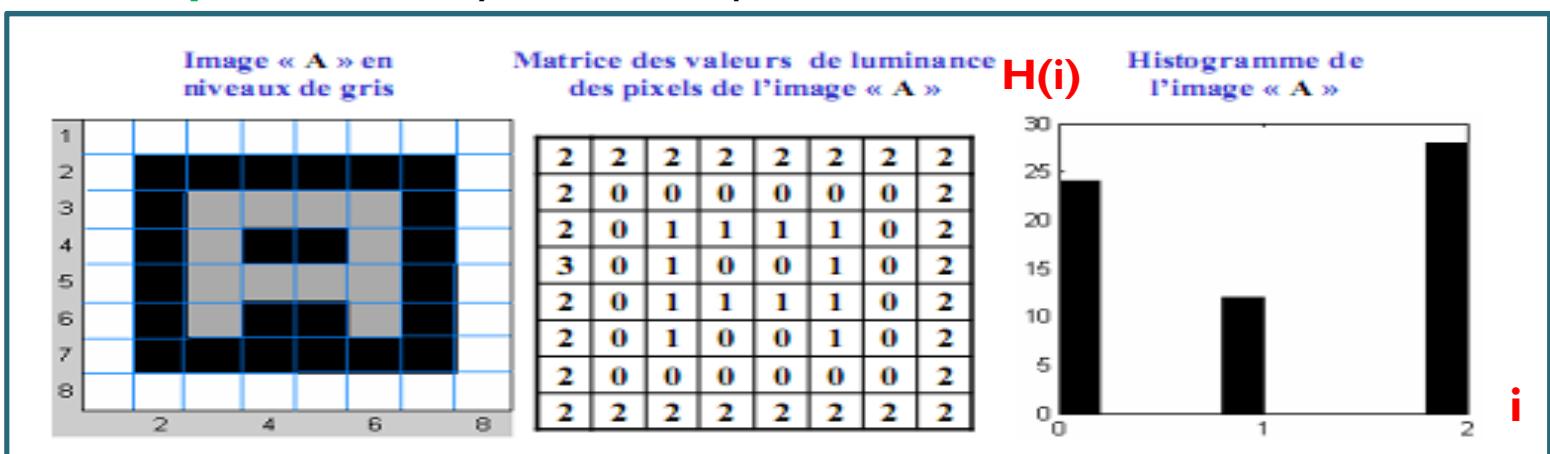
Amélioration des images

Manipulation d'histogramme

a. Définition

- **Histogramme H** : représente la **distribution des intensités des pixels** .
- **H(i)** : le nombre de pixels dans l'image ayant le niveau de gris i
- **Fournit des informations propres à l'image**, telles que :
 - La **distribution statistique** des niveaux de gris
 - Les **bornes de répartition** des niveaux de gris
- Pour **une image couleur**, il est nécessaire d'utiliser **plusieurs histogrammes sur chacune des composantes** du système de représentation de la couleur.

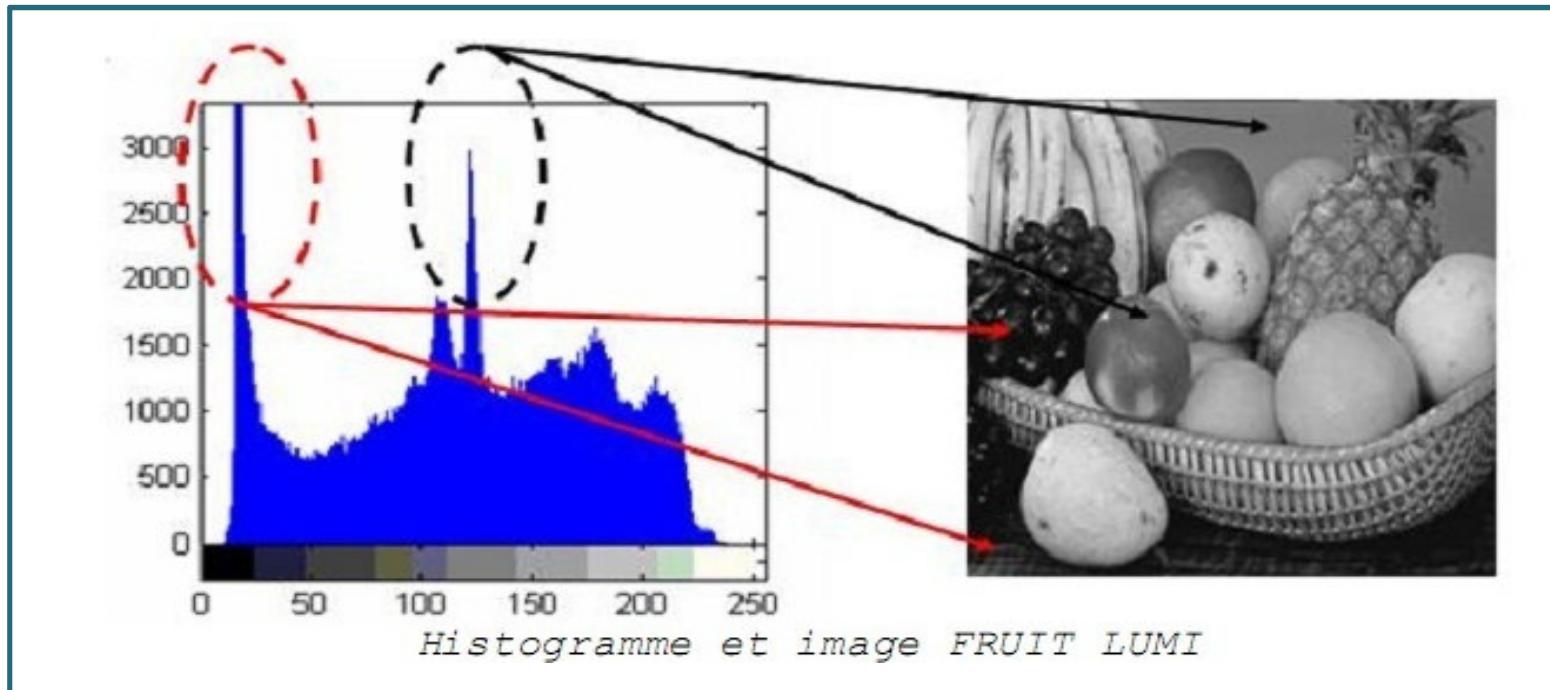
Exemple



Amélioration des images

Manipulation d'histogramme

b. Exemple d'histogramme pour une image en niveaux de gris



Amélioration des images

Manipulation d'histogramme

c. Traitement effectués à partir de l'histogramme

- **Normalisation :**

exploiter toute la **dynamique de codage**.

- **Égalisation:**

équilibrer la dynamique de codage et **augmenter le contraste**.

- **Segmentation:**

simplifier l'image en **regroupant** les **pixels selon leurs valeurs**.

Amélioration des images

Manipulation d'histogramme

d. Histogramme normalisé $H_n(i)$

✓ $H_n(i)$: est le taux de pixels ayant un niveau de gris égal à i

$$H_n(i) = \frac{H(i)}{N}$$

, avec N le nombre de pixels

e. Histogramme cumulé $H_c(i)$

✓ $H_c(i)$: est le **nombre de pixels dont le niveau de gris est inférieur à i**

$$H_c(i) = \sum_{j=0}^i H(j)$$

Il est défini récursivement par

$$\begin{cases} H_c(0) = H(0) \\ H_c(i) = H_c(i-1) + H(i) \end{cases}$$

f. Histogramme cumulé normalisé $H_{cn}(i)$

✓ $H_{cn}(i)$: est le taux de pixels dont le niveau de gris est inférieur à i :

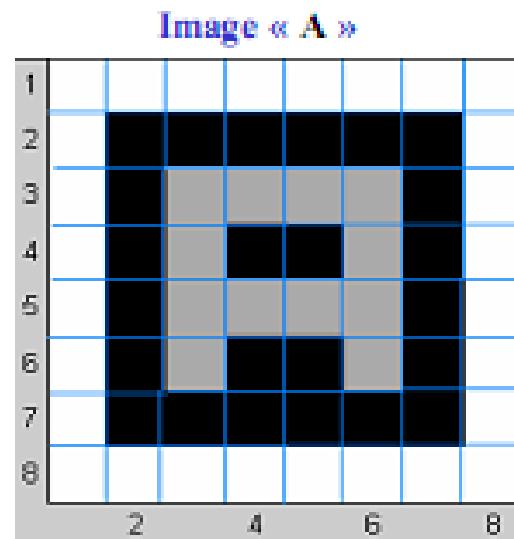
$$H_{cn}(i) = \frac{H_c(i)}{N}$$

Amélioration des images

Manipulation d'histogramme

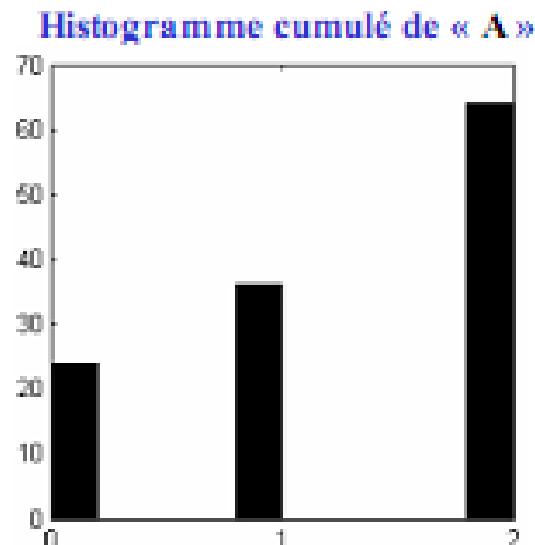
Exemple d' Histogramme cumulé: tracer le H_C de l'image du diapo 66

- Chaque bâton cumule le nombre de pixels du niveaux de gris concerné et des niveaux de gris inférieurs
- les niveaux 0, 1, 2 sont donc représentés respectivement par 24, 36 et 64 pixels.



Valeurs de luminance de « A »

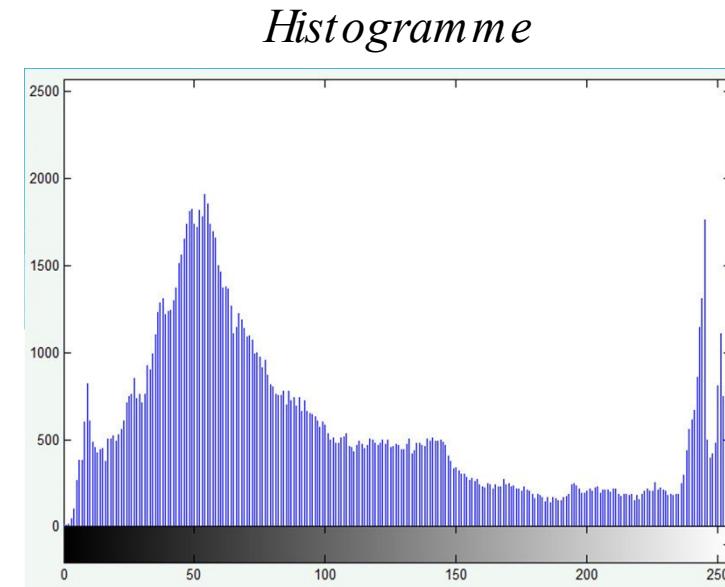
2	2	2	2	2	2	2	2
2	0	0	0	0	0	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	1	1	1	1	0	2
2	0	1	0	0	1	0	2
2	0	0	0	0	0	0	2
2	2	2	2	2	2	2	2



Amélioration des images

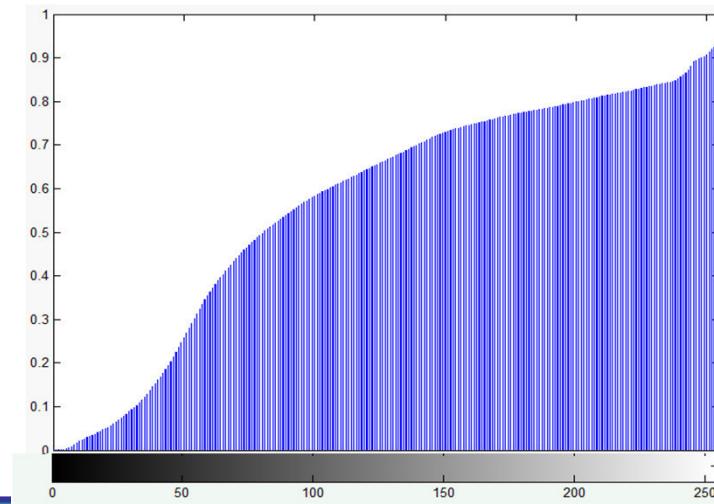
Manipulation d'histogramme

Exemple d' Histogramme cumulé:



Histogramme Cumulé normalisé

L'application la plus connue de l'histogramme cumulé est l'égalisation d'histogramme

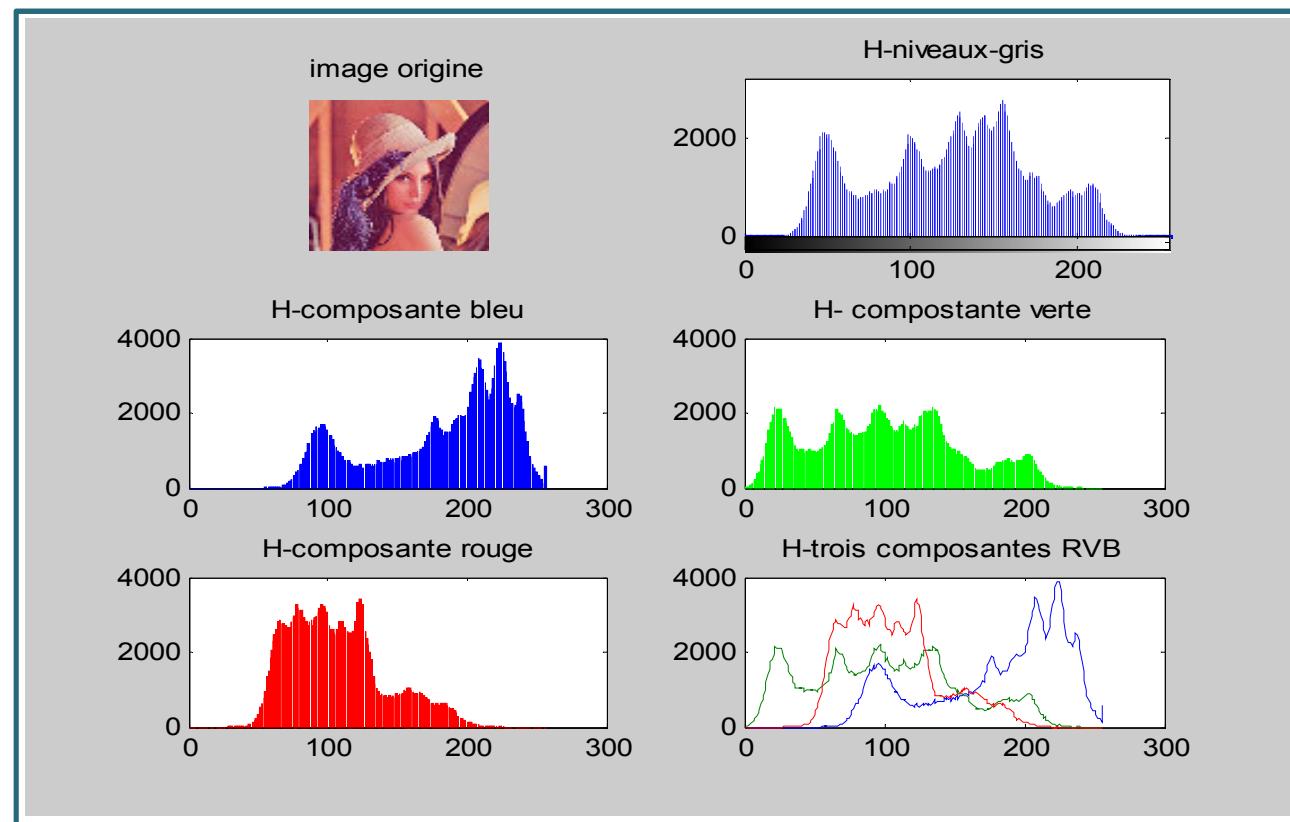


Amélioration des images

Manipulation d'histogramme

Exemple d'Histogramme d'une image en couleur:

- ✓ Plusieurs histogrammes sont nécessaires (selon l'espace colorimétrique).
- ✓ Pour une **image couleur en RVB**; 4 histogrammes sont nécessaires : **distribution de la luminance et distributions respectives des composantes rouge, Verte et bleu**



Amélioration des images

Manipulation d'histogramme

g. Egalisation (ou linéarisation) de l'histogramme

- But : **amélioration du contraste**
- Consiste à **équilibrer le mieux possible la distribution des pixels** dans la dynamique.
- Le niveaux de gris $f(x,y)$ de chaque pixel (x,y) est transformé en $g(x,y)$ en appliquant la transformation suivante :

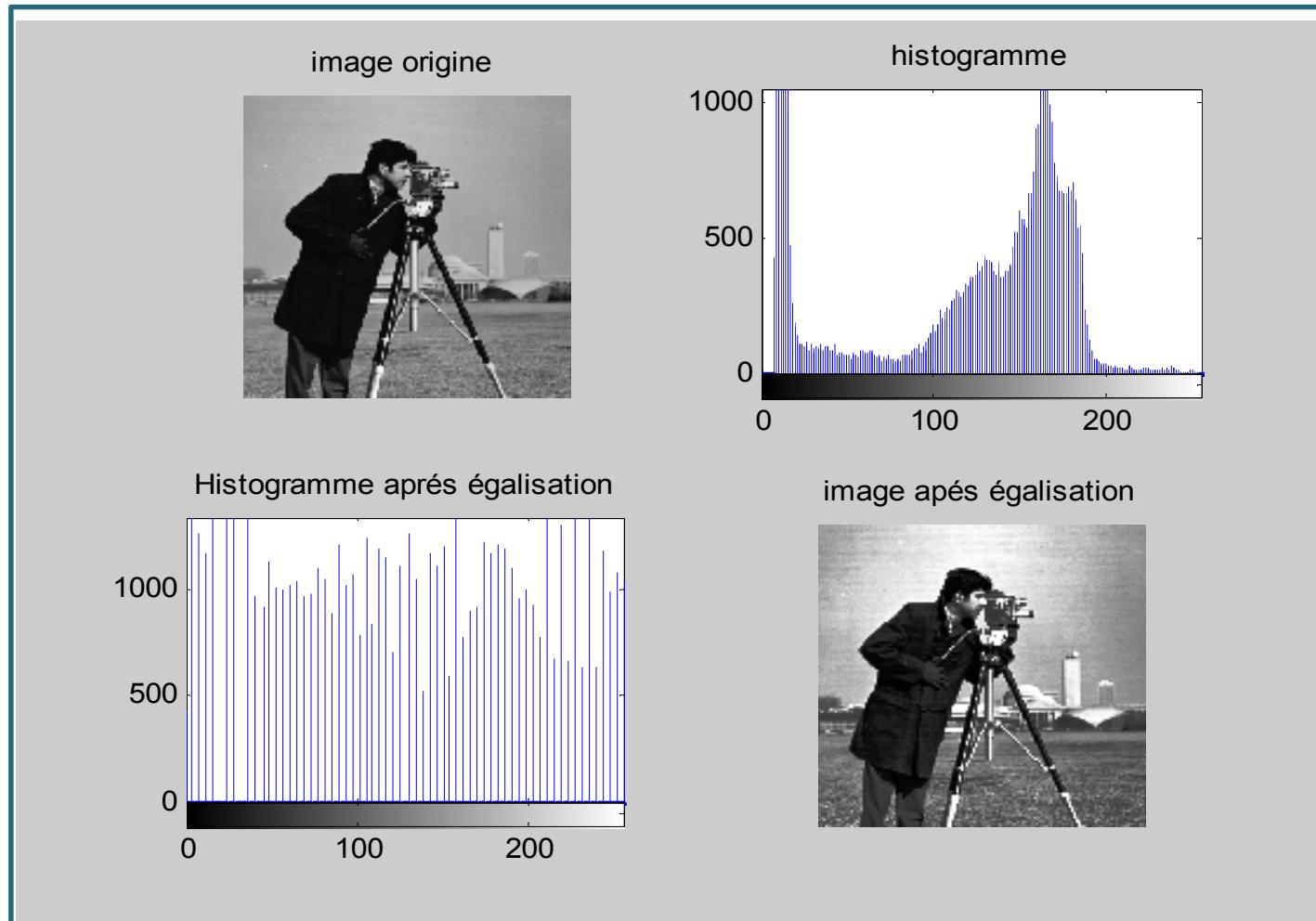
$$g(x,y) = f_{max}(x,y) * H_{cn}(f(x,y))$$

$f_{max} = 255$: pour une Image niveau de gris .

