

## Chapitre 2. Capteurs d'images et dispositifs d'acquisition numérique (2 Semaines)

- Schéma de principe d'une chaîne de traitement d'images.
- Principe des capteurs CCD et CMOS
- Spécifications des capteurs couleur
- Numérisation d'une image
- Notions de définition, résolution et quantification d'une image numérique (taille, dpi, ppi, bpp ...etc)
- Exemples de formats d'images numériques (BMP, TIFF, JPG, GIF et PNG)

Ce chapitre traite du passage du monde physique (la lumière) au monde numérique (les pixels). En Master 1, il est essentiel de comprendre que la qualité d'une analyse d'image dépend d'abord de la qualité de son acquisition.

---

## Chapitre 2 : Capteurs et Acquisition Numérique

### 1. Schéma de principe d'une chaîne de traitement d'images

Une chaîne complète transforme une scène réelle en une décision ou une information.

1. **Acquisition** : Capture de la lumière par un capteur (CCD/CMOS) à travers une optique.
2. **Prétraitement** : Numérisation, débruitage, correction de luminosité.
3. **Segmentation/Extraction** : Isoler les objets d'intérêt (ex: détection de contours).
4. **Interprétation** : Reconnaissance de formes ou mesure (IA, vision industrielle).

### 2. Principe des capteurs CCD et CMOS

Tous deux utilisent l'**effet photoélectrique** pour transformer des photons en électrons, mais leur lecture diffère :

- **CCD (Charge-Coupled Device)** : Les charges sont transférées de pixel en pixel vers un convertisseur unique.
    - *Avantage* : Très haute qualité, faible bruit.
    - *Inconvénient* : Consommation élevée, lecture lente.
  - **CMOS (Complementary Metal Oxide Semiconductor)** : Chaque pixel possède son propre convertisseur d'amplitude.
    - *Avantage* : Très rapide, faible consommation, moins cher.
    - *Inconvénient* : Plus de bruit électronique (corrigé par le traitement moderne).
- 

### 3. Spécifications des capteurs couleur

Un capteur de base est "aveugle" aux couleurs (il ne voit que l'intensité lumineuse). Pour obtenir une image couleur, on utilise principalement :

- **La Matrice de Bayer** : Un filtre de mosaïque colorée (Rouge, Vert, Bleu) placé sur le capteur. Il y a 2 fois plus de filtres Verts car l'œil humain est plus sensible au vert.
- **Le Démosaïçage** : Algorithme qui interpole les couleurs manquantes pour chaque pixel afin de reconstituer l'image RGB finale.

#### 4. Numérisation d'une image

La numérisation transforme un signal continu en données discrètes via deux étapes :

1. **L'Échantillonnage (Spatial)** : On divise l'espace en une grille de petits carrés (pixels). Plus la grille est fine, plus la précision est grande.
2. **La Quantification (Amplitude)** : On associe à l'intensité de chaque pixel une valeur numérique (ex: de 0 à 255).

---

#### 5. Notions de définition, résolution et quantification

Terme	Définition / Unité	Exemple
<b>Définition</b>	Nombre total de pixels (L x H)	Full HD : 1920 x 1080
<b>Résolution</b>	Densité de pixels par unité de longueur	<b>DPI</b> (Dots Per Inch) pour l'impression, <b>PPI</b> (Pixels Per Inch) pour l'écran.
<b>Quantification</b>	Nombre de bits utilisés par pixel ( <b>bpp</b> )	8 bits = 256 niveaux de gris. 24 bits = 16,7 millions de couleurs.

**Calcul de taille** : Un fichier non compressé (RAW) se calcule ainsi :

Taille = Largeur x Hauteur x octets\_par\_pixel

#### 6. Exemples de formats d'images numériques

- **BMP** : Pas de compression. Très lourd, conserve toute l'information.
- **TIFF** : Format professionnel (imprimerie/médical). Supporte la compression sans perte (LZW).

- **JPG (JPEG) : Compression avec perte.** Utilise la transformée en cosinus discrète (DCT). Excellent pour les photos.
  - **PNG : Compression sans perte.** Gère la transparence (canal Alpha). Idéal pour le web.
  - **GIF : Limité à 256 couleurs.** Gère l'animation simple.
- 

## Exercice MATLAB : Analyse des propriétés d'une image

### Matlab

```
% Lecture de l'image
info = imfinfo('peppers.png');

% Affichage des spécifications
fprintf('Largeur : %d px\n', info.Width);
fprintf('Hauteur : %d px\n', info.Height);
fprintf('Profondeur de bit : %d bpp\n', info.BitDepth);
fprintf('Format : %s\n', info.Format);

% Simulation d'une quantification réduite (Passage de 256 à 4 niveaux)
img = imread('peppers.png');
img_gris = rgb2gray(img);
img_quantifiee = uint8(floor(double(img_gris) / 64) * 64);

subplot(1,2,1); imshow(img_gris); title('Original (8 bits)');
subplot(1,2,2); imshow(img_quantifiee); title('Quantifié (2 bits)');
```

**Souhaitez-vous que je vous explique mathématiquement comment le format JPEG réduit la taille d'un fichier en utilisant la chrominance (vue au Chapitre 1) ?**