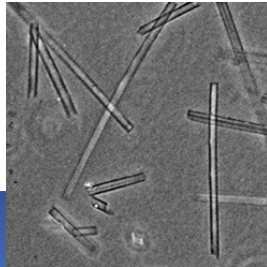
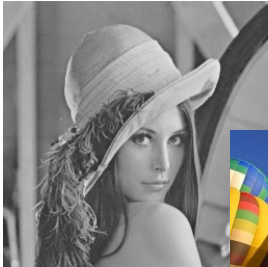


# TS214

## Algorithmes de compression multimédia



# Plan

---

- ▶ Débit
  - Définition
  - Exemples de calcul
  
- ▶ Luminance/Chrominance
  - Codage YUV
  - Reconstitution d'une séquence vidéo
  
- ▶ Entropie
  - Définition
  - Image fixe
  - Séquence vidéo (block-matching)
  
- ▶ Standard JPEG (TP évalué)

# Débit (1/2)

---

## ► Définition

Quantité d'information transmise par unité de temps

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

## ► Unités

- Bits par seconde (bit/s ou bps)
- Bauds (Bd) : nombre de symboles transmis par seconde
  - ✓ Usage souvent abusif : correct pour un signal à 2 valeurs possibles

# Débit (2/2)

- ▶ Lien « confus » entre bits et octets

- Définition : 1 octet (o) = 8 bits (bit)

- Usage « abusif » historique (mémoires, ...)

- ✓ 1 Ko = 1024 o ( $2^{10}$  octets)

- ✓ 1 Mo = 1024 Ko =  $1024^2$  o

- ✓ 1 Go = 1024 Mo =  $1024^2$  Ko =  $1024^3$  o

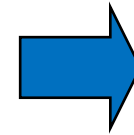
- Standard IEC 60027-2 (1999)

- ✓ 1024 o = 1 Kio (kibiocet), 1 048 576 o = 1 Mio (mébiocet), ...

- Différence de comptage mathématiques/numériques

- ✓ 256 Go = 238,42 Gio

- ✓  $256 \times 10^9$  o =  $238,42 \times 2^{30}$  o



**Octets à  
proscrire  
pour les  
débits**

# Exemples de calcul

---

## ► Temps de transmission (sans bits de contrôle)

- Image binaire  $256 \times 256$ , à 2400 Bd
- Image  $1024 \times 1024$  en 256 niveaux de gris, à 2400 Bd
- Image couleur RGB (8 bits/plan)  $512 \times 512$ , à
  - ✓ 9600 Bd (réseau GSM)
  - ✓ 56 kBd (réseau V.90)
  - ✓ 10 Mbit/s (réseau Ethernet)

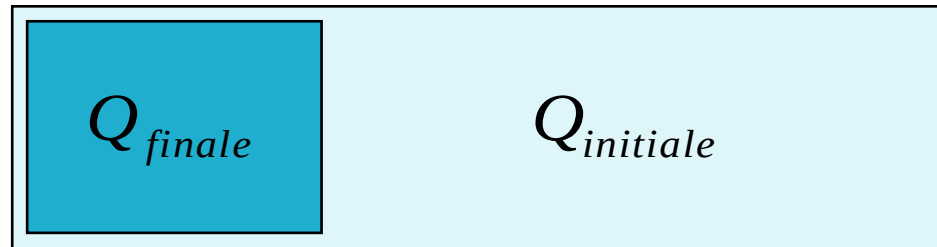
## ► Débit

- Séquence d'images couleur RGB (8 bits/plan)  $720 \times 576$  à 25 images par seconde avec un facteur de compression égal à 100

# Facteur et taux de compression

## ► Facteur

$$F = \frac{Q_{initiale}}{Q_{finale}}$$



## ► Taux (exprimé en %)

$$T = 100 \times \frac{Q_{initiale} - Q_{finale}}{Q_{initiale}} = 100 \left( 1 - \frac{1}{F} \right)$$