Amélioration des images

Manipulation d'histogramme

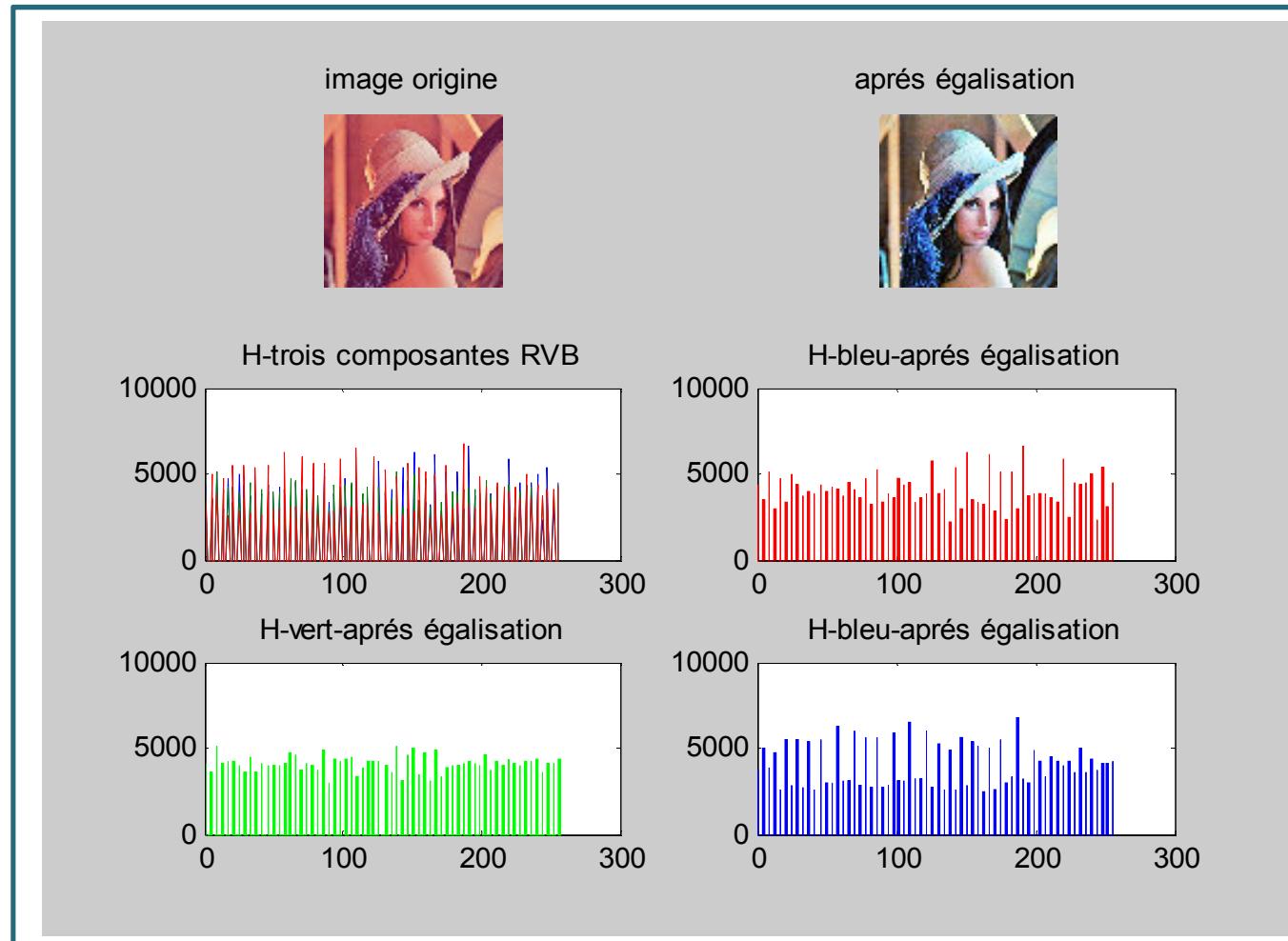
Exemple d'égalisation de l'histogramme d'une image en niveaux de gris:



Amélioration des images

Manipulation d'histogramme

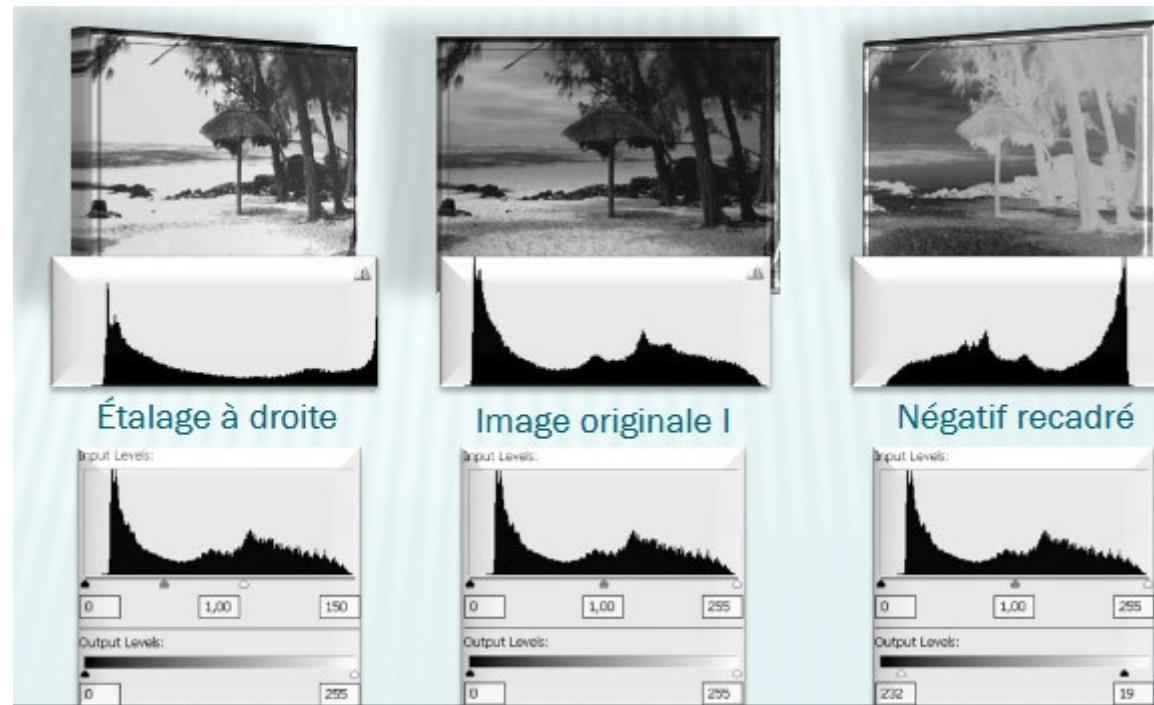
Exemple d'égalisation de l'histogramme d'une image couleur:



Amélioration des images

Manipulation d'histogramme par Photoshop

- Les **courbes**;
 - **Niveaux**;
 - Correction de **Gamma**
- **Niveaux:** Permet d'ajuster la dynamique de l'image

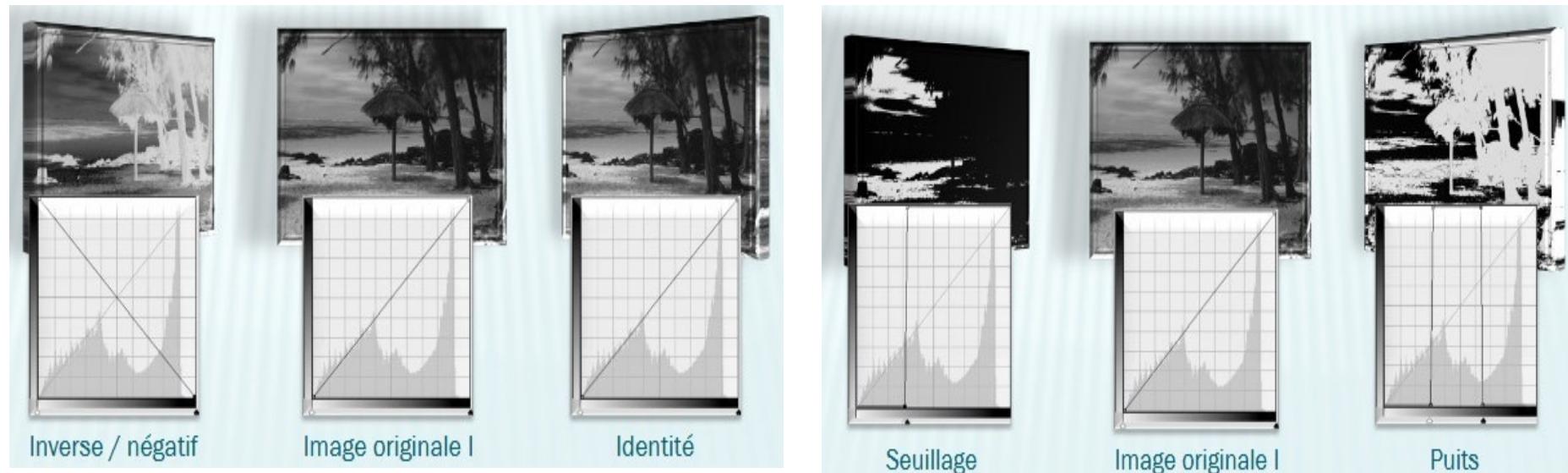


Amélioration des images

Manipulation d'histogramme par Photoshop

➤ Les courbes :

- ✓ Transformation **point à point** de l'intensité suivant la courbe
- ✓ Modification **globale** car affecte toute l'image
- ✓ Modification de **l'histogramme** en conséquent



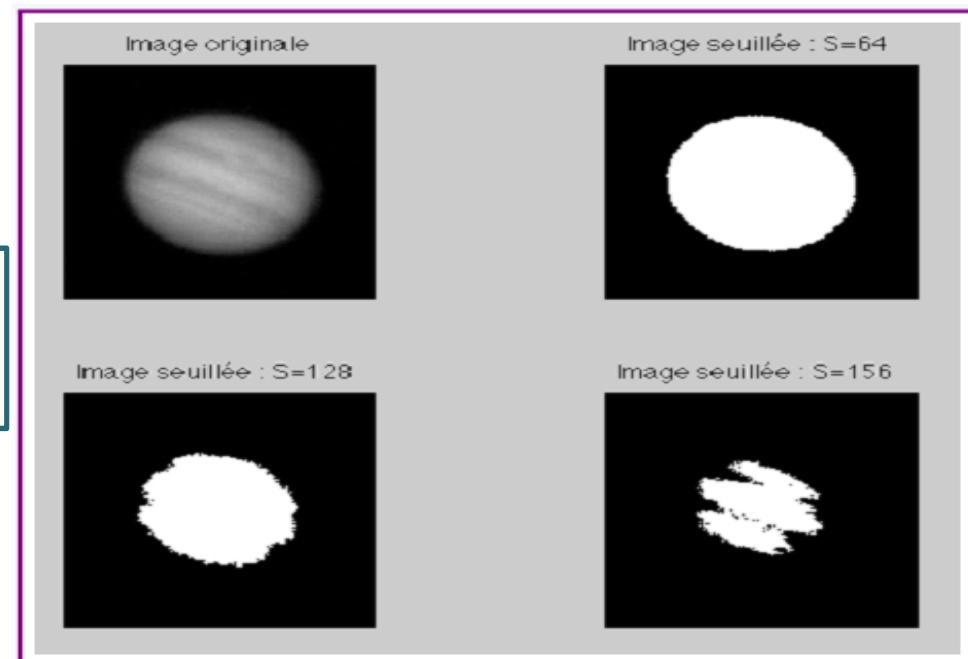
Amélioration des images

Manipulation d'histogramme

h. Seuillage

- **Seuillage (thresholding)** : traitement ramenant l'image à deux ou quelques niveaux d'intensité
- **Binarisation (binarization)** : traitement ramenant l'image à deux Niveaux⇒ seuillage binaire
- Le seuillage binaire est défini par :

$$f_s(x,y) = \begin{cases} 0 & \text{si } f(x,y) < S \text{ avec } S \in [0,255] \\ 255 & \text{sinon} \end{cases}$$

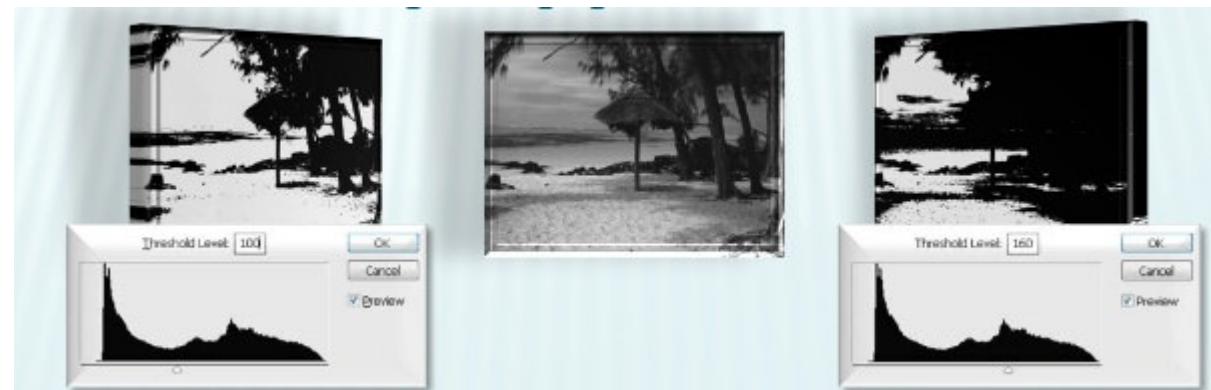


Amélioration des images

Manipulation d'histogramme

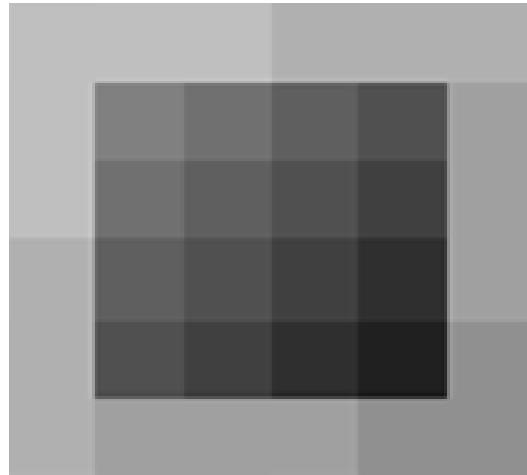
h. Seuillage

- Effet
 - ✓ Toutes les valeurs **d'intensité de gris supérieures à une valeur V** sont annulées (noires)
 - ✓ Sinon, elles sont blanches
- Équivalent photoshop
 - ✓ Image > Réglage > Seuil



Exercices

L'image de la figure ci-dessous est une image en niveaux de gris de définition 6x6 pixels et dont les valeurs des niveaux de gris sont codés sur **4 bits**. Cette image représente une forme rectangulaire sur un fond.



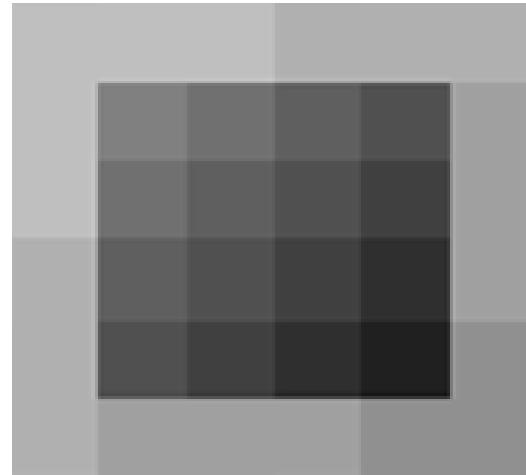
9	9	9	8	8	8
9	5	5	4	3	7
9	5	4	4	3	7
8	5	4	4	3	7
8	5	4	3	2	6
8	8	8	6	6	6

Partie 1:

1. Calculer la moyenne des niveaux de gris de l'image.
2. Calculer le contraste de cette image.
3. Donner l'Inversion dynamique de cette image.
4. Additionner cette image avec elle-même.

Exercices

L'image de la figure ci-dessous est une image en niveaux de gris de définition 6x6 pixels et dont les valeurs des niveaux de gris sont codés sur **4 bits**. Cette image représente une forme rectangulaire sur un fond.



9	9	9	8	8	8
9	5	5	4	3	7
9	5	4	4	3	7
8	5	4	4	3	7
8	5	4	3	2	6
8	8	8	6	6	6

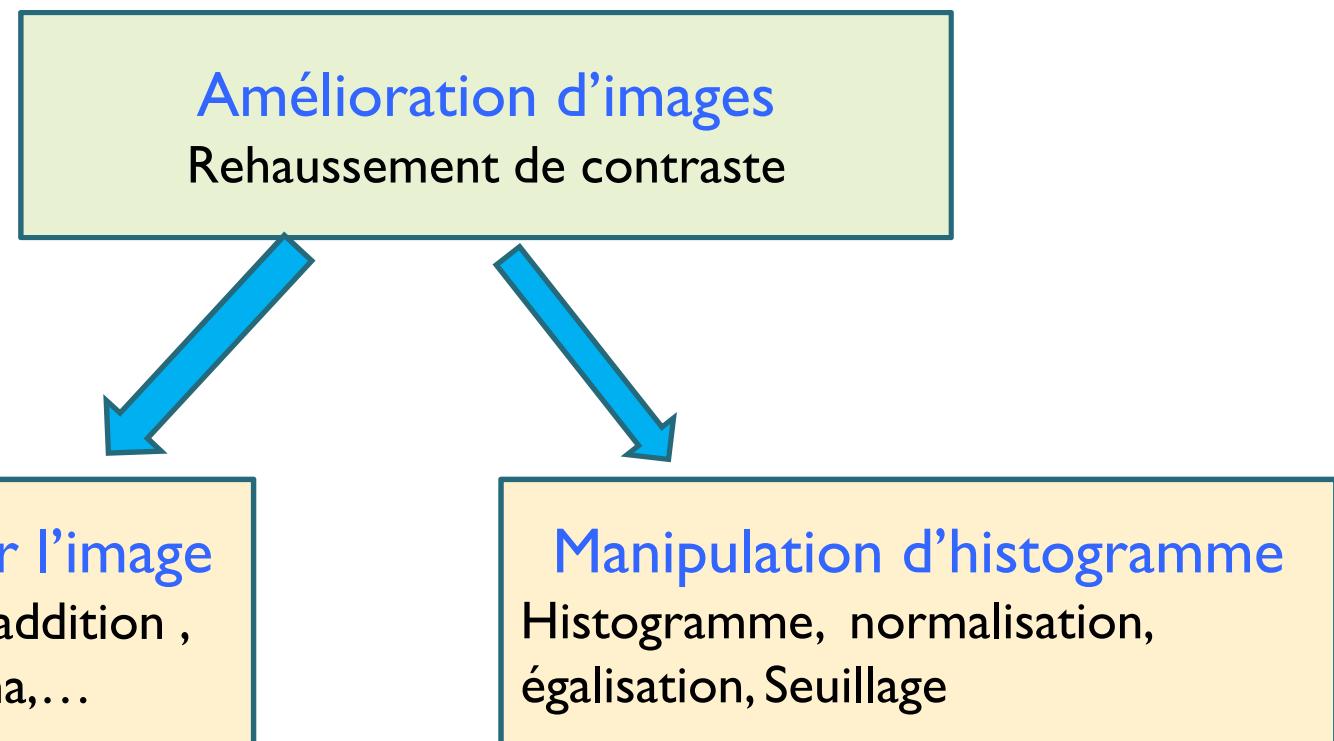
Partie 2:

1. Calculer et représenter l'histogramme de cette image.
2. Calculer et tracer l'histogramme cumulé de cette image
3. En analysant l'image et son histogramme, binariser correctement l'image de façon à séparer la forme rectangulaire du fond. Donner la valeur du seuil choisi et représenter l'image binaire par des 0 et des 1.

Amélioration des images

Conclusion

A retenir pour ce chapitre:



Techniques de traitement d'images

- *Concepts et définitions*
- *Chapitre 1: Images Numériques*
- *Chapitre 2: Amélioration des images*
- *Chapitre 3: Restauration d'images*
- *Chapitre 4: Compression d'images*
- *Conclusion*

Restauration d'images

- Définition
- Echantillonnage & Quantification
- Notion de voisinage
- Produit de convolution
- Techniques de filtrage

Restauration d'image

□ Définition

- La restauration d'images a pour objet la **réduction**, voire **l'élimination** des **distorsions** introduites (**bruits**) par le système ayant servi à acquérir l'image.

□ Objectif

- Obtenir une image qui soit la **plus proche possible de l'image idéale** qui aurait été obtenue si le **système d'acquisition** était **parfait**.

□ Différentes approches :

- Le filtrage (**temporel**)
- Le filtrage (**fréquentiel**)
- Le filtrage **non linéaire**

Restauration d'images

□ Echantillonnage & Quantification

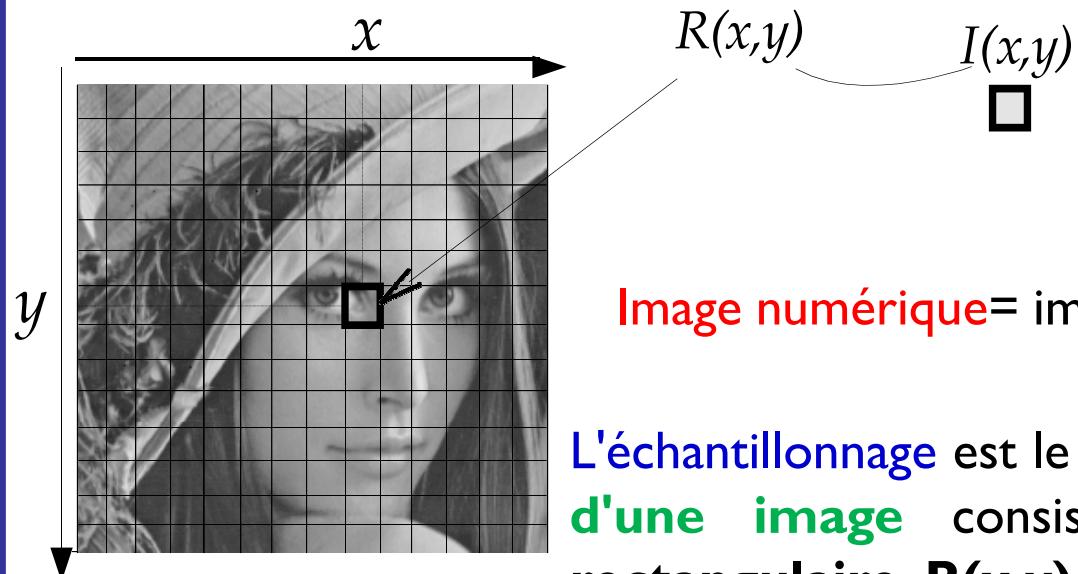


Image numérique = image échantillonnée + image quantifiée

L'échantillonnage est le procédé de **discrétisation spatiale d'une image** consistant à associer à chaque zone rectangulaire $R(x,y)$ d'une image continue **une unique valeur $I(x,y)$.**

On parle de **sous échantillonnage** lorsque l'image est déjà discrétisée et qu'on diminue le nombre d'échantillons.

La **quantification désigne la limitation du nombre de valeurs différentes que peut prendre $I(x,y)$.**



Restauration d'images

□ Echantillonnage & Quantification

L'intensité I est quantifié sur m bits et peut prendre $L = 2^m$ valeurs: $I \in [0, 2^m - 1]$

- ✓ $m = 1$: 2 (2^1) valeurs possibles (images **binaires**)
- ✓ $m = 8$: 256 (2^8) valeurs possibles (images en **niveaux de gris**)
- ✓ $m = 24$: 16777216 (2^{24}) valeurs possibles (images en **couleurs**)

...spatiale :



256x256

Échantillonnage



128x128



64x64



32x32

...tonale :

Quantification



6 bits



4 bits



3 bits



2 bits

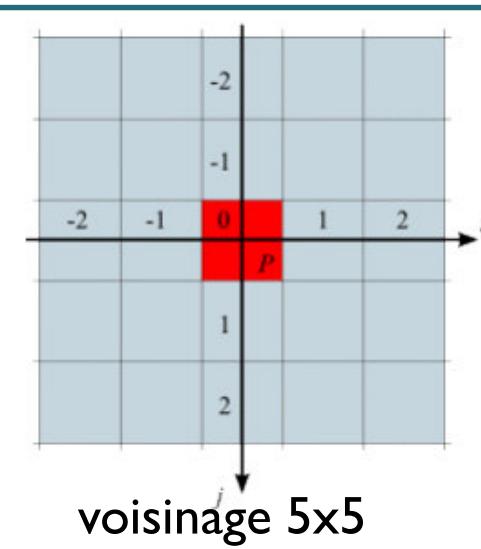
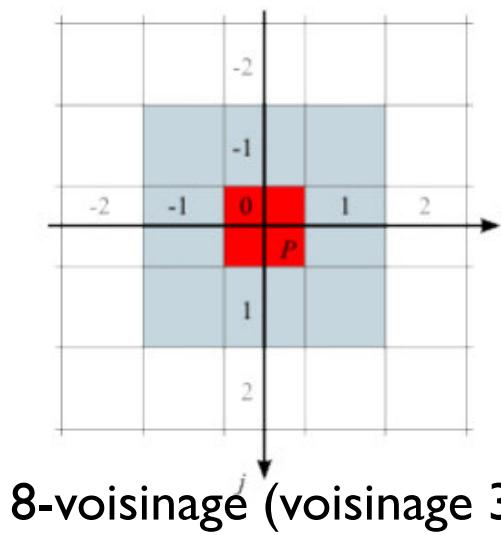
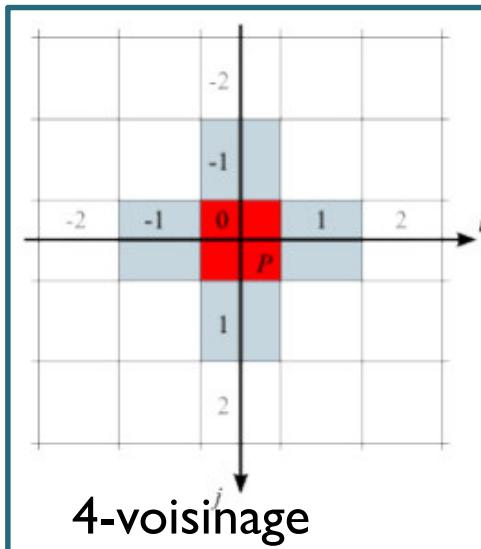


1 bit

Restauration d'images

□ Voisinage $V(P)$ d'un pixel P

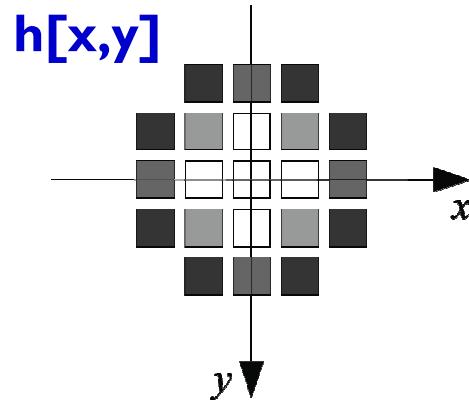
- Hypothèses :
 - V est centré en P
 - les pixels sont disposés selon une **maille carrée**
- Définition : $V(P)$ est l'ensemble des pixels Q situés à moins d'une certaine **distance de P**
- Forme du voisinage (et le **nombre de voisins**) de P dépendent de la **distance considérée**.
- **Voisinages les plus usités en traitement d'images :**



Restauration d'images

□ Produit de convolution

- ✓ Soit I une image numérique.
- ✓ Soit h une fonction de $[x_1, x_2] \times [y_1, y_2]$ à valeurs réelles.



$$(h * I)[x, y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h[i, j] \cdot I[x - i, y - j]$$

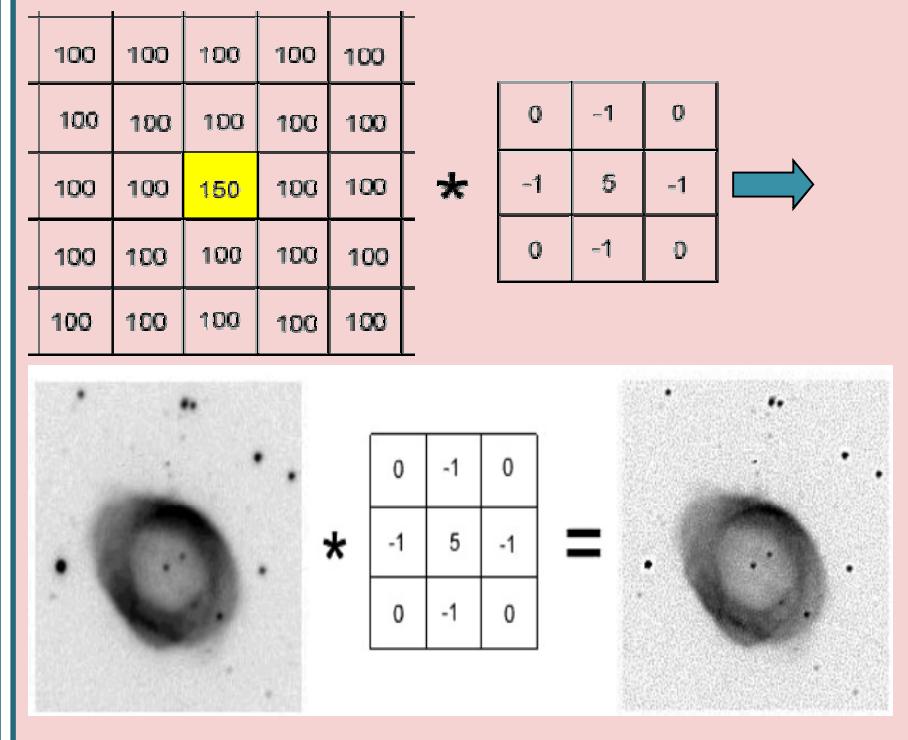
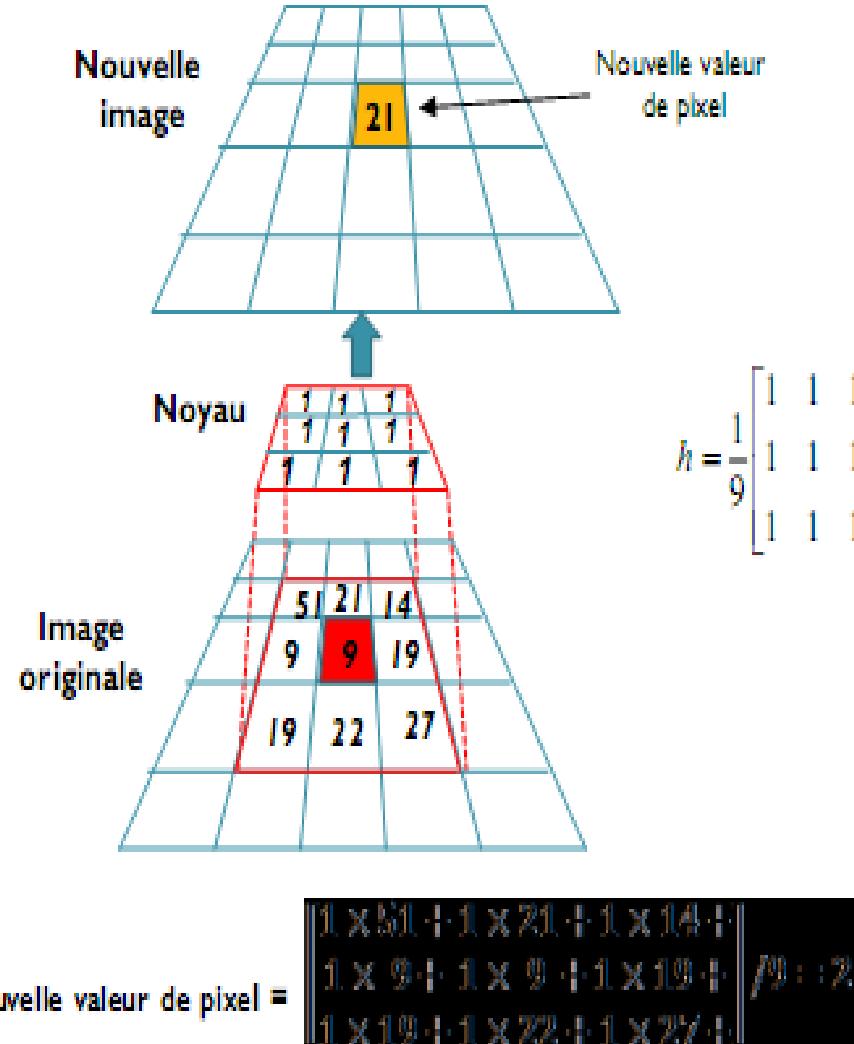
- Fonction h est appelée **noyau de convolution**
- Les **nouvelles valeurs des pixels** de I sont calculées par **produit scalaire** entre le **noyau** et le **voisinage** correspondant du pixel.

Importance de la norme du noyau

Conditions aux bords de l'image

Restauration d'images

□ Produit de convolution (exemple et applications)



- La convolution dans le traitement d'images:
 - Filtrage linéaire des bruits
 - Détection de contours
 - Rehaussement de contraste

Restauration d'images

□ Produit de convolution : Exercice

Soit l'image numérique I de définition 5×5 suivante:

2	1	0	2	3
9	5	4	2	0
2	3	4	5	6
1	2	3	1	0
0	4	4	2	8

$I =$

Soit h le noyau de convolution suivant:

-1	0	1
-1	0	1
-1	0	1

Calculer le produit de convolution $R = h * I$ en mettant à zéro les pixels en dehors de l'image d'origine (bords de l'image)

$$0 * (-1) + 0 * 0 + 0 * 1 + 0 * (-1) + 2 * 0 + 1 * 1 + 0 * (-1) + 9 * 0 + 5 * 1 = 6$$

I : Image originale

0	0	0	0	0	0	0
0	2	1	0	2	3	0
0	9	5	4	2	0	0
0	2	3	4	5	6	0
0	1	2	3	1	0	0
0	0	4	4	2	8	0
0	0	0	0	0	0	0

h : noyau de convolution

-1	0	1
-1	0	1
-1	0	1

*

=

6	-7	-2	-1	-4
9	-5	0	1	-9
10	-1	-2	-5	-8
9	8	-1	3	-8
6	6	-3	1	-3

$R = I * h$

$$5 * (-1) + 4 * 0 + 2 * 1 + 3 * (-1) + 4 * 0 + 5 * 1 + 2 * (-1) + 3 * 0 + 1 * 1 = -2$$

Restauration d'images

Filtrage

- ❖ Le **filtrage** est une opération qui **élimine des éléments perturbateurs / non significatifs (Bruit)** dans les images numériques, soit pour **améliorer leur visualisation**, soit pour **les simplifier en but d'un traitement postérieur**
- ❖ L'objectif est donc **d'améliorer la qualité** de l'image et de mettre en évidence des caractéristiques (contours, formes, ...)
- ❖ Deux types de filtrage :
 - ✓ **Filtrage fréquentiel** (dans le domaine des fréquences, à base de transformées de Fourier non abordé dans ce cours)
 - ✓ **Filtrage spatial**

Restauration d'images

❑ Bruit

- Toute information **parasite** ou **dégradation** que subit l'image de **l'instant de son acquisition** jusqu'à **son enregistrement**.

➤ Bruit lié au **contexte de l'acquisition**

- Bougé, dérive lumineuse, flou, poussière, ...

➤ Bruit lié au **capteur**

- Distorsion de la gamme des niveaux de gris, distorsion géométrique, mauvaise mise au point, ...

➤ Bruit lié à la **numérisation**

- Codage, quantification, échantillonnage



Image source



Flou de mise au point



Bruit aléatoire (impulsionnelle)



Bruit uniforme (gaussien)



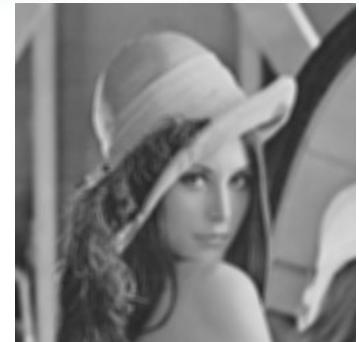
Flou de bougé

Restauration d'images

❖ Filtrage: Deux types de filtre

□ Filtres passe-bas, ou de lissage

- ✓ **Principe** : moyenne pondérée des valeurs du **voisinage**
- ✓ **Effet** : lissage de l'image (variations atténuées)
- ✓ **Avantage** : **atténuation du bruit**
- ✓ **Inconvénient** : **atténuation des détails, flou**
- ✓ **Caractérisation** : coefficients tous positifs



□ Filtres passe-haut, ou de contours

- ✓ **Principe** : **dérivation de la fonction image**
- ✓ **Effet** : **accentuation des détails de l'image**
- ✓ **Avantage** : mise en évidence **des contours/détails**
- ✓ **Inconvénient** : **accentuation du bruit**
- ✓ **Caractérisation** : coefficients de somme nulle



Lissage

Dérivation



Filtrage : Filtres de lissage (passe-bas)

❖ Principe

- ✓ **Utilité**: restauration de l'image (**élimination du bruit**) par lissage.
- ✓ **Inconvénient** : suppression des hautes fréquences (filtres passe-bas), d'où **dégradation des contours et effet de flou**.

❖ Variétés

- ✓ **Plusieurs types de filtres** possédant chacun des avantages propres.
- ✓ **Plusieurs tailles possibles**, selon l'étendue du voisinage à considérer : 3x3, 5x5, ... l'effet de flou est d'autant plus marqué que la taille est grande.

❖ Principaux filtres de lissage

- ✓ **Linéaire**
 - Caractérisés par un masque(**réalisables par convolution**).
 - Exemples : filtres **moyenneurs, gaussiens**
- ✓ **Non-linéaire**
 - Caractérisés par un opérateur non-linéaire(**non réalisables par convolution**).
 - Exemple: filtre **médián**

Filtrage : Filtres de lissage linéaires – Filtre Moyenneur

- ❖ **Configuration** : dépend de l'importance à donner au pixel d'analyse et à ses voisins :

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Ou

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Ou

$$\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- ❖ **Le filtre moyenneur permet de :**

- ✓ Lisser l'image (Smoothing)
- ✓ Remplacer chaque pixel par la valeur moyenne de ses voisins
- ✓ Réduire le bruit et les détails non-important
- ✓ Brouiller ou rendre floue l'image (blur edges).



- ❖ **Inconvénients:**

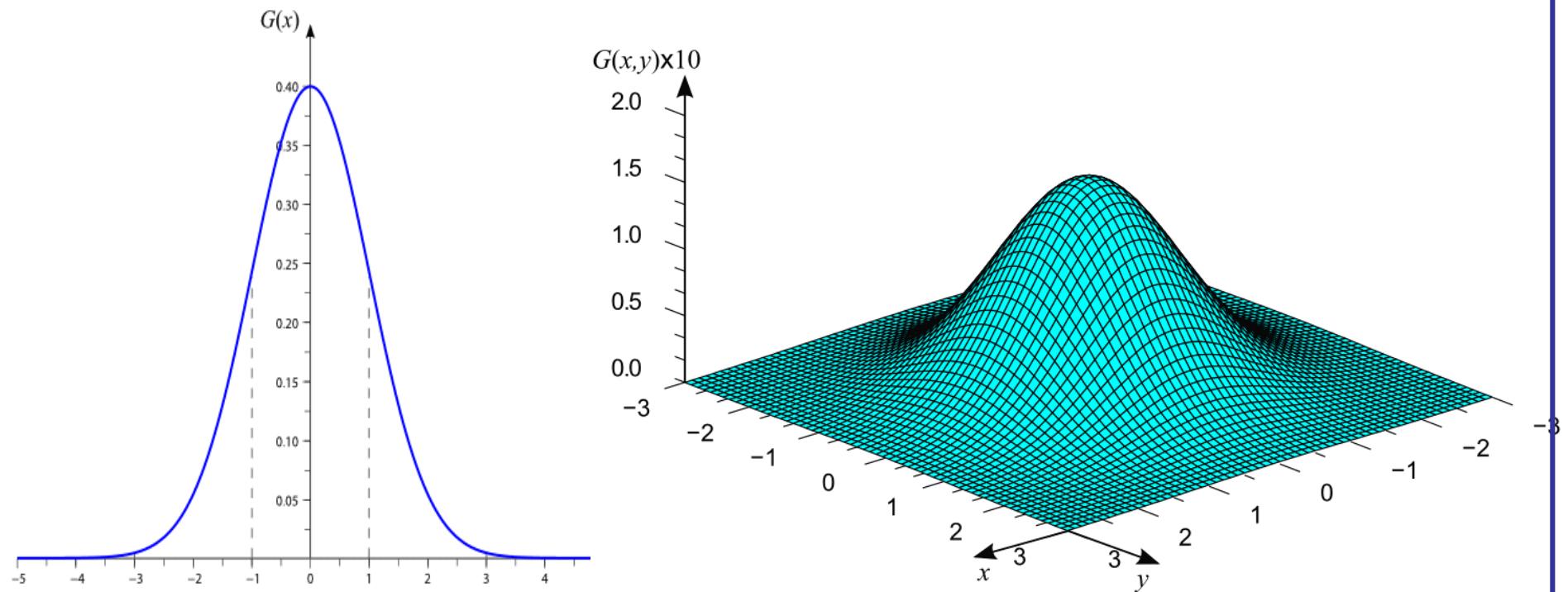
- ✓ Forte atténuation des contours (limite la performance des traitements ultérieurs)
- ✓ Forte influence des pixels aberrants isolés

Filtrage : Filtres de lissage linéaires – Filtre gaussien

❖ **Paramètres** : moyenne μ , écart-type σ

✓ EN 1 D : $G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$

✓ EN 2 D : $G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-\mu)^2+(y-\mu)^2}{2\sigma^2}\right)$



Filtrage : Filtres de lissage linéaires – filtre gaussien

- ❖ **Avantage** : limite l'effet de flou (**contours mieux conservés**)
- ❖ **Configurations** : approximations discrètes de la distribution gaussienne de moyenne $\mu = 0$ et d'écart-type σ dans un filtre fini. Exemple pour $\sigma = 0.6$:

$G(-1, -1)$	$G(0, -1)$	$G(+1, -1)$	$\cong \frac{1}{16}$	1	2	1
$G(-1, 0)$	$G(0, 0)$	$G(+1, 0)$		2	4	2
$G(-1, +1)$	$G(0, +1)$	$G(+1, +1)$		1	2	1

- ❖ **Écart-type:**
 - ✓ Détermine le degré de lissage
 - ✓ Impose la taille du masque (idéalement $5\sigma \times 5\sigma$)
- ❖ **Remarque** : En général un filtre gaussien avec $\sigma < 1$ est utilisé **pour réduire le bruit**, et si $\sigma > 1$ c'est dans le but de fabriquer une image qu'on **va utiliser pour faire un masque flou personnalisé**.
- ❖ Il faut noter que plus σ est grand, plus le flou appliqué à l'image sera marqué.

Filtrage : Filtres de lissage non-linéaires – filtre médian

❖ Principe

- ✓ Le niveau de gris **résultat** est le niveau de gris médian des pixels voisins.
- ✓ Filtre non-linéaire, donc **non réalisable par masque de convolution**.

❖ Calcul

- ✓ Trier les niveaux par **ordre croissant**.
- ✓ Donner au pixel d'analyse le niveau situé au **milieu des niveaux triés**.

❖ Exemple

18	20	25
14	255	32
22	27	25

→ $14 \leq 18 \leq 20 \leq 22 \leq 25 \leq 25 \leq 27 \leq 32 \leq 255 \rightarrow 25$

| ← 4 valeurs → | ← médiane → | ← 4 valeurs → |

Filtrage : Filtres de lissage non-linéaires – filtre médian

❖ Avantages par rapport aux filtres moyenneur et Gaussien

- ✓ Filtre **mieux** le bruit impulsif (type «poivre et sel»).
- ✓ Ne crée pas de nouveau niveau, **préserve mieux les contours sans altérer le fond.**



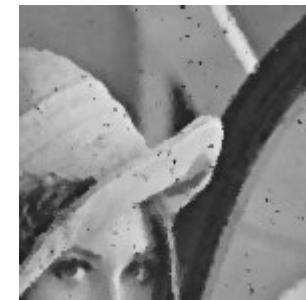
I



moy 3x3



gauss 5x5



méd 3x3

❖ Limites et inconvénients

- ✓ Supprime les détails fins qui ne sont pas du bruit.
- ✓ Détruit les coins.
- ✓ Coûteux en temps de calcul (tri).

Filtrage : Filtres de lissage non-linéaires – filtre d'ordre

❖ Principe

- ✓ Le filtre d'ordre est une généralisation du filtre médian.
- ✓ Après avoir trier les valeurs au voisinage du pixel, le filtre d'ordre procède **par remplacer la valeur du pixel par celle qui occupe un certain rang**

❖ Calcul

$$f(1) \leq f(2) \leq \dots \leq f(k) \leq \dots \leq f(2m + 1)$$

- ✓ Le filtre d'ordre **de rang k** est défini par:

$$I(x, y) = f(k)$$

❖ Cas particuliers

- ✓ $k=m$: Filtre médian
- ✓ $k=1$: l'érosion morphologique (la valeur minimale)
- ✓ $K=2m+1$: la dilatation morphologique (la valeur maximale)

Filtrage : Filtres de lissage non-linéaires – Filtre min - max

❖ Principe

- ✓ Garantit que la valeur de tout pixel appartient à l'intervalle des valeurs de ses voisins → débruitage efficace.
- ✓ Préserve mieux les contours que le filtre médian.

❖ Calcul

$$I'(x, y) = \begin{cases} I(x, y) & \text{si } i_{min} \leq I(x, y) \leq i_{max} \\ i_{min} & \text{si } I(x, y) < i_{min} \\ i_{max} & \text{si } I(x, y) > i_{max} \end{cases}$$

❖ Exemple

124	126	127
120	255	125
115	119	123

$$i_{min} = 115 \quad i_{max} = 127$$

$$\text{moyenne} = 137$$

$$\text{médiane} = 124$$

$$\text{min-max} = 127$$

Mesure de distorsion

❖ PSNR (Peak Signal to Noise Ratio)

- ✓ Le PSNR (unité décibel dB) permet de mesurer la similarité entre deux images I_1 et I_2 codées sur 8 bits et de définition $M \times N$ pixels.
- ✓ Il peut être utilisé pour quantifier la puissance du bruit ajouté à l'image ou pour mesurer la qualité du débruitage.

❖ Calcul

$$PSNR(I_1, I_2) = 10 \log_{10} \left(\frac{255^2}{EQM(I_1, I_2)} \right)$$

Où EQM désigne l'erreur quadratique moyenne entre les 2 images :

$$EQM(I_1, I_2) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (I_2(x, y) - I_1(x, y))^2$$

❖ Remarque :

- ✓ Si $I_1 = I_2$ alors $PSNR(I_1, I_2) = +\infty$

Exercice 1

Répondez par VRAI ou FAUX aux affirmations suivantes **en justifiant votre réponse dans le cas où vous répondez par faux :**

- a. Le bruit d'une image désigne les pixels de l'image dont l'intensité est très proche de celles des pixels voisins.
- b. Le principe du filtre gaussien est d'effectuer une convolution avec une exponentielle
- c. Le filtre gaussien ne possède pas un masque de convolution.
- d. La caractéristique essentielle du filtre médian est sa capacité à conserver des transitions fortes tout en supprimant une partie importante du bruit.

Exercice 2

On considère l'image de taille 3x3, numérisée selon 8 niveaux de gris, suivante :

$$I = \begin{bmatrix} 0 & 10 & 20 \\ 30 & 20 & 10 \\ 40 & 30 & 40 \end{bmatrix}$$

- a. Réaliser le filtrage de cette image en utilisant le filtre moyenneur ci-dessous
(Effets de bords: Pixels à 0).

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

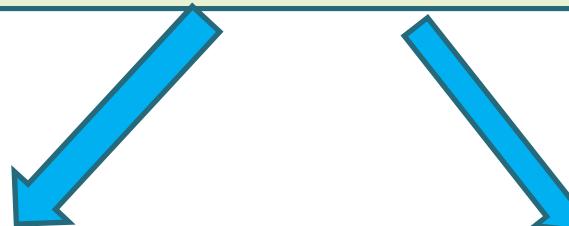
- b. Réaliser le filtrage de cette image en utilisant le filtre médian de taille 3x3
(Effets de bords: Pixels à 0).

Restauration d'images

Conclusion

A retenir pour ce chapitre:

Restauration d'images
Définition, Produit de convolutions



Filtrage linéaire
Filtre moyenneur, filtre gaussien, ...

Filtrage non-linéaire
Filtre min - max, filtre médian, ...

Techniques de traitement d'images

- *Concepts et définitions*
- *Chapitre 1: Images Numériques*
- *Chapitre 2: Amélioration des images*
- *Chapitre 3: Restauration d'images*
- *Chapitre 4: Compression d'images*
- *Conclusion*

Compression d'images

- Contexte & Intérêt de la compression
- Compression sans perte (codage)
- Compression avec perte
- Compression audio/vidéo
- Conclusion

Contexte & Intérêt de la compression

- ❖ Position du problème
- ❖ Définition & Intérêt de la compression
- ❖ Types de compression
- ❖ Evaluation de la compression et des pertes

Contexte & Intérêt de la compression

Position du problème

→ Exemple 1: Transmission par télécopie

Une page à transmettre est constituée de points blancs et noirs. Chaque point est représenté par un élément binaire (“0” si la couleur est noire, “1” si elle est blanche).

La taille (Largeur * Hauteur) de cette page est de **8,5x11 pouces**. Sachant que la résolution est de **200 points par pouce**.