# Luminance/Chrominance (1/2)

---

- ▶ Format YUV 4:2:0 utilisé par les caméscopes DV en Europe : images codées sous la forme de blocs de luminance de taille  $W \times H$  suivis par deux sous-blocs de chrominance de taille  $(W/2) \times (H/2)$ 
  - Écrire une fonction *Matlab* qui lit une séquence de  $N$  images de taille  $W \times H$  (utiliser **fopen**, **fread**, **fclose**)
  - Valider la fonction sur la séquence *news30.yuv* (30 images  $352 \times 288$ )
  - Visualiser les différents canaux de luminance et de chrominance (utiliser **im2frame** et **movie**)

# Luminance/Chrominance (2/2)

---

- Reconstruire et visualiser la séquence en couleur

$$\begin{cases} R = 1.164(Y - 16) + 1.596(V - 128) \\ G = 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ B = 1.164(Y - 16) + 2.018(U - 128) \end{cases}$$

Rappel : les données lues par *Matlab* sont stockées « en colonne »



# Entropie (1/3)

---

- ▶ Entropie : Variabilité/Quantité d'information d'un signal

- ▶ Information ~ Innovation

Degré de l'information inversement proportionnel à la probabilité d'apparition

il neige à Tahiti

le soleil se ~~lève~~ à l'est

- ▶ Notations

- Chaîne de symboles (source) à compresser

- ✓ Exemple : AABCABBCBBCDCCCD

- Notations

- ✓ Dictionnaire et alphabet (symboles)  $D = \{C_i\}_{i=1:N}$

- ✓ Source  $S = \{C_j\} \quad C_j \in D$

# Entropie (2/3)

## Quantité (unitaire) d'information

$$Q(C_i) = -k \times \log(p_i)$$

$$k = \frac{1}{\log(K)}$$

nombre d'états différents du codage

$$p_i = \frac{n_i}{M}$$

nombre d'occurrences du symbole

nombre total de symboles de la source

## Codage binaire

$$k = \frac{1}{\log(2)} \longrightarrow Q(C_i) = -\log_2\left(\frac{n_i}{M}\right)$$

# Entropie (3/3)

Entropie H :

$$H(S) = E\{Q(C_i)\} = - \sum_{i=1}^N \frac{n_i}{M} \log_2 \left( \frac{n_i}{M} \right)$$

bits/symbole

→ L'entropie donne la limite basse théorique du minimum de bits sur lequel coder un symbole.

Exemples :



$H_1 = 1$  bit/symbole



$H_2 = 7.56$  bits/symbole

# Entropie (3/3)

---

Entropie H :

$$H(S) = E\{Q(C_i)\} = - \sum_{i=1}^N \frac{n_i}{M} \log_2 \left( \frac{n_i}{M} \right)$$

bits/symbole

→ L'entropie donne la limite basse théorique du minimum de bits sur lequel coder un symbole.

Exemples :

$$S = AABCABBBCBBCDCCCD \quad N = 4 \quad \text{et} \quad M = 16$$

# Entropie (3/3)

---

Entropie H :

$$H(S) = E\{Q(C_i)\} = - \sum_{i=1}^N \frac{n_i}{M} \log_2 \left( \frac{n_i}{M} \right)$$

bits/symbole

→ L'entropie donne la limite basse théorique du minimum de bits sur lequel coder un symbole.

Exemples :

$$S = 0011110011$$

# Calculs d'entropie (1/2)

---

- ▶ Écrire une fonction qui mesure l'entropie d'une image définie par des entiers non signés codés sur 8 bits et l'appliquer aux images :

- *implant.bmp* et *leopard.bmp*

## Rappels

- ✓ La fonction ***hist*** permet de calculer le tableau des occurrences
  - ✓ La syntaxe `A(:)` permet de convertir une matrice en vecteur
- ▶ Comparer les résultats obtenus avec la fonction ***entropy*** de *Matlab*