1. Calculer le nombre d’éléments binaires nécessaires pour représenter cette page .

$$(8,5 \times 200) \times (11 \times 200) = 3,74 \text{ Mbits}$$

2. Si on utilise un modem au débit de 14,4 Kbits/s, donner le temps nécessaire pour la transmission de cette page.

$$3,74 \cdot 10^6 / 14,4 \cdot 10^3 = 4 \text{ mins } 20 \text{ s.}$$



Grâce aux méthodes de compression, cette durée est réduite à 17s !

Contexte & Intérêt de la compression

Position du problème

→ Exemple 2: Fichiers musicaux

Considérons un signal analogique stéréo. Sa numérisation en “qualité CD” requiert une **fréquence d'échantillonnage de 44,1 KHz**, et une quantification des échantillons sur 16 bits (2 octets).

- Donnez le nombre d'élément binaire nécessaire à la représentation d'une **seconde** de musique (stéréophonique \Leftrightarrow Nombre de voies=2)

$$44,1 \cdot 10^3 \times 2 \times 2 = 176,4 \text{ ko/s}$$

- Avec un tel débit, donnez la durée qu'un CD de 650 Mo peut stocker .

$$650 / 0,1764 = 61 \text{ minutes de musique.}$$



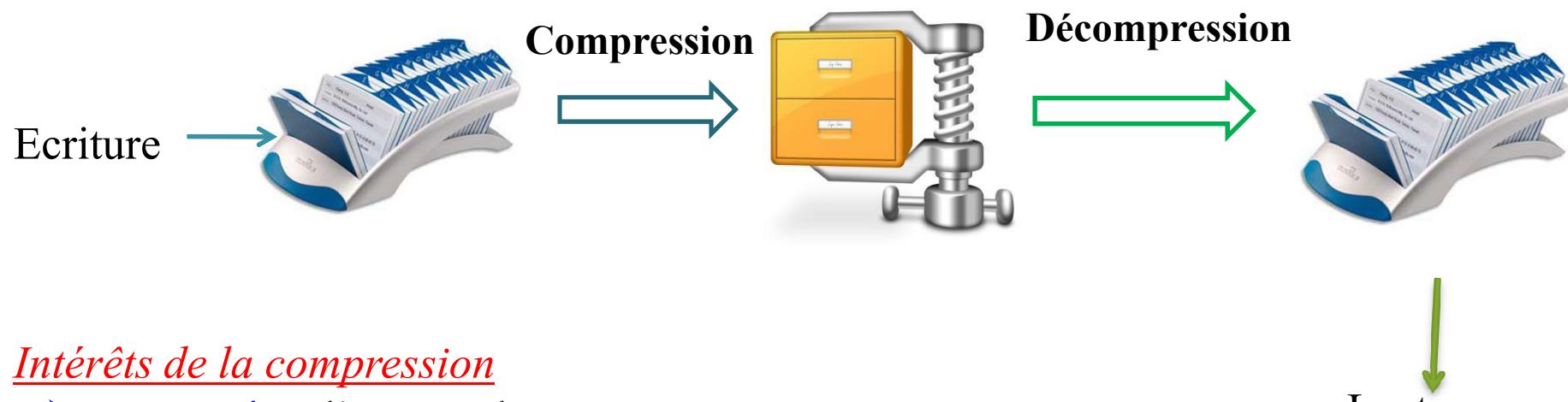
Grâce aux méthodes de compression, on peut stocker sur ce même CD plus de 10 heures de musique !

Contexte & Intérêt de la compression

➊ Définition & Intérêts de la compression

Définition

- La compression de données informatiques consiste à **réduire** la taille de l'information pour son **stockage** et son **transport**.
- Le coût et les limites technologiques nécessitent d'utiliser la compression de données pour le stockage d'importants volumes d'information.



Intérêts de la compression

- Economiser l'espace de **stockage**.
- Diminuer le temps de **transmission** du fichier.

Contexte & Intérêt de la compression

➊ Principe & Types de de la compression

Principe

1. Détection de **redondances** dans le signal
2. Un algorithme de *compression permet le codage réduit du signal*
3. Un algorithme (inverse) de *décompression permet d'exploiter le signal*

Types de compressions

❖ **Compression sans perte (ou non-destructive, i.e. codage ou compactage):**

- Le signal obtenu après décompression est strictement identique à l'original
- Utilisation : fichier exécutable, fichier texte

❖ **Compression avec perte (ou destructive, ou avec dégradation) :**

- Le signal obtenu après décompression est différent (légèrement) de l'original
- Utilisation : image, son, vidéo

Contexte & Intérêt de la compression

● Evaluation de la compression et des pertes

→ Quotient de compression :

$$Q = \frac{\text{Taille_Initiale}}{\text{Taille_Finale}}$$

➤ Plus une compression est **forte**, plus le **quotient** de compression est **élevé**.

→ Taux de compression :

$$T = \frac{1}{Q}$$

→ Gain de compression (%) :

$$G = (1-T) * 100\%$$

➤ Plus le **gain** de compression est **élevé**, plus la **taille** du fichier compressé résultant est **faible**.

→ Erreur quadratique moyenne :

$$\text{EQM} = \frac{1}{M \times N} \times \sum_{n=1}^N \sum_{m=1}^M \left(X_{\text{origine}}(n, m) - X_{\text{reconstruite}}(n, m) \right)^2$$

Avec:

$X_{\text{origine}}(n, m)$ et $X_{\text{reconstruite}}(n, m)$: représentent les valeurs des pixels d'image d'origine et reconstruite

Contexte & Intérêt de la compression

➊ Evaluation de la compression et des pertes

▪ Qu'est ce qu'une bonne compression ?

Un algorithme performant de compression possède un **gain de compression maximal** et une **erreur quadratique moyenne minimale**.

→ ATTENTION : compte tenu des information effectivement perçues par notre œil, il est possible d'avoir à la fois une image de “qualité” et une EQM importante !

Contexte & Intérêt de la compression

Compression des principaux formats d'images bitmaps

Format	Espaces couleur	Compression(s)	C. α	Domaines d'utilisation, rem.
TIFF (.tif)	<i>RGB</i> , CIE $L^*a^*b^*$, <i>CMYB</i> , couleurs indexées, ndg	Aucune Sans perte (LZW, Huffman) Avec perte (JPEG)	Oui	PAO, Infographie, bureautique Très flexible, mais nombreuses variantes pas toujours supportées
BMP (.bmp)	<i>RGB</i> , couleurs indexées, ndg	Aucune Sans perte (RLE)	Non	Bureautique sous Windows Compression peu efficace
GIF (.gif)	couleurs indexées (2 à 256)	Sans perte (LZW)	Oui	Pages web Animations possibles
JFIF (.jpg)	<i>RGB</i> , <i>CMYB</i> , ndg	Avec perte (JPEG)	Non	Pages web, photographie Compr. efficace mais destructive
PNG (.png)	<i>RGB</i> , ndg, 256 couleurs indexées	Sans perte (deflate=LZ77+Huffman)	Oui	Pages web, photo. sans perte Format libre. Jusqu'à 48 bits.

Canal α : une 4ème couche qui vient s'ajouter aux 3 couches Rouge, Vert et Bleu (RVB), qui permet de stocker la **transparence** de l'image

- Contexte & Intérêt de la compression*
- Compression sans perte (codage)*
- Compression avec perte*
- Compression audio/vidéo*
- Conclusion*

Compression sans perte (codage)

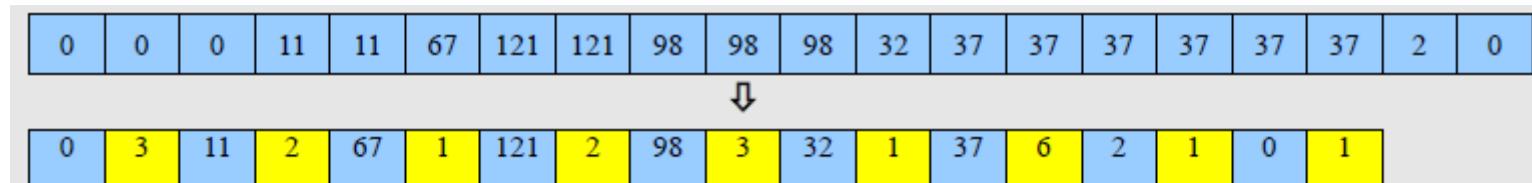
- ❖ RLE (Run-Length Encoding)
- ❖ Codage de Huffman
- ❖ Méthodes par dictionnaire

Compression sans perte (codage)

➊ RLE (Run-Length Encoding)

Principe

- Codage par plage (« Running Length Encoding »).
- Recherche de séquences de données **redondantes** (ex. niveaux identiques).
- **Codage de la valeur et du nombre de répétitions:**



Avantage

- Algorithmes de compression et décompression très **simples** et **rapides**.

Limites

- Efficace seulement pour **nombreuses et longues plages constantes**.
 - Cas des images de synthèse simples ; **peu adapté aux photos**.
 - Utilisé *ponctuellement dans de nombreux formats (BMP, JPG, TIFF, PCX, ...)*.
- Nécessite de **fixer un maximum pour la longueur des plages** (ex. 255).

Compression sans perte (codage)

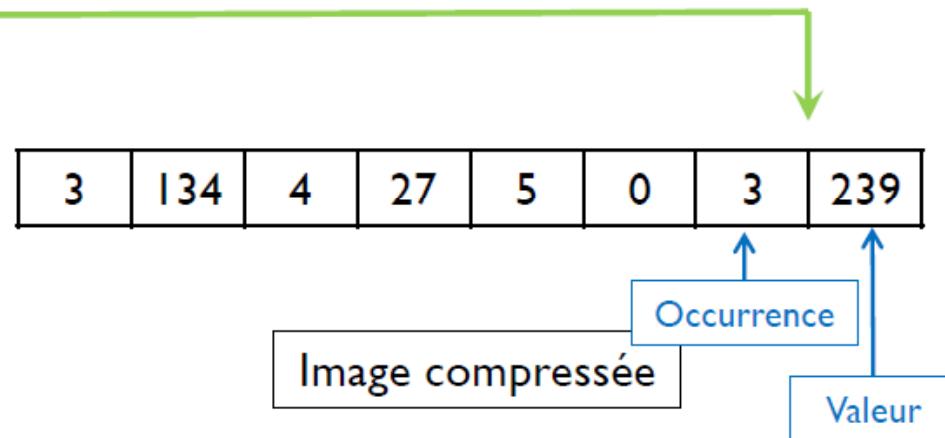
RLE (Run-Length Encoding)

Exemple 1: soit l'image en niveaux de gris ci-dessous (image non compressée)

Compression par RLE

134	134	134	27	27
27	27	0	0	0
0	0	239	239	239

Image non compressée



Taille initiale : $5 \times 3 \times 1$ (octet) = 15 octet

Taille de l'image compressée : $8 \times 1 = 8$ octet

Compression sans perte (codage)

RLE (Run-Length Encoding)

Exemple 2: Estimation du quotient de compression

Soit une chaîne de caractère de taille N à compresser. Supposons que cette chaîne contienne M répétitions de longueur moyenne L d'un caractère. Quel est le quotient de compression si cette chaîne est compressée selon l'algorithme RLE ?

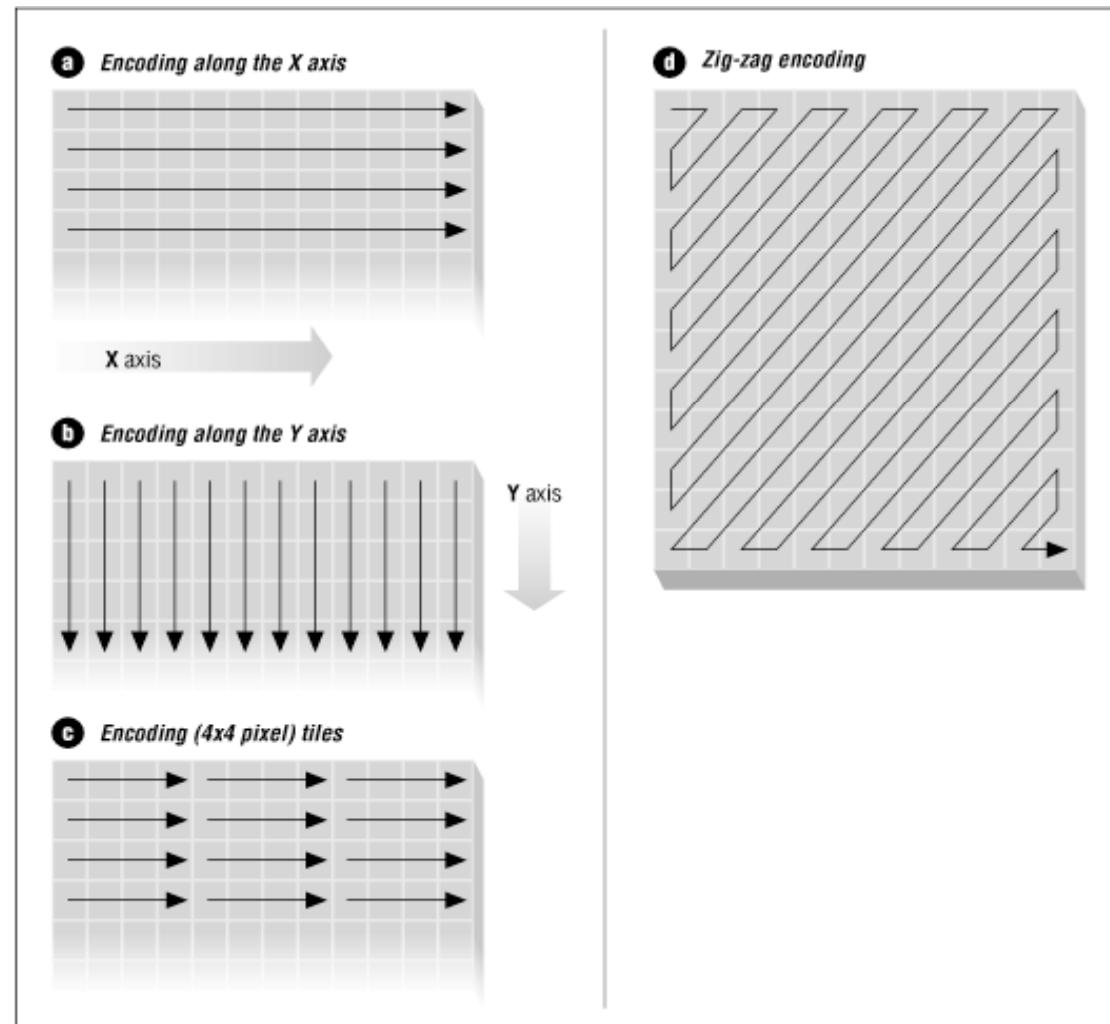
- ✓ Chacune des M répétitions est remplacée par 2 caractères
- ✓ Donc la taille de la chaîne compressée est $N - ML + 2M$
- ✓ D'où

$$Q \approx \frac{N}{N - M(L - 2)}$$

Compression sans perte (codage)

❶ RLE (Run-Length Encoding)

Comment parcourir l'image ?



Compression sans perte (codage)

- ❖ RLE (Run-Length Encoding)
- ❖ Codage de Huffman
- ❖ Méthodes par dictionnaire

Compression sans perte (codage)

❶ Codage de Huffman

Principe

➤ Coder les valeurs avec un **nombre de bits différent**.

Code (utilisant des mots) à **longueur variable** (ang. « *Variable Length Coding* »), dit aussi codage entropique (ang. « *Entropy coding* »).

➤ Plus une **valeur apparaît fréquemment**, plus le **nombre de bits** utilisés pour la coder est **petit** (i.e. plus son code est court).

Algorithme de Huffman : codage

Phase 1 : Construction de l'arbre.

1. Trier les différentes valeurs par ordre **décroissant de fréquence d'apparition**
 - Table de *poids*.
2. Fusionner les **deux poids minimaux** dans un arbre binaire et **affecter leur somme à la racine**.
3. Réordonner la table de poids par poids décroissants.
4. Recommencer en 2. jusqu'à obtenir un **seul arbre**.

Phase 2 : Construction du code à partir de l'arbre obtenu dans la phase 1.

À partir de la racine, attribuer des **0** aux **sous-arbres de gauche** et des **1** à **droite**.

Compression sans perte (codage)

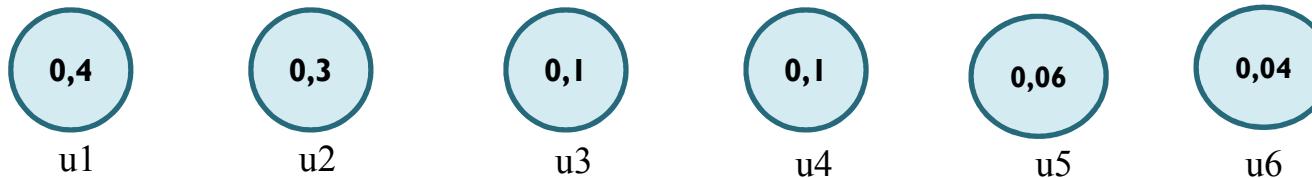
Codage de Huffman

Exemple 1:

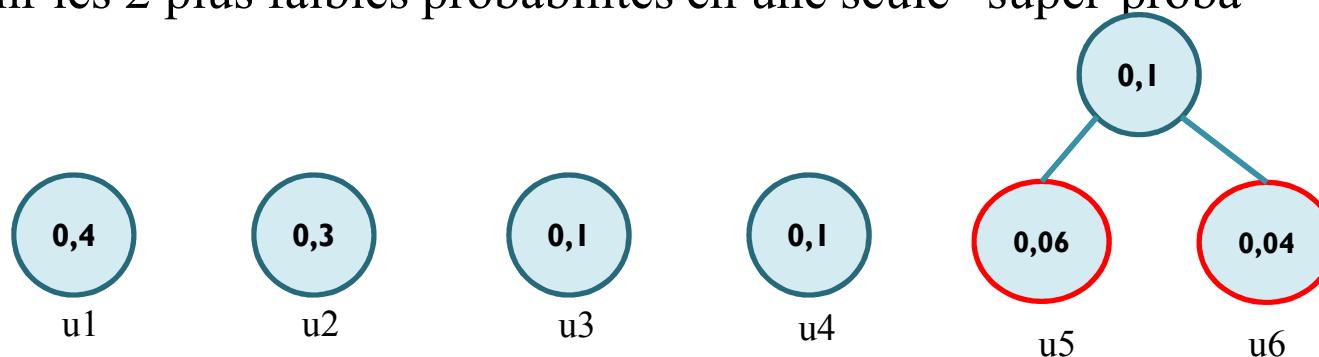
Soit une source émettant les symboles u_1, \dots, u_6 , avec des probabilités de 0.4, 0.3, 0.1, 0.1, 0.06 et 0.04 respectivement.

Phase 1 : Construction de l'arbre.

Etape 1: Classer les symboles par ordre décroissant de probabilité



Etape 2: Réunir les 2 plus faibles probabilités en une seule “super-proba”

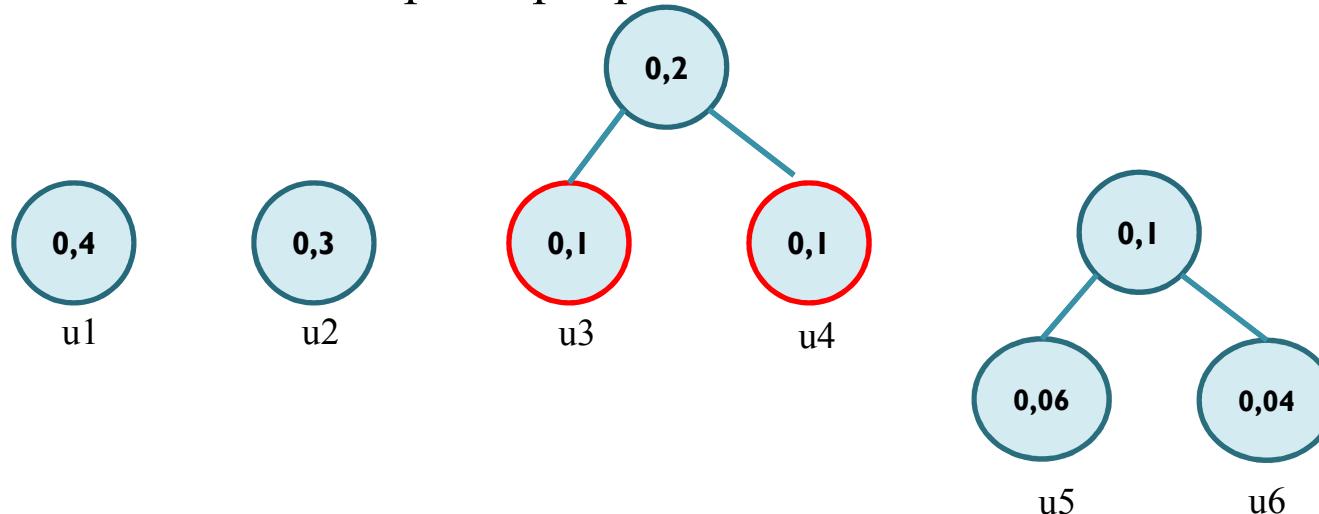


Compression sans perte (codage)

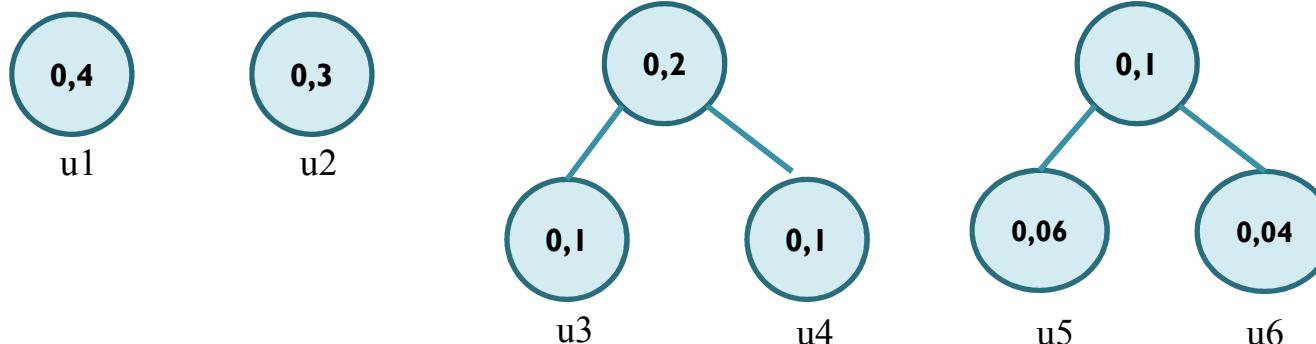
Codage de Huffman

Exemple 1:

Etape 3: Réordonner la table de poids par poids décroissants



Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre

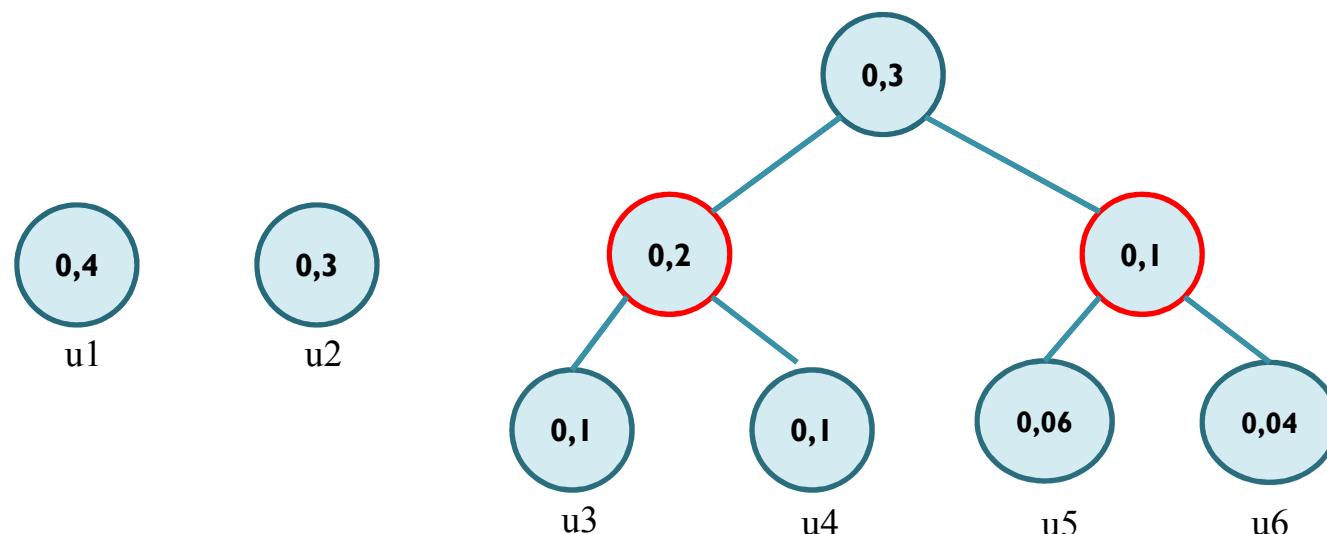


Compression sans perte (codage)

❶ Codage de Huffman

Exemple 1:

Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre

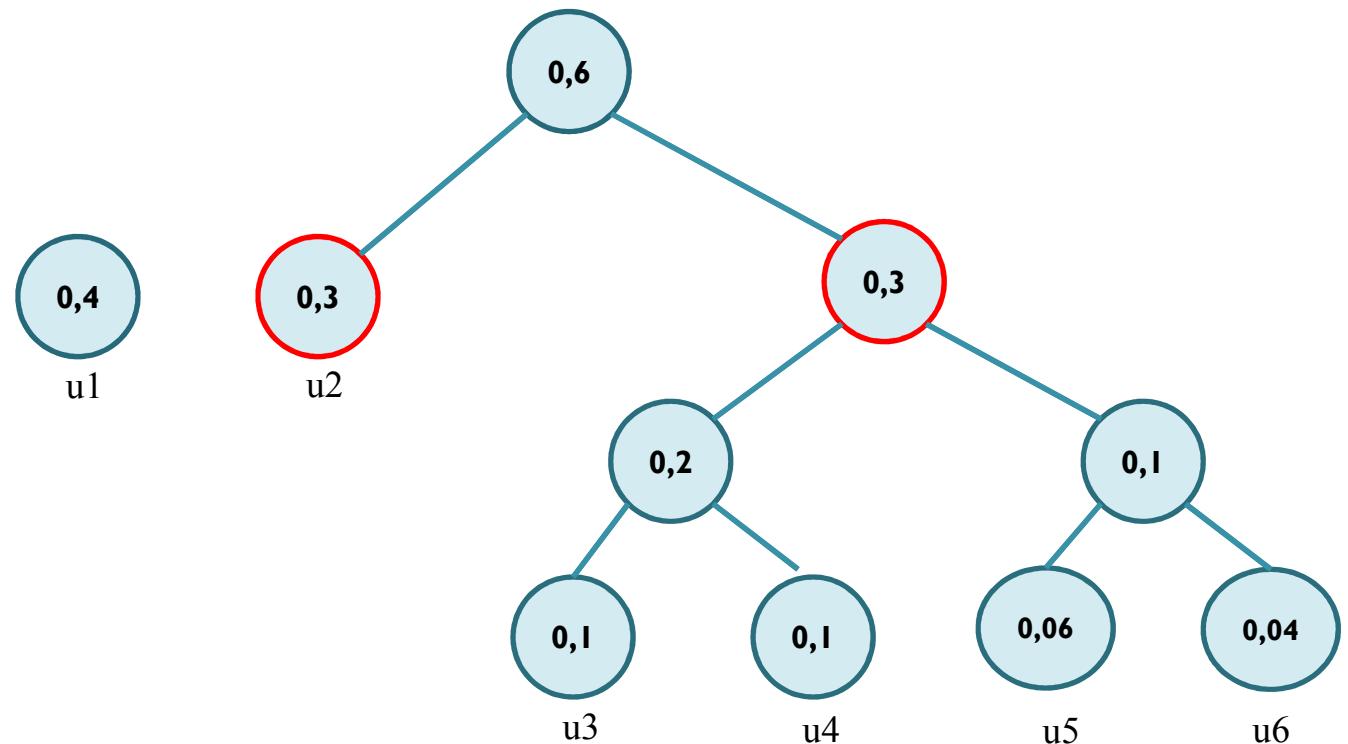


Compression sans perte (codage)

Codage de Huffman

Exemple 1:

Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre

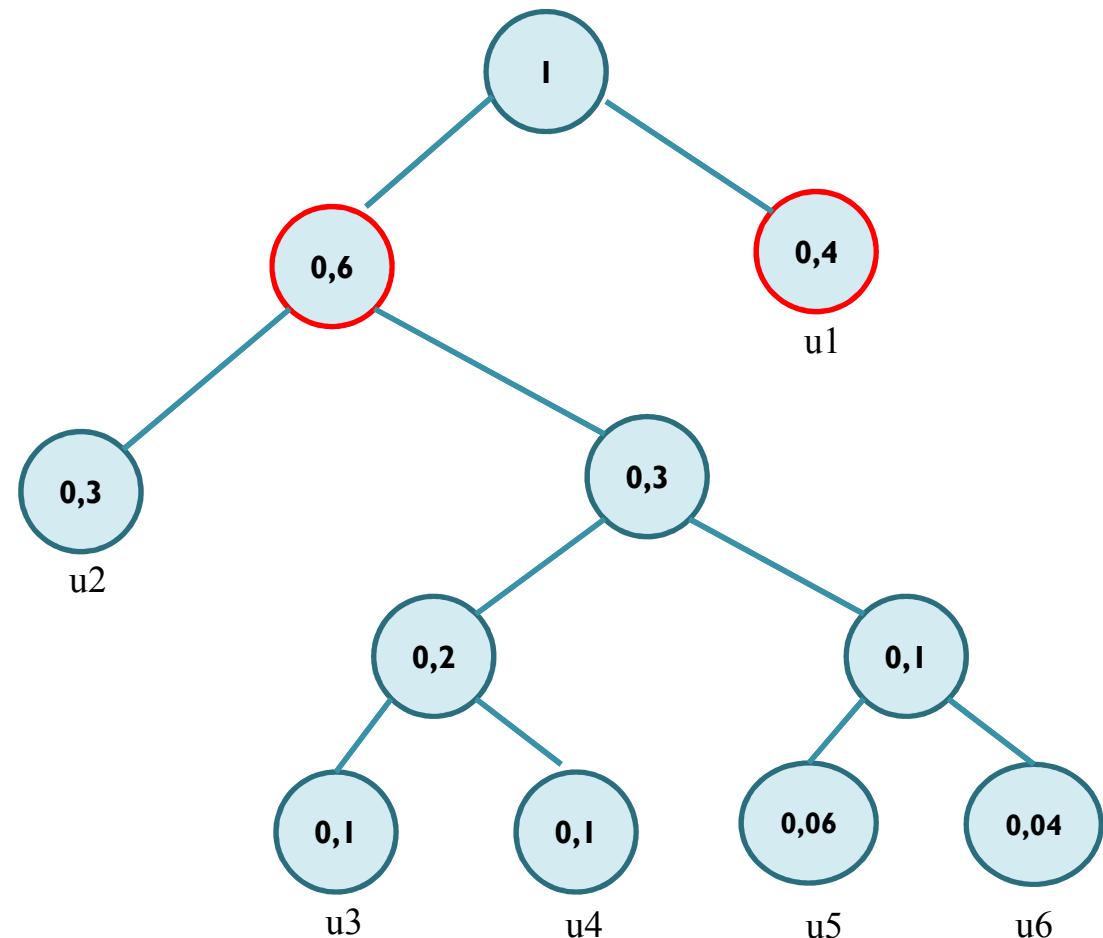


Compression sans perte (codage)

❶ Codage de Huffman

Exemple 1:

Etape 4: répéter les étapes 2 et 3 jusqu'à obtenir un seul arbre

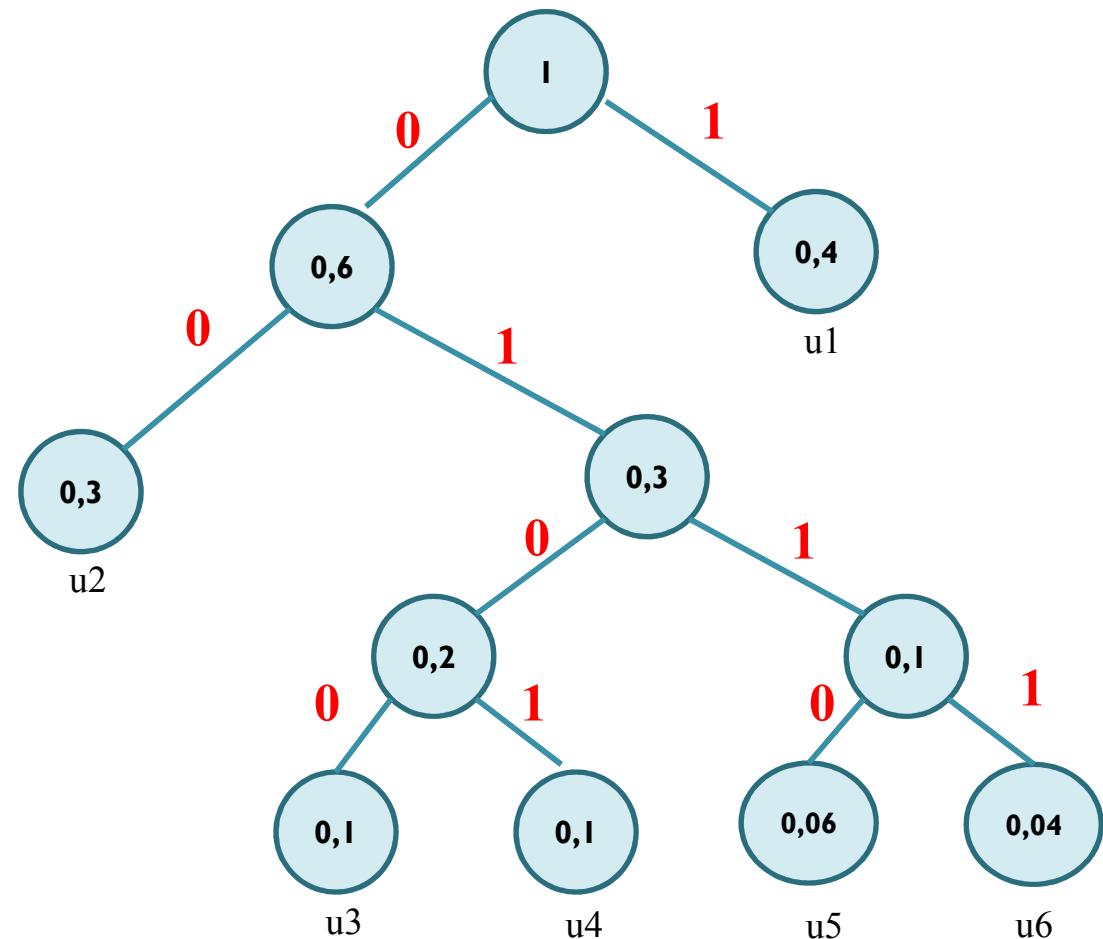


Compression sans perte (codage)

❶ Codage de Huffman

Exemple 1:

Phase 2 : Construction du code à partir de l'arbre obtenu dans la phase 1.



Compression sans perte (codage)

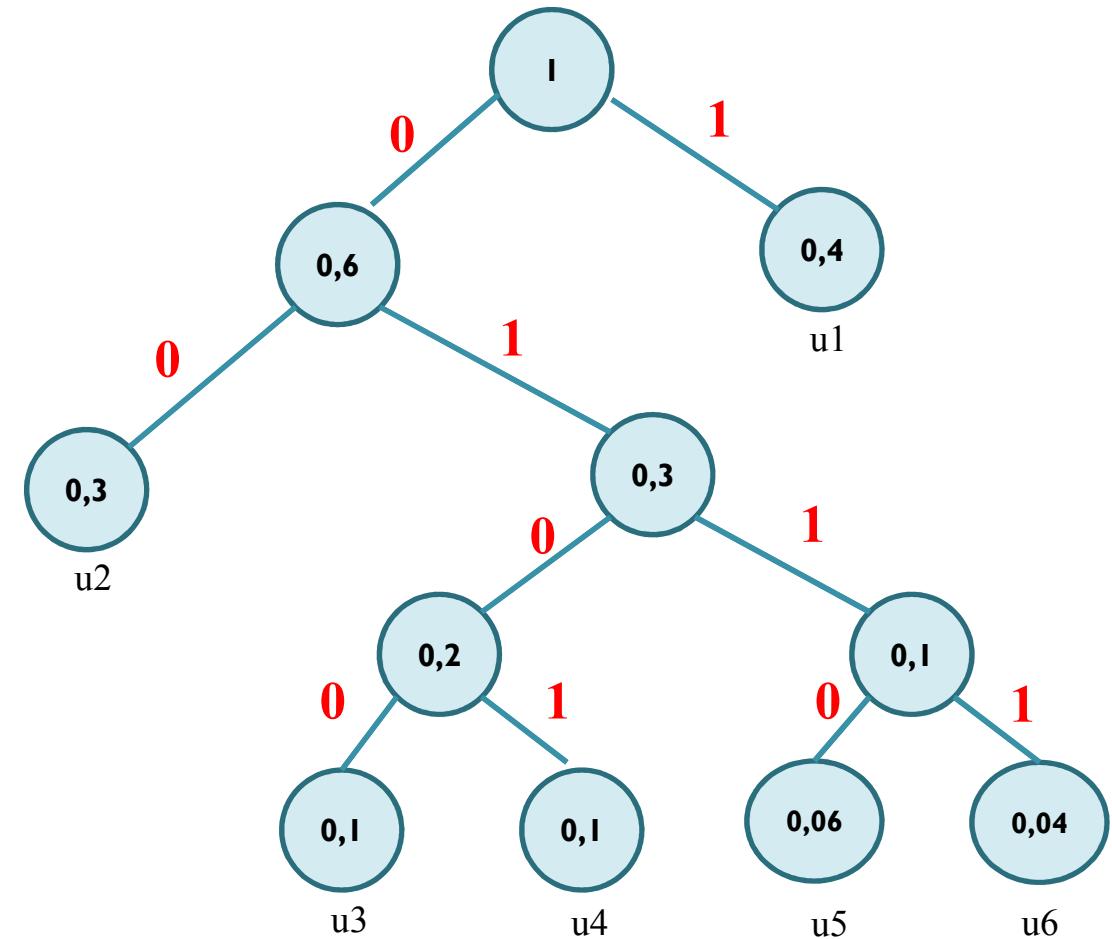
Codage de Huffman

Exemple 1:

Phase 2 : Construction du code à partir de l'arbre obtenu dans la phase 1.

Lecture du codage obtenu,
du haut vers le bas

Symbole	Probabilité	Code
u1	0,4	“1”
u2	0,3	“00”
u3	0,1	“0100”
u4	0,1	“0101”
u5	0,06	“0110”
u6	0,04	“0111”



Compression sans perte (codage)

Codage de Huffman

Exemple 1:

Symbole	Probabilité	Nombre de bits (avant compression)	Nombre de bits (après compression)
u1	0,4	8	1
u2	0,3	8	2
u3	0,1	8	4
u4	0,1	8	4
u5	0,06	8	4
u6	0,04	8	4

Poids avant compression : 48 bits;

Poids après compression : 19 bits

Compression sans perte (codage)

Codage de Huffman

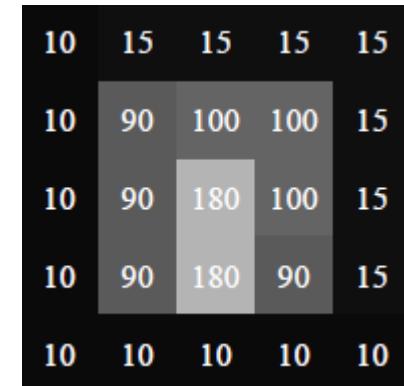
Exemple : compression d'image niveaux de gris

➤ Construction de l'arbre

1. Table des poids

$$10_9 \mid 15_7 \mid 90_4 \mid 100_3 \mid 180_2 \mid$$

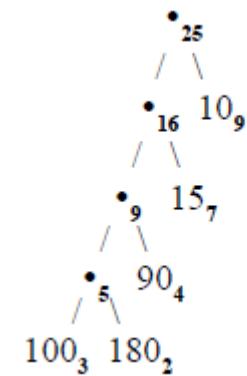
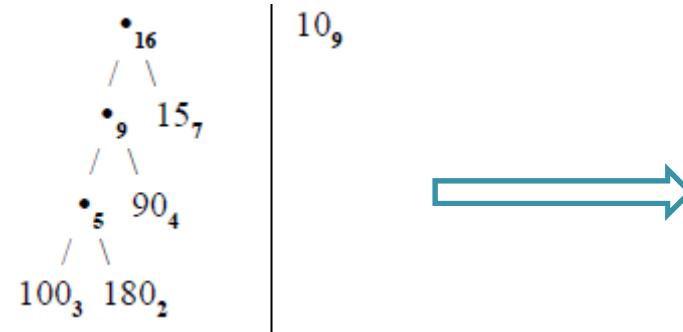
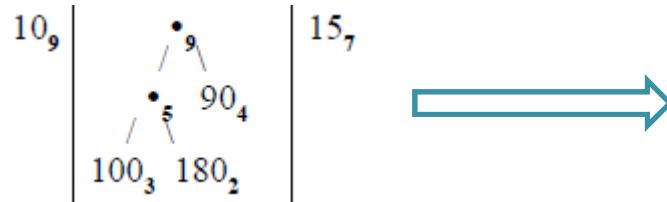
2. Fusion des poids minimaux

$$10_9 \mid 15_7 \mid 90_4 \mid \begin{array}{c} \bullet_5 \\ / \backslash \\ 100_3 \ 180_2 \end{array}$$


3. Réordonnancement

$$10_9 \mid 15_7 \mid \begin{array}{c} \bullet_5 \\ / \backslash \\ 100_3 \ 180_2 \end{array} \mid 90_4$$

4. Itérations

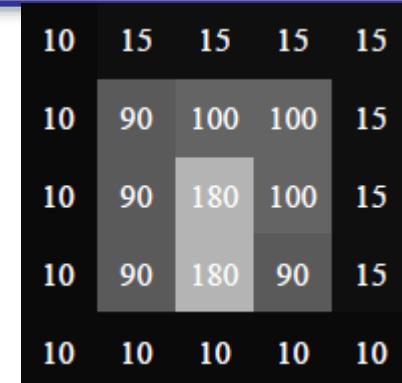
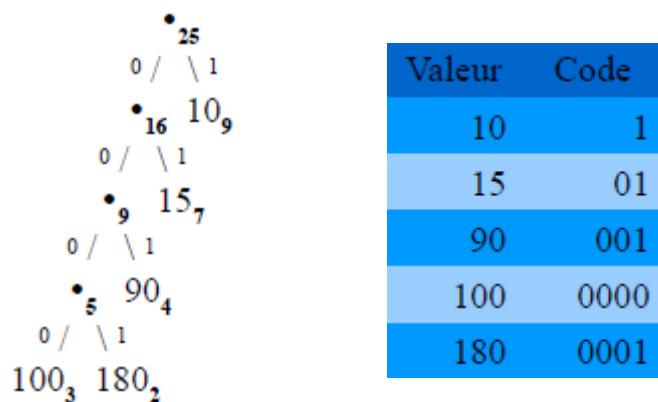


Compression sans perte (codage)

Codage de Huffman

Exemple : compression d'image niveaux de gris

➤ **Construction du code:** Affectation de valeurs binaires aux arcs



➤ **Image codée (en lignes)**

10101010110010000 ...

soit 55 bits (image compressée) vs. $25 \times 8 = 200$ bits

Décompression

Le décodeur doit connaître la table de codage (entête) ;
extrait les valeurs au plus tôt :

Entrée	Action	Buffer	Émission
1	Identification de 10	vide	10
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
0	Bufferise et attend	0	rien
1	Identification de 15	vide	15
...

Compression sans perte (codage)

Codage de Huffman

Exercice 1

Soit la phrase suivante : *TO BE OR NOT TO BE*

1. Déterminer le nombre d'occurrence de chaque caractère
2. Trier les caractères selon les fréquences d'apparition décroissantes et en cas d'égalité l'ordre alphabétique
3. Appliquer l'algorithme de Huffman
4. Donner le code binaire obtenu,
5. Calculer le quotient, le taux et le gain de compression, conclure.

Exercice 2

Supposons que nous ayons une image en niveaux de gris avec les nombres d'apparitions suivants :

Niveau de gris	128	150	200	220	254
Nbr. app	39	17	16	15	13

1. Construire l'arbre de Huffman correspondant.
2. Donner le code binaire obtenu,
3. Calculer le quotient, le taux et le gain de compression, conclure.

Compression sans perte (codage)

Codage de Huffman

Avantages

- Méthode de compression très **performante**.
- Très bon compromis **temps d'exécution/taux de compression**.
- Algorithme de compression le **plus utilisé**.

Formats utilisant cette méthode

- Images
 - ✓ PNG
 - ✓ JPEG (Joint Photographic Experts Group) – voir plus loin
- Fichiers compressés
 - ✓ Zip
 - ✓ Gzip

Compression sans perte (codage)

- ❖ RLE (Run-Length Encoding)
- ❖ Codage de Huffman
- ❖ Méthodes par dictionnaire

Compression sans perte (codage)

● Méthodes par dictionnaire: **LZW(Lempel-Ziv-Welch)**

Définition

LZW(Lempel-Ziv-Welch): Algorithme de compression qui **construit dynamiquement un dictionnaire à partir des données à compresser, pour remplacer les chaînes de caractères du dictionnaire par un nouveau code.**

Dictionnaire

Construction

- ✓ Initialisé avec les 256 valeurs de la table ASCII.
- ✓ Découpage des données en chaînes d'octets, comparées au dictionnaire et ajoutées si jamais elle n'y est pas présente.

Compression

- ✓ Parcours des données en les codant ; si jamais une chaîne est plus petite que le plus grand mot du dictionnaire, elle est transmise.

Décompression

- ✓ Le dictionnaire est reconstruit dans le sens inverse, et n'a donc pas besoin d'être stocké.

Compression sans perte (codage)

● Méthodes par dictionnaire: LZW(Lempel-Ziv-Welch)

Avantages

- Méthode efficace et facile à mettre en place.
- Plus la phrase est longue, plus les séquences de lettres ajoutées au dictionnaire sont grandes.
- L'algorithme gagne en efficacité avec de grandes données.

Inconvénients

- On rajoute un bit pour coder les symboles du dictionnaire tous les 2^n nouveaux symboles.
- Si il y a trop de symboles à rajouter ➔ l'algorithme n'a plus d'intérêt.

Formats d'image utilisant cette compression

- TIFF (Adobe).
- GIF .