# Calculs d'entropie (2/2)

---

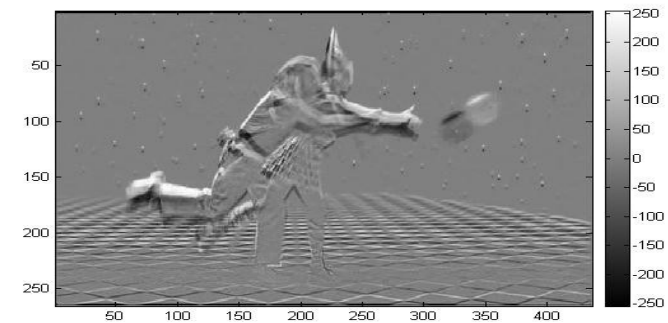
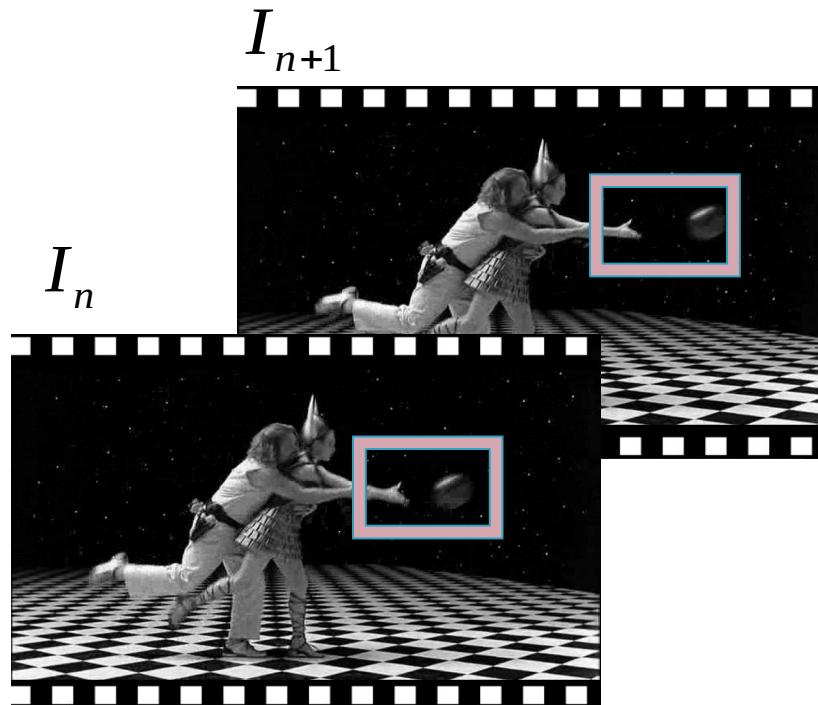
- ▶ Écrire une fonction qui calcule l'entropie de la différence « colonne à colonne » définie par

$$D(i, j) = A(i, j) - A(i, j + 1)$$

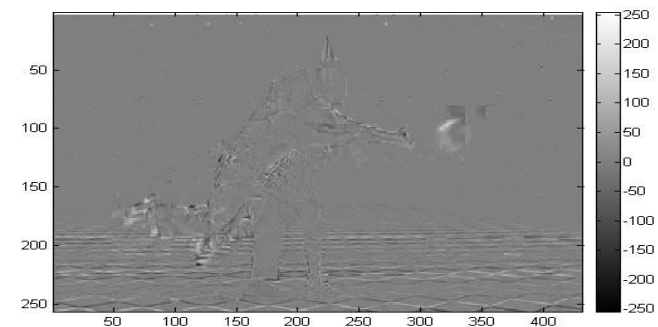
- ▶ Observer l'histogramme de  $D$  et expliquer les résultats obtenus