PLAN

- Contexte & Intérêt de la compression*
- Compression sans perte (codage)*
- Compression avec pertes*
- Compression audio/vidéo*
- Conclusion*

Compression avec pertes

- ❖ DCT
- ❖ Ondelettes

Compression avec pertes

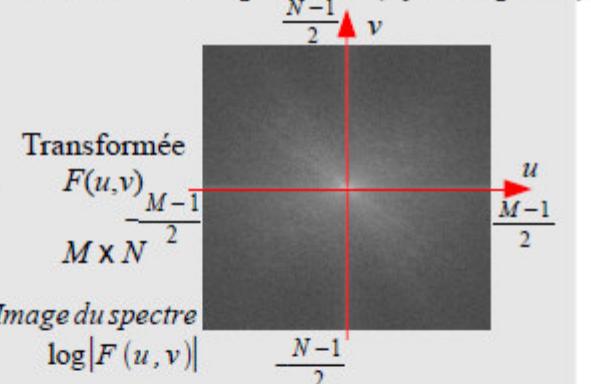
● DCT (Discret Cosine Transform): Rappel de TFD

Transformée de Fourier discrète 2D (ang. DFT) (rappel)

- Notations Domaine spatial (pixels)



- Domaine fréquentiel (cycles/pixel)



- DFT et DFT inverse

$$\begin{aligned}
 F(u, v) &:= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right)} \\
 &= \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cdot \left[\underbrace{\cos 2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right)}_{C_{m,n}^{M,N}(u, v)} - j \cdot \underbrace{\sin 2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right)}_{S_{m,n}^{M,N}(u, v)} \right] \\
 f(m, n) &:= \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{+j2\pi \left(\frac{mu}{M} + \frac{nv}{N} \right)}
 \end{aligned}$$

Compression avec pertes

➊ DCT (Discret Cosine Transform): Rappel de TFD

❖ Inconvénients de la TFD : sur un signal *f réel*,

- Produit un signal **F de spectre symétrique** ; seule la moitié des coefficients
- **Spectraux** a donc besoin d'être **calculée** ;
- Produit un **signal F complexe**, sans que sa partie réelle ou imaginaire seule
- Permette de représenter (donc de reconstruire) le signal f.

❖ DCT est une transformation spectrale (parmi d'autres) qui

- Possède les **mêmes propriétés que la DFT** ;
- S'applique uniquement sur les **signaux réels** ;
- Est définie par des fonctions de base en ***cosinus seulement*** ;
- Est utilisée en **compression d'images /vidéo(JPEG, MPEG)**.

Compression avec pertes

● DCT (Discret Cosine Transform):

Définition

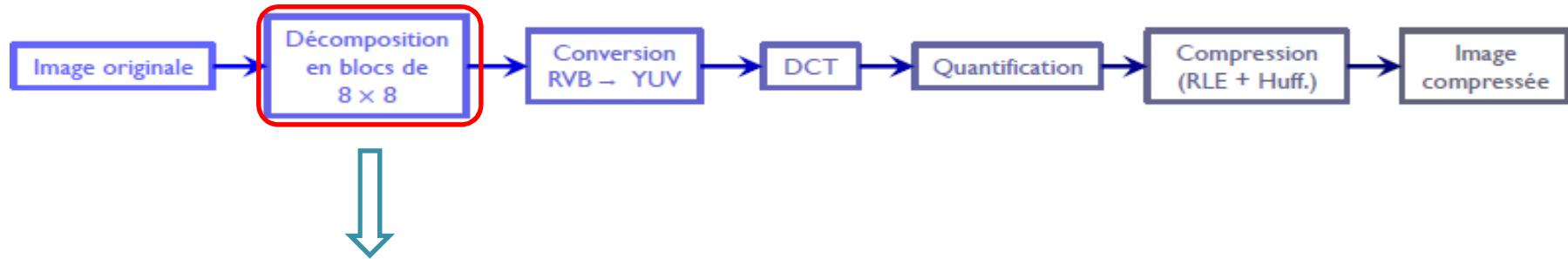
- Algorithme de compression basé sur la transformée DCT (*Discret Cosine Transform, en français transformée en cosinus discrète*), variante de la transformée de Fourier.
- Cette méthode permet de décrire chaque **bloc** en une carte de **fréquences** et en **amplitudes** plutôt qu'en **pixels** et **couleurs**.

Remarques

- Fréquence → **importance et rapidité d'un changement de couleur**
- Amplitude → **écart pour chaque changement de couleur**
- Utilisée pour **les formats JPEG et MPEG**.

Compression avec pertes

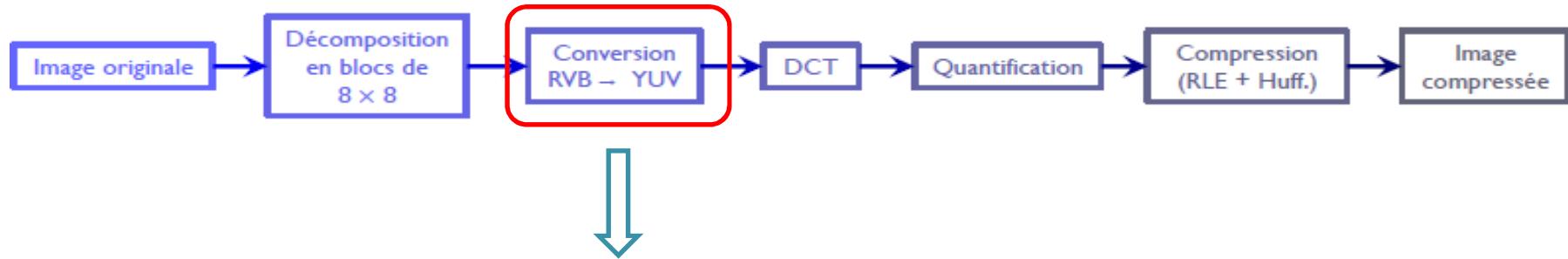
➊ DCT (Discret Cosine Transform): étapes de compression



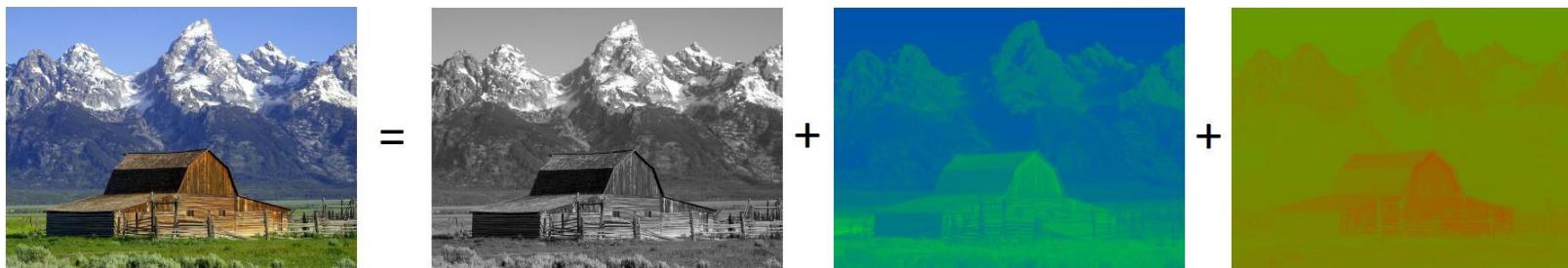
- L'image est divisée en **blocs de 8×8 pixels**
- Prolongement éventuel de l'image
 - Par des zéros.
 - Par symétrie.
 - ...
- Chaque bloc est une ‘petite’ image

Compression avec pertes

➊ DCT (Discret Cosine Transform): étapes de compression

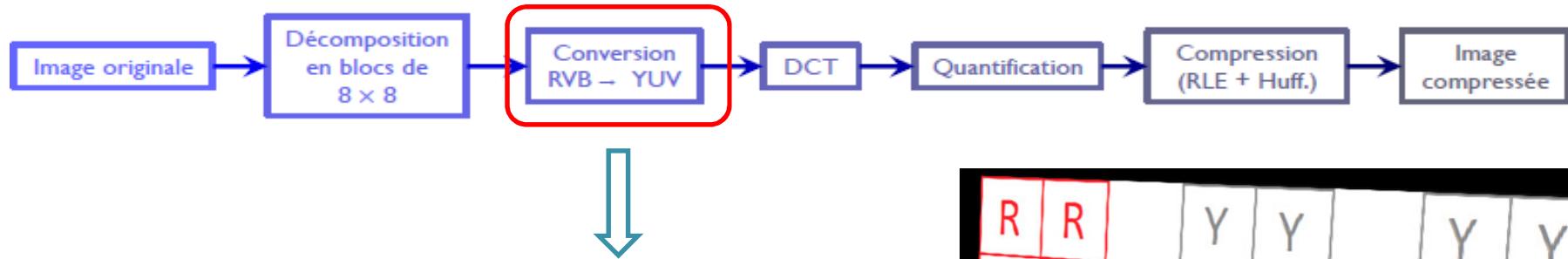


- Conversion des couleurs dans l'espace YUV
 - ✓ Y est une composante de **luminance**.
 - ✓ U et V sont deux **composantes de couleurs**.
 - ✓ Conversion réversible : $\text{RVB} \leftrightarrow \text{YUV}$
$$Y = 0,299 R + 0,587 V + 0,114 B$$
$$U = -0,1687 R - 0,3313 V + 0,5 B$$
$$V = 0,5 R - 0,4187 V - 0,0813 B$$



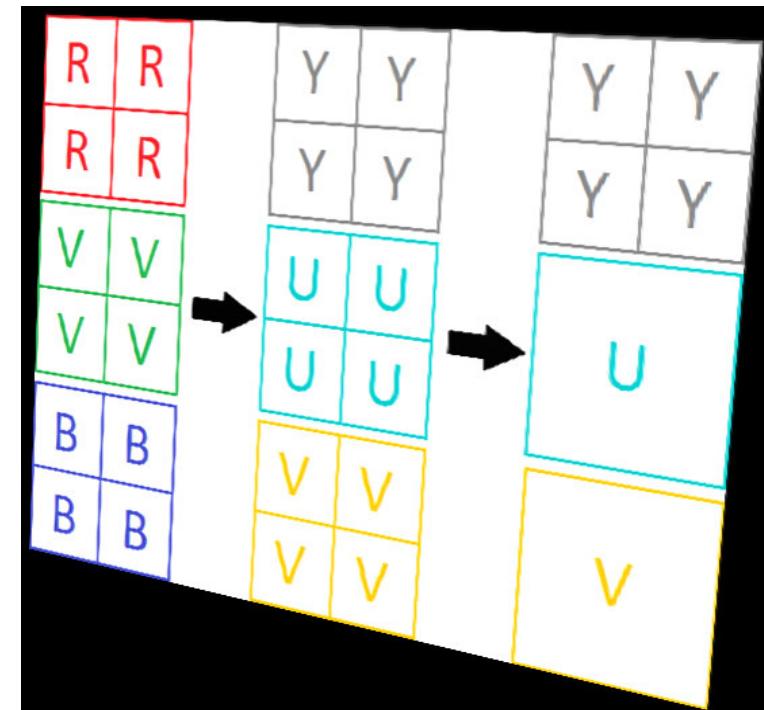
Compression avec pertes

● DCT (Discret Cosine Transform): étapes de compression



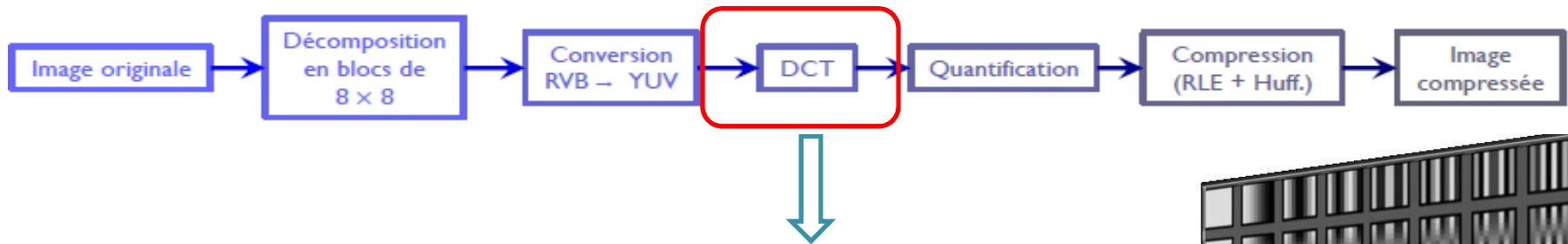
On exploite le fait que l'oeil humain est plus sensible à l'intensité lumineuse qu'aux informations colorées.

- ✓ On réduit les **composantes de chrominances par 4**.
- ✓ On réduit l'information de **chaque bloc par 2**.
- ✓ **Moins de valeurs : image plus légère.**
- ✓ **Perte d'informations : dégradation de l'image.**

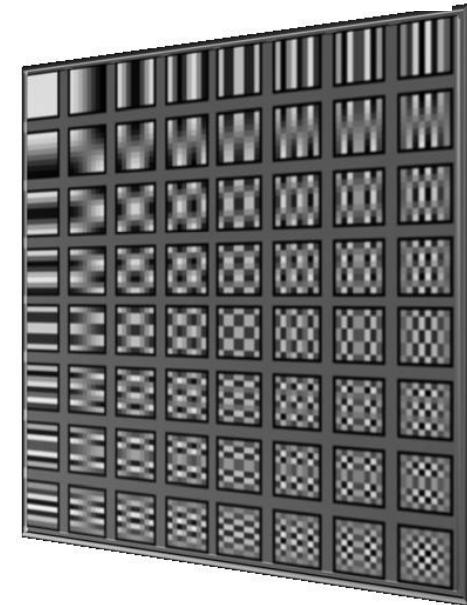


Compression avec pertes

➊ DCT (Discret Cosine Transform): étapes de compression

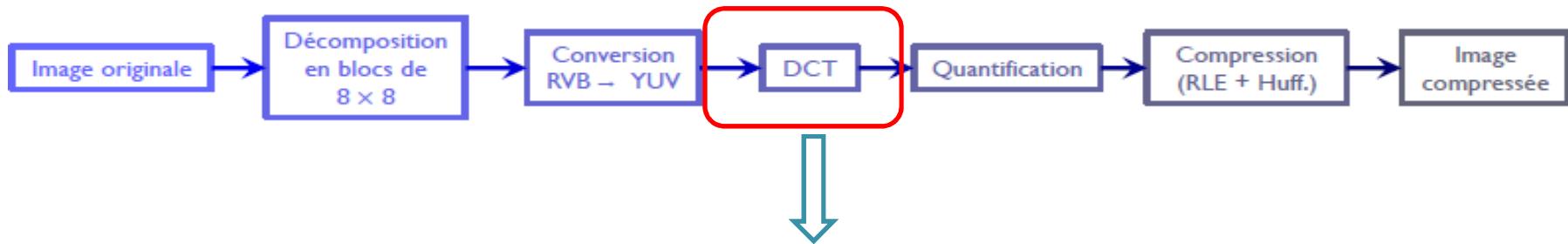


- C'est une représentation **fréquentielle**
 - ✓ Composante continue *au coin en haut à gauche*
 - ✓ Basses fréquences *en haut à gauche*
 - ✓ Hautes fréquences *en bas à droite*



Compression avec pertes

➊ DCT (Discret Cosine Transform): étapes de compression

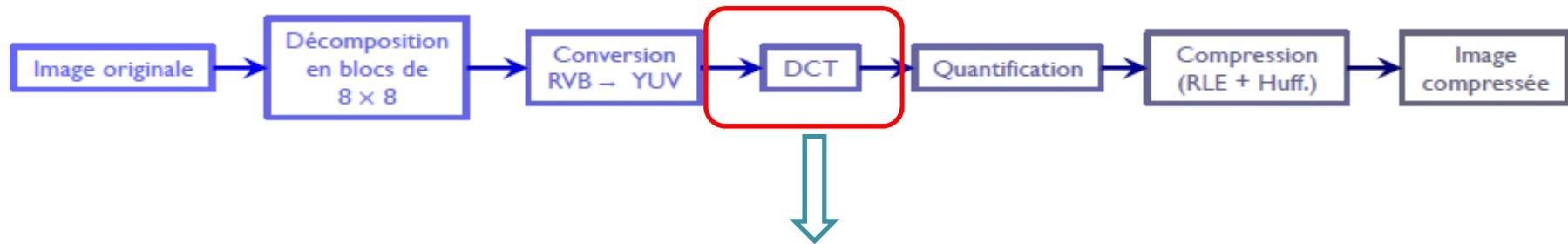


- On applique la DCT sur chaque bloc (matrice) de 8×8
 - ✓ Chaque **matrice** est transformée en une **autre matrice**.
 - ✓ Il n'y a pas de perte d'informations.
 - ✓ La **transformée inverse existe**.

$$DCT(i, j) = \frac{1}{\sqrt{2}} C(i)C(j) \sum_{x=0}^7 \sum_{y=0}^7 pixel(x, y) \cdot \cos \frac{(2x+1)i\pi}{16} \cdot \cos \frac{(2y+1)i\pi}{16}$$
$$\begin{cases} C(x) = \frac{1}{\sqrt{2}} & si \ x = 0 \\ C(x) = 1 & si \ x > 0 \end{cases}$$

Compression avec pertes

➊ DCT (Discret Cosine Transform): étapes de compression



➤ Chaque **sous-bloc en 8x8** est transformée en **une autre matrice par la DCT**

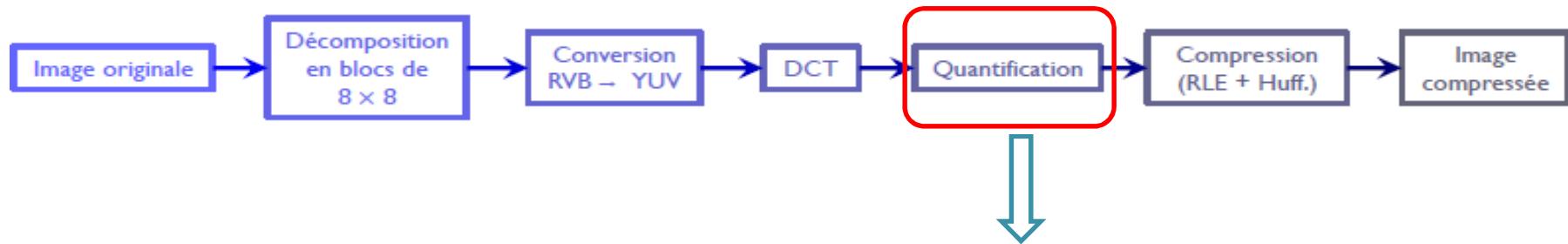
$$I_0 = \begin{matrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 160 & 161 & 162 & 162 & 155 & 155 & 155 & 161 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{matrix} \quad I_1 = \begin{matrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{matrix}$$

Matrice originale

Matrice transformée par DCT

Compression avec pertes

➊ DCT (Discret Cosine Transform): étapes de compression



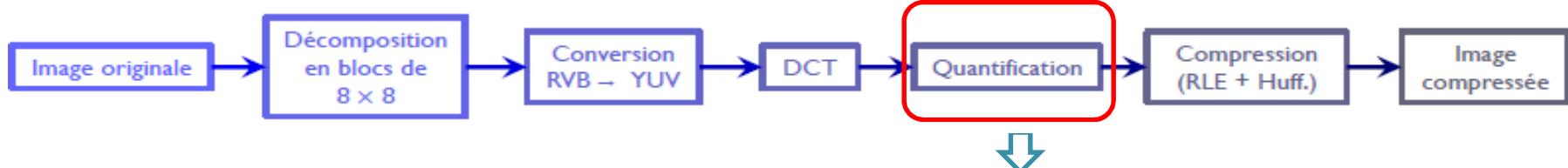
But

➤ Atténuer les hautes fréquences (auxquelles l'oeil humain est peu sensible) en mettant les coefficients à 0.

- ❖ Processus principalement responsable de la dégradation de l'image.
- ❖ Elle permet une compression (étape suivante) très efficace.
- ❖ C'est ici que la majorité de l'information est perdue.
- ❖ Elle est réversible mais induit des pertes.

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



➤ Grâce à un **facteur de qualité**, on crée une table de quantification 8x8 selon la formule : $Q(i,j) = 1 + (i + j + 1) \times q$

➤ Exemple pour un facteur de qualité de (q=3) :

$$I_1 = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Matrice transformée par DCT

$$Q = \begin{bmatrix} 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 \\ 7 & 10 & 13 & 16 & 19 & 22 & 25 & 28 \\ 10 & 13 & 16 & 19 & 22 & 25 & 28 & 31 \\ 13 & 16 & 19 & 22 & 25 & 28 & 31 & 34 \\ Q(2,4) = 1 + (2 + 4 + 1) \times 3 \\ 22 & 25 & 28 & 31 & 34 & 37 & 40 & 43 \\ 25 & 28 & 31 & 34 & 37 & 40 & 43 & 46 \end{bmatrix}$$

Matrice de quantification de qualité 3

$$I_2 = \begin{bmatrix} 315 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

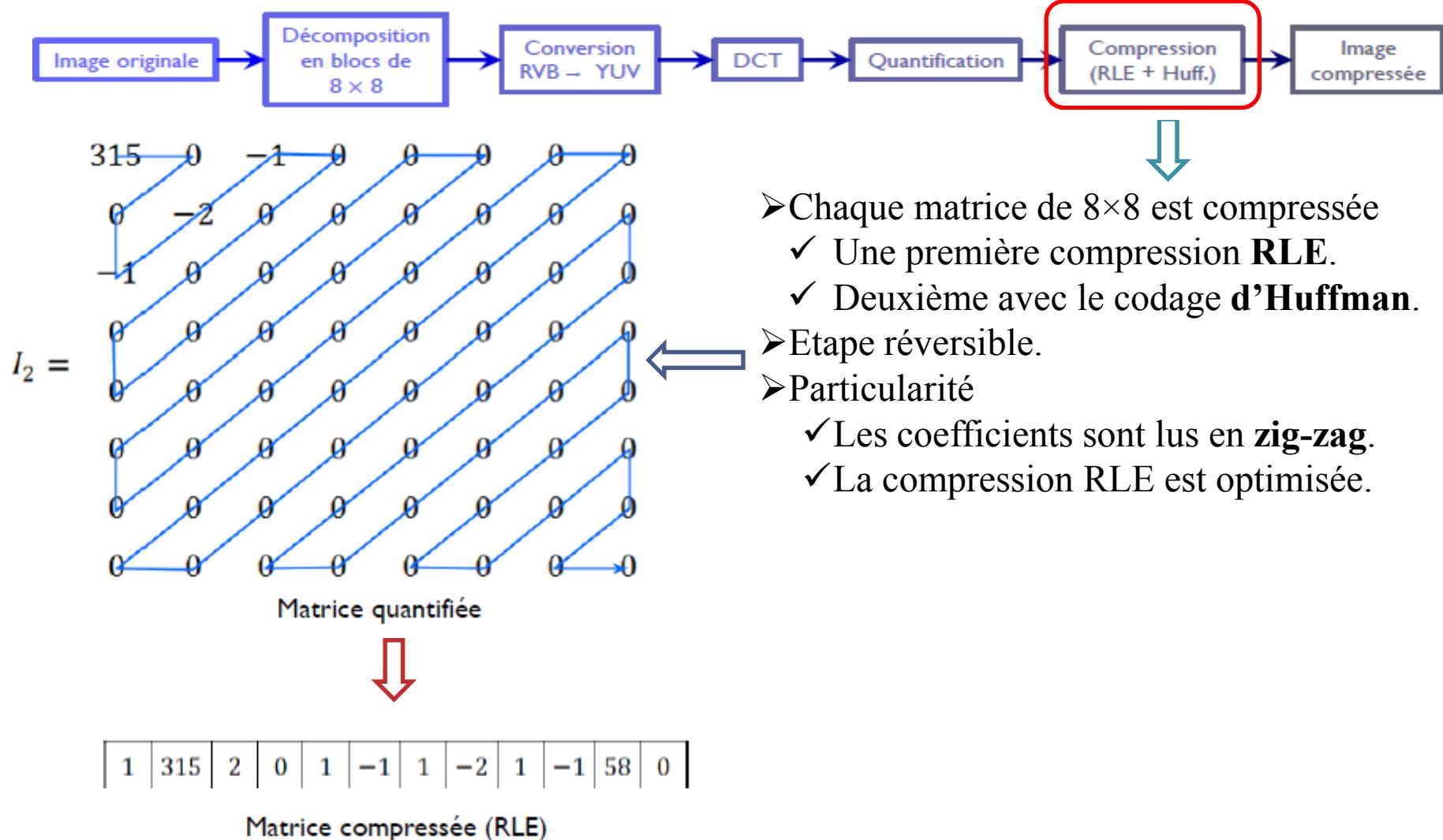
Matrice quantifiée

$$I_2(i,j) = \left\lfloor \frac{I_1(i,j)}{Q(i,j)} \right\rfloor$$

= La valeur d'un élément de la matrice DCT quantifiée sera **égale à l'arrondi, à l'entier le plus proche**, du quotient de la valeur correspondante de la matrice DCT par la valeur correspondante de la matrice de quantification

Compression avec pertes

DCT (Discret Cosine Transform): étapes de compression



Compression avec pertes

● DCT (Discret Cosine Transform): étapes de compression



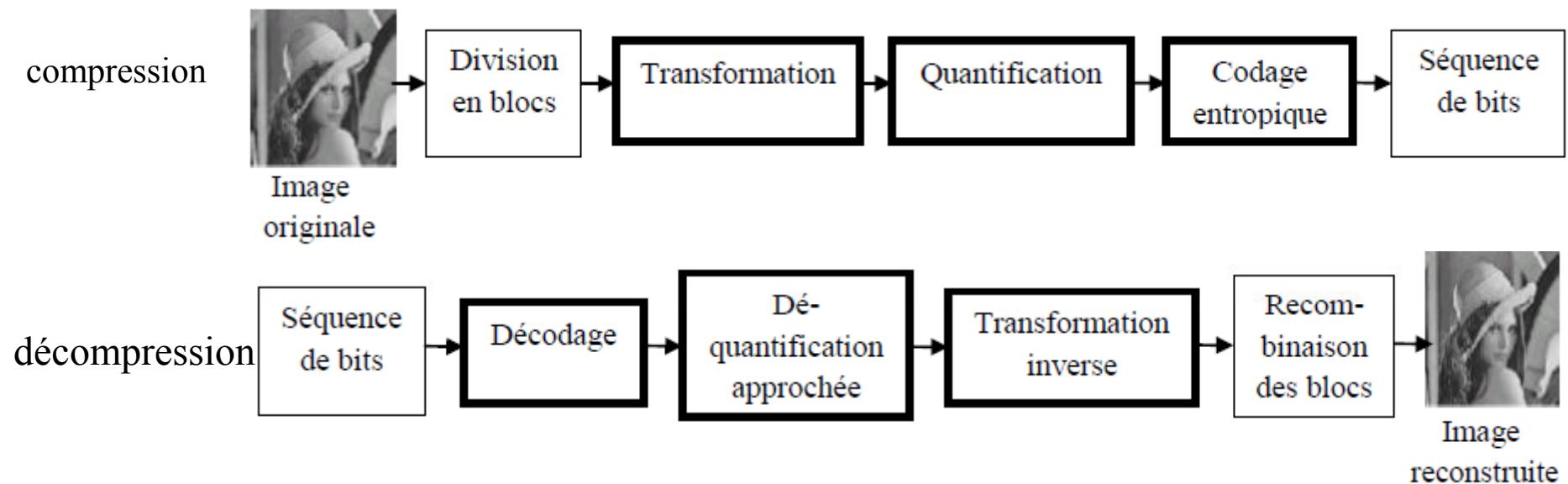
En bref

- Compression réalisée en **supprimant les hautes fréquences** (détails et contours).
 - ✓ **Perte** d'information peu visible.
 - ✓ **Bonne** compression.
- Formats d'image utilisant cette compression
 - ✓ **JPEG**.

Compression avec pertes

● DCT (Discret Cosine Transform): Exercice I

1. Donnez le schéma d'un système de compression/décompression JPEG .



2. Décrivez l'étape transformation (DCT) de ce système de compression

Elle consiste à appliquer une transformation mathématique à chaque bloc. Les transformations utilisées en compression d'image sont des transformations orthogonales, leur but est de décorrélérer les pixels, ce qui a pour effet en général de redistribuer l'énergie de l'image dans un nombre restreint des coefficients transformés.

Compression avec pertes

● DCT (Discret Cosine Transform): Exercice I

3. Quel effet apporte la DCT sur les coefficients de la matrice d'entrée

La DCT a pour effet de concentrer l'information en très peu de coefficients fréquentiels correspondant aux basses fréquences, et que les autres coefficients sont de haute fréquence.

Dans la matrice, suite à l'application de la DCT, les basses fréquences se trouvent en haut à gauche et les hautes fréquences en bas à droite. Les hautes fréquences représentent les zones à forts contrastes dans l'image,

4. Quel l'intérêt de la quantification dans ce système de compression

La quantification permet de mettre à zéro tous les coefficients inférieurs au quantum de la table, et donc de mettre à zéro une grande partie des hautes fréquences. Car la grande partie de l'information de l'image est stockée dans les basses fréquences après l'application du TCD

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 2

On cherche à coder l'image suivante en JPEG.



100	150	150	100
100	100	150	150
200	100	100	150
100	200	100	100

La matrice DCT obtenue est la matrice 4x4 ci-dessous :

$$\text{DCT}(i,j) = \begin{matrix} 512 & 7 & -13 & -16 \\ -7 & -39 & -20 & 44 \\ -13 & 20 & -87 & -49 \\ 16 & 44 & 49 & 14 \end{matrix}$$

On utilise la matrice de quantification définie par $Q = [q_{i,j}]$ avec $q_{i,j} = 1 + \kappa(1 + i + j)$, $i \in [0, N-1]$, $j \in [0, N-1]$, N la taille du bloc

1. Ecrire la matrice de quantification pour les facteurs de qualité $k=9$ et $k=20$.
2. Calculer les valeurs de la matrice DCT quantifiée pour les deux cas.
3. Donner la suite générée après la lecture zigzag pour les deux cas.
4. Donnez le codage RLE pour cette image dans les deux cas. Conclure.

Compression avec pertes

● DCT (Discret Cosine Transform): Exercice 3

On cherche à coder une image (4x4) en JPEG. On a obtenu la matrice DCT avec les valeurs indiquées dans la matrice 4x4 ci-dessous.

1. Donner la matrice de quantification classique de facteur $\kappa = 20$.
2. Donner la matrice-résultat après quantification.
3. Donner la suite générée après la lecture zigzag.
4. Donner son codage en Run Length Encoding.

DCT(i,j)=

1120	82	-48	36
56	8	32	4
-208	-40	22	-12
30	8	-8	2

Compression avec pertes

DCT (Discret Cosine Transform): Exercice 4

On cherche à coder l'image suivante en JPEG.



Matrice source (Non signée) :							
29	95	60	60	64	52	61	63
123	85	119	114	118	103	95	103
65	79	82	106	115	95	97	73
71	79	99	82	119	122	96	106
106	86	90	148	132	132	116	162
93	99	79	103	123	113	115	117
141	155	88	102	108	95	118	103
134	160	93	104	97	102	105	128

La matrice DCT obtenue est la matrice 4x4 ci-dessous :

Matrice DCT :							
-214	-23	-4	8	18	-27	-24	-12
-103	-15	-52	-16	-8	8	19	19
-40	66	41	-11	-24	-13	-30	-6
-20	-19	4	-7	8	-3	-9	-7
-21	-19	9	-20	-4	-20	7	-17
-68	11	0	6	-37	-18	-23	8
-58	-19	-17	-4	-13	-10	-18	-5
4	1	-2	-6	13	8	-21	-17

On utilise la matrice de quantification définie par $Q = [q_{i,j}]$ avec $q_{i,j} = 1 + \kappa(1 + i + j)$, $i \in [0, N-1]$, $j \in [0, N-1]$, N la taille du bloc

1. Ecrire la matrice de quantification pour les facteurs de qualité k=2 et k=20.
2. Calculer les valeurs de la matrice DCT quantifiée pour les deux cas.
3. Donner la suite générée après la lecture zigzag pour les deux cas.
4. Donnez le codage RLE pour cette image dans les deux cas. Conclure.
5. Donnez le codage de Huffman.

Compression avec pertes

● DCT (Discret Cosine Transform): Exercice 3 (solution)

Matrice de quantification :

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Matrice DCT quantifiée :

-71	-4	0	0	1	-2	-1	0
-20	-2	-5	-1	0	0	1	1
-5	7	3	0	-1	0	-1	0
-2	-1	0	0	0	0	0	0
-1	-1	0	-1	0	0	0	0
-5	0	0	0	-1	0	0	0
-3	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Développement en zig-zag :

-71 -4 -20 -5 -2 0 0 -5 7 -2 -1 -1 3 -1 1 -2 0 0 0 -1 -5 -3 0 0 0 -1 0 -1
0 -1 0 -1 0 0 0 0 0 -1 1 0 0 0 -1 0

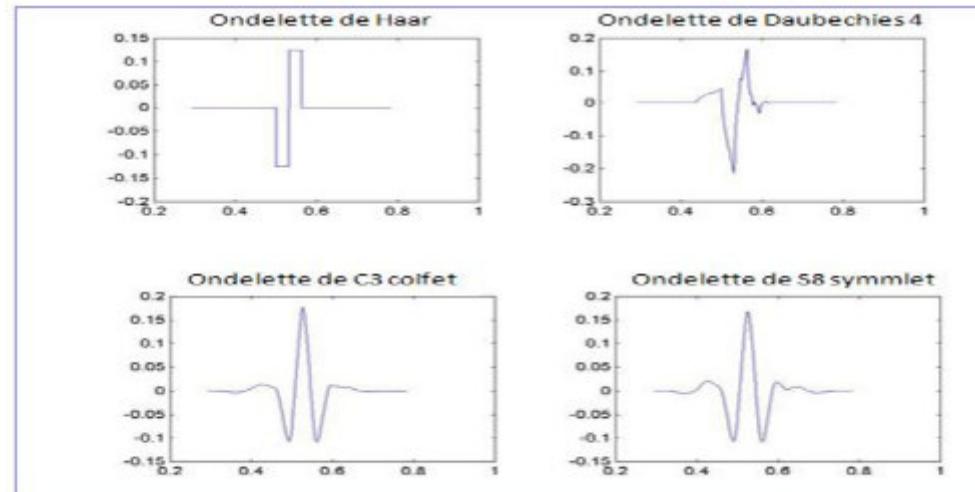
Compression avec pertes

- ❖ DCT
- ❖ Ondelettes

Compression avec pertes

● Transformée en ondelettes: Définition

- Une **ondelette** est une **forme d'onde** qui a une **valeur moyenne zéro** et une **durée limitée**.
- Transformée en ondelettes (**Wavelet Transform**); est un **outil mathématique** qui **décompose** un signal en **fréquences** en conservant une localisation spatiale. **Le signal de départ est projeté sur un ensemble de fonctions de base** qui varient en fréquence et en espace.



Quelques familles
d'ondelettes

- Séparer de l'**information général** (basses fréquences) et les **détails** (hautes fréquences), contenue dans une image, un son, ...
- Utilisée pour les formats **JPEG2000 et MPEG2000**.

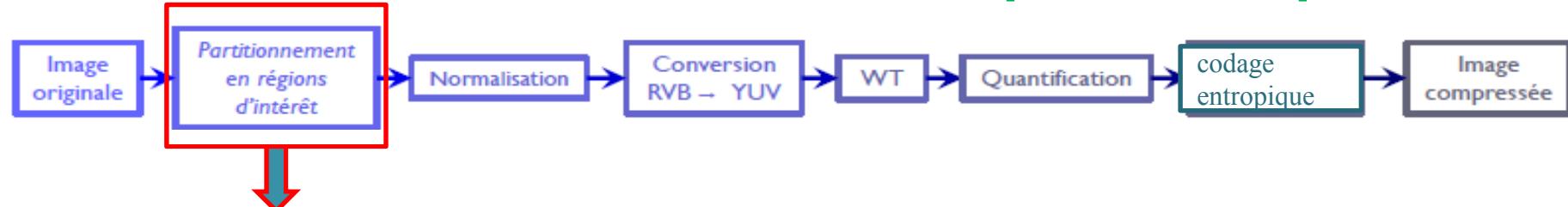
Compression avec pertes

Transformée en ondelettes: domaines d'application

1. Compression des données de haute résolution (e.g. images, vidéo)
2. Filtrage (filtering, de-noising)
3. Lissage (smoothing)
4. Extraction des propriétés caractéristiques (Feature Extraction)
5. Détection des discontinuités (discontinuity detection)
6. Analyse des données (data analysis) (par exemple, biomédicales, financières)
7. Télécommunication (par exemple, codage de sources et canaux (Source and Channel Coding))
8. Astronomie (par exemple, distances dans l'univers, galaxies formant de structures hiérarchiques à différent niveaux de l'échelle).
9. Analyse de séries temporelles pour des prévisions de marché boursier.
10. Réseaux d'ondelette (wavelet networks) : apprentissage en temps réel des fonctions inconnues.
11.

Compression avec pertes

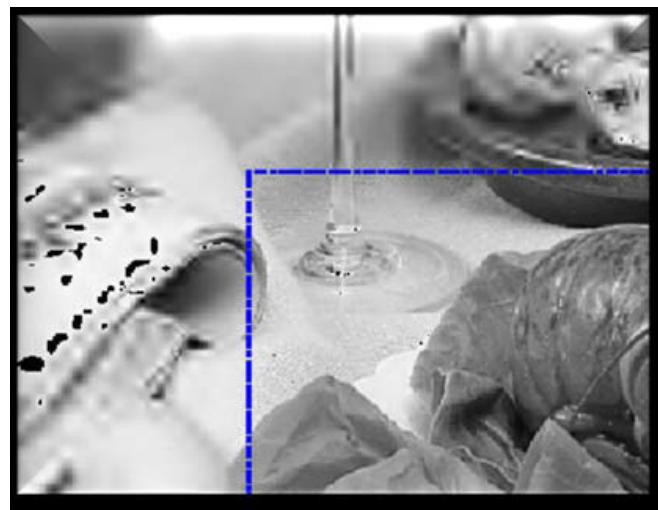
Transformée en ondelettes discrètes : étapes de compression



➤ Etape facultative

➤ On y définit des **zones (régions d'intérêt ou ROI)**

- ✓ La qualité de ces zones est préservée.
- ✓ Ces régions sont **mieux encodées**, au détriment du reste de l'image.



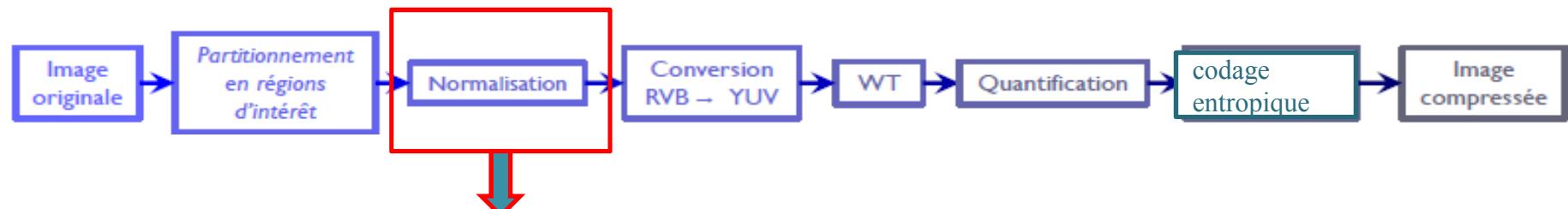
Avec ROI



Sans ROI

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression

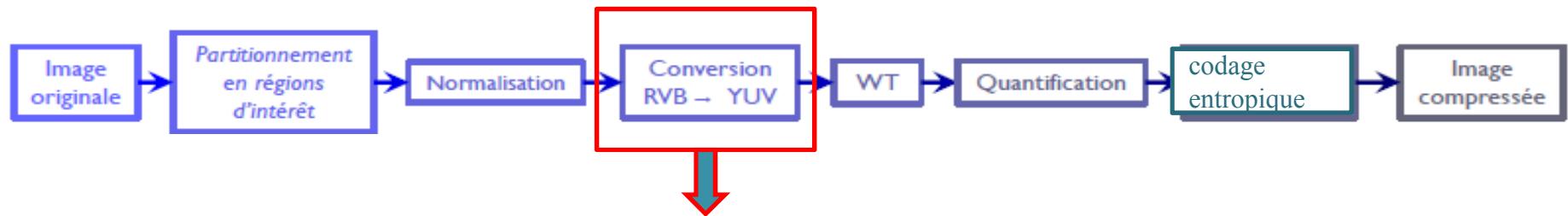


Décalage des valeurs de l'intervalle [0;255] vers [-128;127]

- But : **centrer les valeurs autour de 0.**
- Etape nécessaire à une **bonne compression**

Compression avec pertes

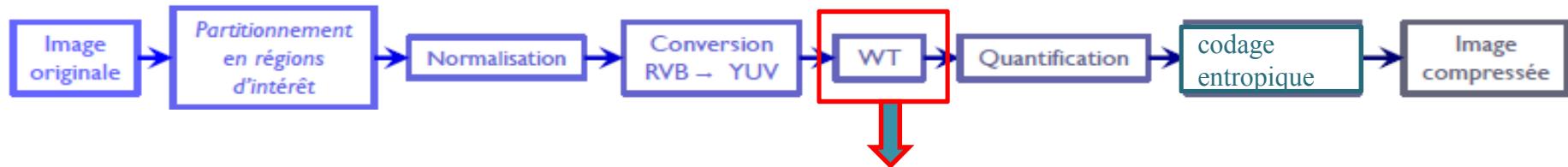
Transformée en ondelettes discrètes : étapes de compression



- Même conversion que pour la compression utilisant la transformée en cosinus discrètes.
- Synthèse de couleurs mieux adaptée à la compression car les composantes sont moins corrélées.

Compression avec pertes

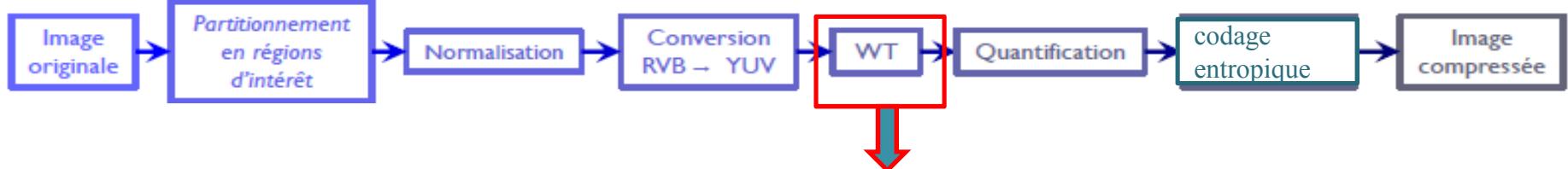
Transformée en ondelettes discrètes : étapes de compression



- N'importe quelle **transformée en ondelettes** peut être utilisée
- Les plus utilisées
 - ✓ Ondelette de **Daubechies**.
 - ✓ Ondelette de **Le Gall**.
- Exemple des ondelettes les plus simples : **ondelettes de Haar**
 - ✓ Permet de décomposer une séquence en basses et hautes fréquences

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



→ Exemple: Interprétation (transformée en ondelettes 1D de Haar)

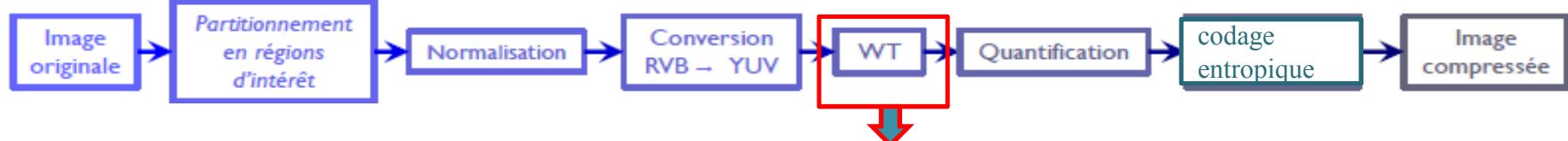
1. On calcule d'abord les moyennes 2 à 2, Ces termes représentent de manière grossière (moyenne) de la séquence
2. On calcule ensuite les différences 2 à 2, Ces termes de différences sont aussi appelés les détails de la séquence.
3. On réitère ces 2 opérations sur les coefficients de moyenne de la nouvelle séquence jusqu' on trouve une seule valeur moyenne.

Niveau de détail #	Coefficients de moyenne	Coefficients de détail
0	[9 ; 7 ; 3 ; 5]	
1	$L_1 = [8 = (9+7)/2 ; 4 = (3+5)/2]$	$H_1 = [1 = (9-7)/2 ; -1 = (3-5)/2]$
2	$L_2 = [6 = (8+4)/2]$	$H_2 = [2 = (8-4)/2]$

- L : filtre passe-bas (*Low pass filter*), H : filtre passe-haut (*High pass filter*)
- Séquence originale : [9 ; 7 ; 3 ; 5], Séquence finale : [6 ; 2 ; 1 ; -1]

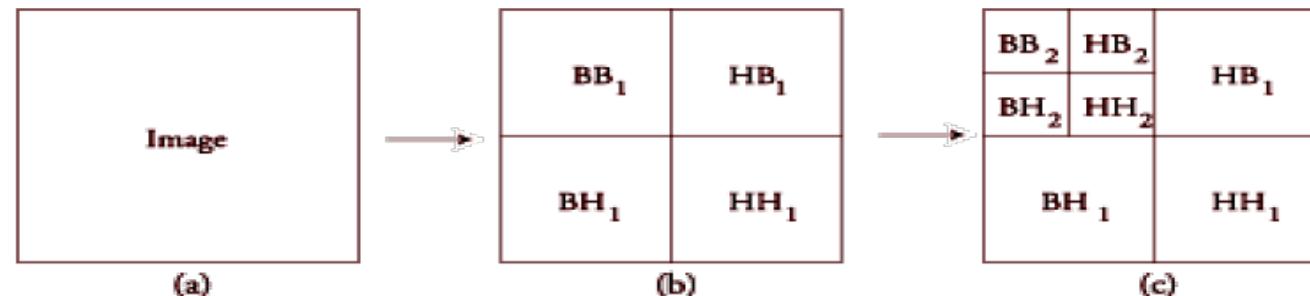
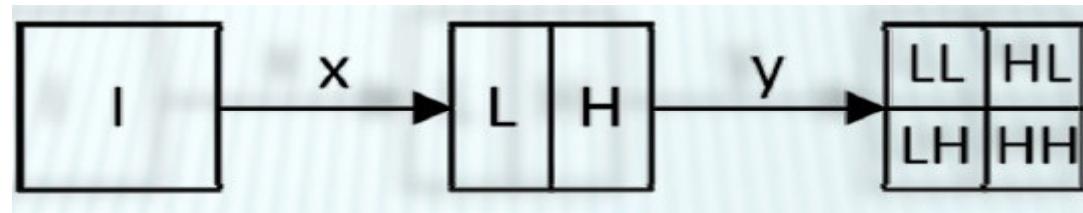
Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



→ Exemple: Interprétation (Transformée en ondelettes discrètes 2D de Haar)

➤ Construction par succession d'ondelettes 1D discrètes suivant les axes x, puis y de l'image 2D

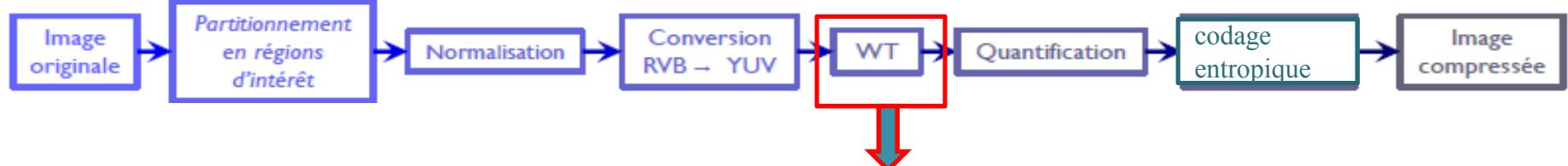


4 niveaux de détail de la décomposition en ondelettes 2D

Pour chaque niveau de détail, l'information générale est en haut à gauche, et les détails dans la zone restante

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



→ Exemple: Interprétation (Transformée en ondelettes discrètes 2D de Haar)

I2	I2	I2	I2	I2	I4	I2	I2	I2
I2	I2	I2	I2	I2	I4	I2	I2	I2
I2	I2	I2	I2	I2	I4	I2	I2	I2
I2	I2	I2	I2	I2	I4	I2	I2	I2
I2	I2	I2	I2	I2	I4	I2	I2	I2
16	16	16	16	14	16	16	16	16
I2	I2	I2	I2	I2	I4	I2	I2	I2
I2	I2	I2	I2	I2	I4	I2	I2	I2

(ligne)

I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
16	16	15	16	0	0	-2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0

Basses fréquences

I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I4	I4	I4	I4	0	0	0	0
I2	I2	I3	I2	0	0	2	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-4	-4	-2	-4	0	0	4	0
0	0	0	0	0	0	0	0

(colonne)

I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
16	16	15	16	0	0	-2	0
I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0

I2	I2	I3	I2	0	0	2	0
I2	I2	I3	I2	0	0	2	0
I4	I4	I4	I4	0	0	0	0
I2	I2	I3	I2	0	0	2	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-4	-4	-2	-4	0	0	4	0
0	0	0	0	0	0	0	0

Reste = hautes fréquences, qui pourront être fortement quantifiées

Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



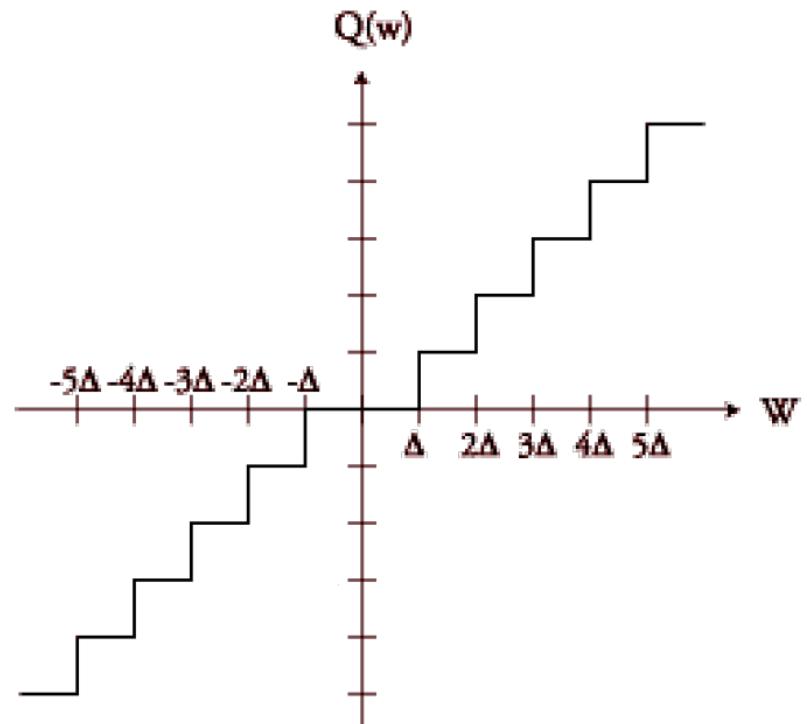
➤ Quantification **uniforme à zone morte**.

- ✓ La **courbe** évolue de manière **uniforme**.
- ✓ Les valeurs quantifiées peuvent être égales à zéro.

➤ Deux caractéristiques

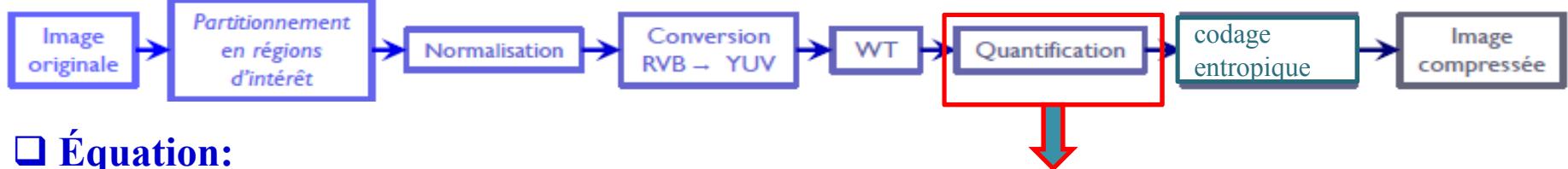
- ✓ Le **pas Δ** : l'intervalle pour lequel on a une même valeur quantifiée.
- ✓ La **largeur de la zone morte Z** : l'intervalle pour lequel la valeur quantifiée est **nulle**.

Attention : la quantification ne s'applique qu'aux coefficients de détail, et non pas au coefficient de moyenne.



Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression

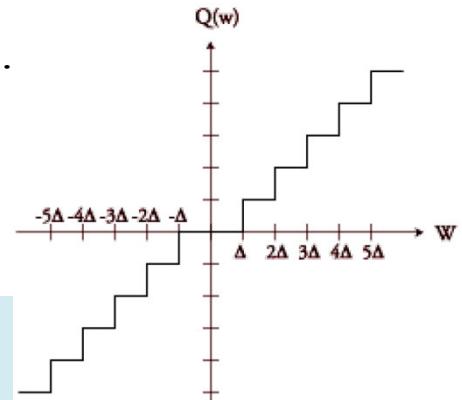


□ Équation:

- Étant donné que l'on a une zone morte de largeur Z , cela signifie **que si un coefficient de détail C , en valeur absolue, est dans la zone morte, c'est-à-dire $|C| < Z$, alors le coefficient est mis à 0.**
- Sinon, si un coefficient de détail C , en valeur absolue, est hors de la zone morte, c'est-à-dire $|C| \geq Z$, il sera quantifié selon le pas de quantification Δ choisi.

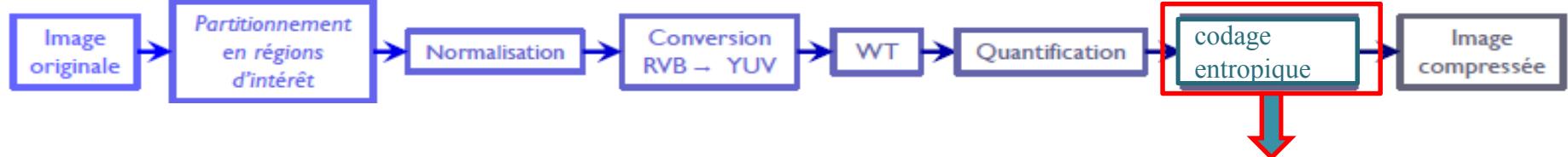
□ L'équation de quantification finale est donc :

- Si $|C| < Z$, alors $C_{\text{résultant}} = 0$
- Si $|C| \geq Z$, alors $C_{\text{résultant}} = \text{signe}(C) \times \Delta \times \text{partie_entière}(|C| / \Delta)$



Compression avec pertes

Transformée en ondelettes discrètes : étapes de compression



➤ Coder le résultat obtenu à l'aide de Huffman

Compression avec pertes

● Transformée en ondelettes discrètes : Exercice I

- Soit une séquence d'origine $S_0 = [2 \ 4 \ 8 \ 12 \ 14 \ 0 \ 2 \ 1]$, correspondant aux données initiales. Donner la décomposition en ondelettes 1D de Haar de la séquence S_0 .

le tableau de la décomposition en ondelettes 1D de Haar de S_0 est le suivant:

Niveau de résolution n	Coefficients de moyenne S_n	Coefficients de détail D_n
0 – séquence d'origine	$S_0 = [2 \ 4 \ 8 \ 12 \ 14 \ 0 \ 2 \ 1]$	-----
1	$S_1 = [3 \ 10 \ 7 \ 1.5]$	$D_1 = [-1 \ -2 \ 7 \ 0.5]$
2	$S_2 = [6.5 \ 4.25]$	$D_2 = [-3.5 \ 2.75]$
3	$S_3 = [5.375]$	$D_3 = [1.125]$

La séquence finale obtenue après décomposition en ondelettes 1D de Haar est de même taille que la séquence initiale, et les coefficients correspondent au dernier coefficient de moyenne (soit S_3) suivi des coefficients de détail des suites D_n du niveau le moins détaillé ($n = 3$) et niveau le plus détaillé ($n = 1$).
Donc : $F = [5.375 \ 1.125 \ -3.5 \ 2.75 \ -1 \ -2 \ 7 \ 0.5]$

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2

Soit l'image I suivante:

$$I = \begin{bmatrix} 0 & 250 & 25 & 50 & 200 & 0 & 0 & 0 \\ 50 & 50 & 50 & 25 & 50 & 0 & 25 & 0 \\ 25 & 50 & 0 & 250 & 0 & 50 & 0 & 0 \\ 75 & 200 & 200 & 0 & 250 & 250 & 0 & 200 \\ 250 & 25 & 250 & 200 & 0 & 75 & 25 & 25 \\ 50 & 0 & 50 & 0 & 0 & 75 & 250 & 250 \\ 250 & 250 & 25 & 250 & 50 & 25 & 50 & 50 \\ 250 & 200 & 50 & 50 & 50 & 25 & 0 & 200 \end{bmatrix}$$

1. Donner la décomposition en ondelettes 1D de Haar de l'image I
2. Donner la matrice de quantification à l'aide d'une quantification uniforme à zone morte de largeur 30 et de pas de quantification 10.
3. Coder le résultat obtenu à l'aide de Huffman
4. Calculer le quotient, le taux et le gain de compression. Conclure

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (solution)

Soit l'image I suivante:

$$I = \begin{bmatrix} 0 & 250 & 25 & 50 & 200 & 0 & 0 & 0 \\ 50 & 50 & 50 & 25 & 50 & 0 & 25 & 0 \\ 25 & 50 & 0 & 250 & 0 & 50 & 0 & 0 \\ 75 & 200 & 200 & 0 & 250 & 250 & 0 & 200 \\ 250 & 25 & 250 & 200 & 0 & 75 & 25 & 25 \\ 50 & 0 & 50 & 0 & 0 & 75 & 250 & 250 \\ 250 & 250 & 25 & 250 & 50 & 25 & 50 & 50 \\ 250 & 200 & 50 & 50 & 50 & 25 & 0 & 200 \end{bmatrix}$$

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

a. Suivant la largeur :

125	37.5	100	0	-125	-12.5	100	0
50	37.5	25	12.5	0	12.5	25	12.5
37.5	125	25	0	-12.5	-125	-25	0
137.5	100	250	100	-62.5	100	0	-100
137.5	225	37.5	25	112.5	25	-37.5	0
25	25	37.5	25	25	25	-37.5	0
250	137.5	37.5	50	0	-112.5	12.5	0
225	50	37.5	100	25	0	12.5	-100

a. Suivant la hauteur:

-87.5	37.5	62.5	6.25	-62.5	0	62.5	6.25
-87.5	112.5	137.5	50	-37.5	-12.5	-12.5	-50
-81.25	125	37.5	25	68.75	25	-37.5	0
-237.5	93.75	37.5	75	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (solution)

1. Donner la décomposition en ondelettes 1D de Haar de l'image I

-87.5	37.5	62.5	6.25	-62.5	0	62.5	6.25
-87.5	112.5	137.5	50	-37.5	-12.5	-12.5	-50
-81.25	125	37.5	25	68.75	25	-37.5	0
-237.5	93.75	37.5	75	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

a. Suyant la largeur:

62.5	34.375	25	28.125	-62.5	0	62.5	6.25
100	93.75	-12.5	43.75	-37.5	-12.5	-12.5	-50
103.125	31.25	-21.875	6.25	68.75	25	-37.5	0
165.625	56.25	71.875	-18.75	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

81.25	64.0625	6.25	35.9375	-62.5	0	62.5	6.25
134.375	43.75	25	-6.25	-37.5	-12.5	-12.5	-50
-18.75	29.6875	18.75	-7.8125	68.75	25	-37.5	0
-31.25	-12.5	-46.875	12.5	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

b. Suyant la hauteur:

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (solution)

- Donner la décomposition en ondelettes 1D de Haar de l'image I

81.25	64.0625	6.25	35.9375	-62.5	0	62.5	6.25
134.375	43.75	25	-6.25	-37.5	-12.5	-12.5	-50
-18.75	29.6875	18.75	-7.8125	68.75	25	-37.5	0
-31.25	-12.5	-46.875	12.5	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

a. Suivant la largeur:

72.65625	8.59375	6.25	35.9375	-62.5	0	62.5	6.25
89.0625	45.3125	25	-6.25	-37.5	-12.5	-12.5	-50
-18.75	29.6875	18.75	-7.8125	68.75	25	-37.5	0
-31.25	-12.5	-46.875	12.5	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

80.959375	26.953125	6.25	35.9375	-62.5	0	62.5	6.25
-8.103125	-18.359375	25	-6.25	-37.5	-12.5	-12.5	-50
-18.75	29.6875	18.75	-7.8125	68.75	25	-37.5	0
-31.25	-12.5	-46.875	12.5	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

b. Suivant la hauteur:



Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (Solution)

2. Donner la matrice de quantification à l'aide d'une quantification uniforme à zone morte de largeur $z=30$ et de pas de quantification $\Delta=10$.

80.959375	26.953125	6.25	35.9375	-62.5	0	62.5	6.25
-8.103125	-18.359375	25	-6.25	-37.5	-12.5	-12.5	-50
-18.75	29.6875	18.75	-7.8125	68.75	25	-37.5	0
-31.25	-12.5	-46.875	12.5	12.5	-56.25	12.5	-50
37.5	0	62.5	6.25	-62.5	-12.5	37.5	-6.25
-50	12.5	-112.5	-50	25	-112.5	-12.5	50
56.25	100	0	0	43.75	0	0	0
12.5	43.75	0	-25	-12.5	-56.25	0	50

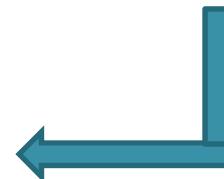
L'application de l'équation de quantification ci-dessous aux coefficients de détail de la matrice précédente

- Si $|C| < Z$, alors $C_{\text{résultant}} = 0$
- Si $|C| \geq Z$, alors $C_{\text{résultant}} = \text{signe}(C) \times \Delta \times \text{partie_entière}(|C|/\Delta)$

Avec

- C : coefficient de détail
- Z : largeur de la zone morte
- Δ : le pas de quantification

80.959375	0	0	30	-60	0	60	0
0	0	0	0	-30	0	0	-50
0	0	0	0	60	0	-30	0
-30	0	-40	0	0	-50	0	-50
30	0	60	0	-60	0	30	0
-50	0	-110	-50	0	0	0	50
50	100	0	0	40	0	0	0
0	40	0	0	0	-50	0	50



Compression avec pertes

● Transformée en ondelettes discrètes : Exercice 2 (Solution)

- Coder le résultat obtenu à l'aide de l'arbre de Huffman

80.959375	0	0	30	-60	0	60	0
0	0	0	0	-30	0	0	-50
0	0	0	0	60	0	-30	0
-30	0	-40	0	0	-50	0	-50
30	0	60	0	-60	0	30	0
-50	0	-110	-50	0	0	0	50
50	100	0	0	40	0	0	0
0	40	0	0	0	-50	0	50



Sur l'image quantifiée ci-dessus , la compression HUFFMAN donne les fréquences d'apparition:

Symboles	Nb Occurr.
0	38
-50	6
30	3
60	3

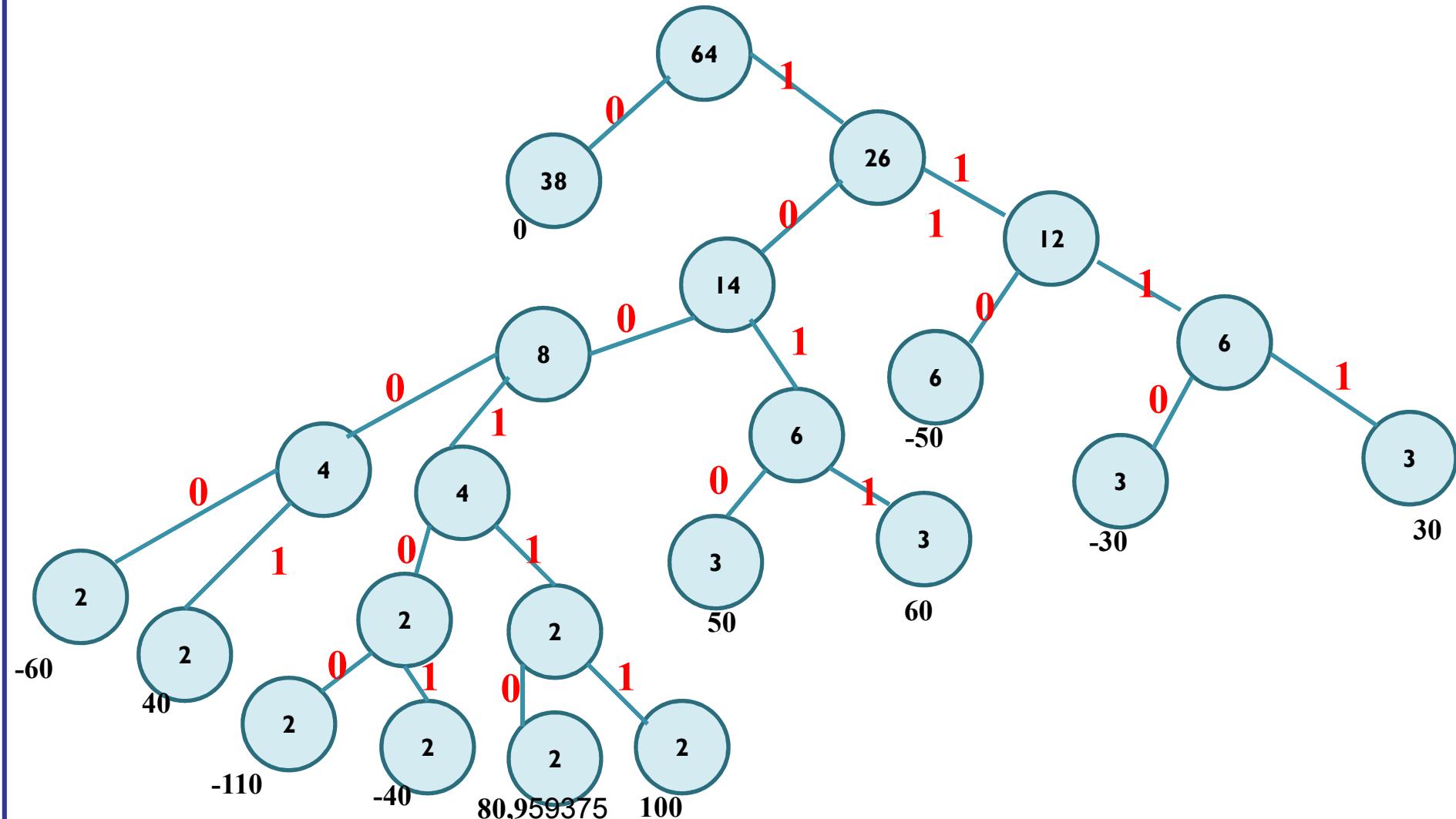
Symboles	Nb Occurr.
-30	3
50	3
-60	2
40	2

Symboles	Nb Occurr.
80.959375	1
-40	1
-110	1
100	1

Compression avec pertes

Transformée en ondelettes discrètes : Exercice 2 (Solution)

3. Coder le résultat obtenu à l'aide de l'arbre de Huffman



Compression avec pertes

● Transformée en ondelettes discrètes : Exercice 2 (Solution)

3. Coder le résultat obtenu à l'aide de l'arbre de Huffman

Symboles	Code binaire
0	0
-50	110
-30	1110
30	1111

Symboles	Code binaire
50	1010
60	1011
-60	10000
40	10001

Symboles	Code binaire
-110	100100
-40	100101
80,959375	100110
1000	100111

4. Calculer le quotient, le taux et le gain de compression. Conclure

- La taille initiale de l'image est égale à

$$\text{Taille_initiale} = L \cdot H \cdot \text{poids}_{\text{pixel}} = 8 \cdot 8 \cdot 8 = 512 \text{ bits}$$

- La taille finale de l'image compressée est égale à :

$$\text{Taille_finale} = (38 \cdot 1 + 6 \cdot 3 + 3 \cdot 4 + 3 \cdot 4 + 3 \cdot 4 + 3 \cdot 4 + 2 \cdot 5 + 2 \cdot 5 + 1 \cdot 6 + 1 \cdot 6 + 1 \cdot 6) = 148 \text{ bits}$$

- $\text{quotient} = \text{Taille_initiale} / \text{Taille_finale} = 3,4594$
- $\text{taux} = 1 / \text{quotient} = 0,2890$
- $\text{gain} = (1 - \text{taux}) * 100\% = 71,1\%$

Comparatif des méthodes de compression

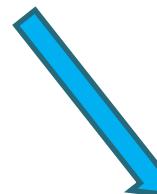
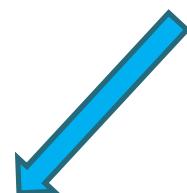
- Sans compression
 - ✓ Format très lourd.
 - ✓ Exemples : BMP (pour des images d'une profondeur de pixel différente de 4 ou 8 bits).
- Compression sans perte
 - ✓ RLE, LZW, Huffman : basé sur la redondance des valeurs.
 - ✓ Exemples : PCX, GIF, PNG.
- Compression avec pertes
 - ✓ Décomposition des images (détails / formes générales) et élimination de certaines parties de ces informations pour mieux compresser.
 - ✓ Compression par DCT (JPEG)
 - Taux de compression réglable.
 - ✓ Compression par ondelettes (JPEG2000)
 - Meilleur rapport compression/qualité d'image.
 - Compression sans pertes possibles (moins bonne compression).

Conclusion

A retenir pour ce cours:

Compression et stockage

- Position du problème
- Définition & Intérêts de la compression
- Types de compression
- Evaluation de la compression et des pertes



Compression sans perte
RLE, Huffman, LZW



Compression avec pertes
DCT, Ondelettes

Références

➤ Livres

- ✓ *Compression d'image – Algorithmes et standards*, Éric Incerti, Vuibert 2003
- ✓ *Introduction au traitement d'images – Simulation sous Matlab*, Gilles Burel, Hermès 2001 (chapitre 8)
- ✓ *Comprendre la vidéo numérique*, Jean-Charles Fouché, 2ème édition.
- ✓ *Digital Video and Television*, Ioannis Pitas.
- ✓ *ACQUISITION et TRAITEMENT D'IMAGE NUMERIQUE* - Université Paul Sabatier IUT - Département de Mesures Physiques, J.P. Gastellu-Etchegorry - Avril 2008.

➤ Sites web

- ✓ Cours de Xavier Heurtebise
<http://x.heurtebise.free.fr/>
- ✓ Basics of DCT and Entropy Coding, par Nimrod Peleg (Consulter septembre 2022)
www.lokminglui.com/J4DCT-Huff2009.pdf
- ✓ Cours de D. Marshall (U. Cardiff) (Consulter octobre 2022)
<http://www.cs.cf.ac.uk/Dave/Multimedia/PDF/> (cf. chapitres 9,10 et 11)