# Compression Vidéo

## Séquence et mouvement



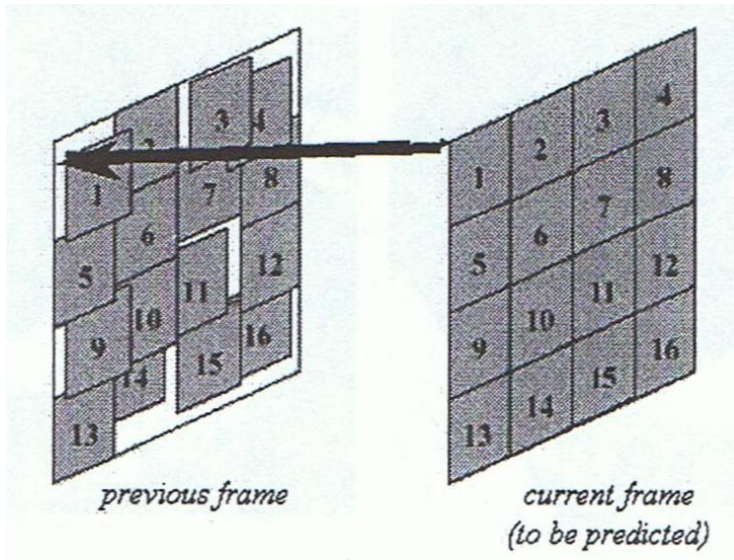
différence simple



différence avec  
compensation de  
mouvement

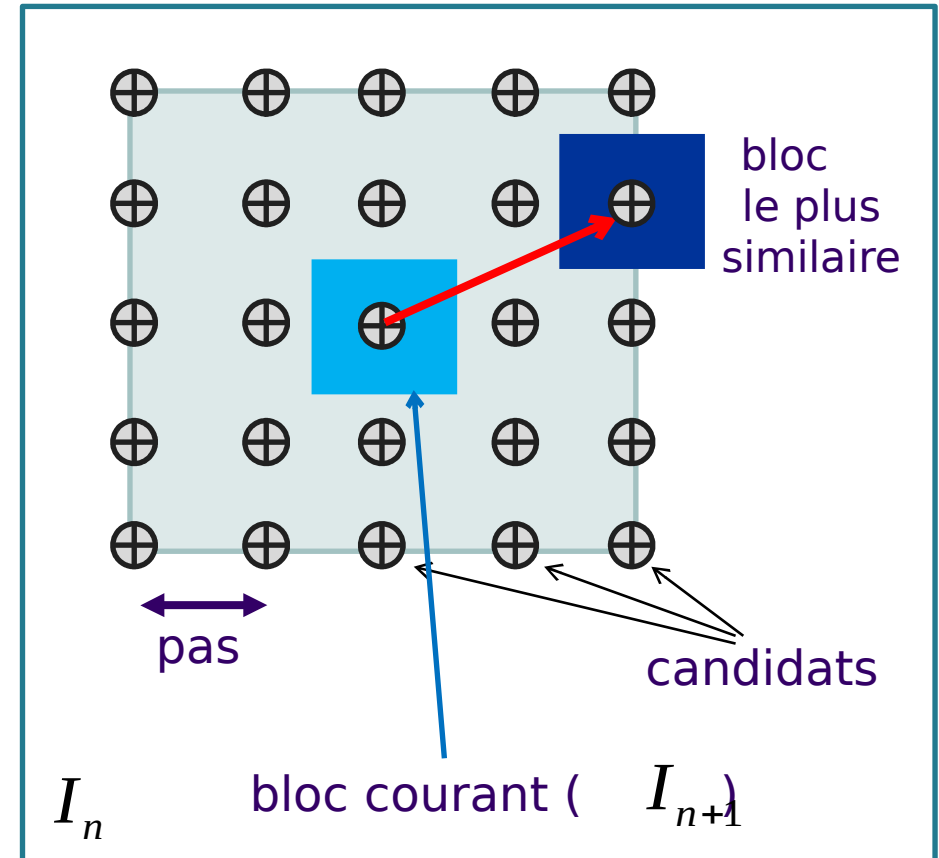


# Block Matching (1/2)



$I_n$

$I_{n+1}$



$I_n$

bloc courant ( $I_{n+1}$ )

# Block Matching (2/2)

---

## ► Paramètres

### ○ Découpe en blocs

✓  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , ...

### ○ Fenêtre de recherche de candidats

### ○ Algorithme de recherche de candidats

✓ Exhaustif

✓ Diamant

✓ ...

### ○ Critère de similarité

✓ SAD

✓ SSE

✓ ...

$$\sum_{\substack{i,j \\ (u,v) \in \Omega}} |I_{n+1}(i, j) - I_n(i+u, j+v)|$$

# Compensation de mouvement (1/3)

---

- ▶ Lire et afficher deux images consécutives  $I_1$  et  $I_2$  de la séquence « *football* »
- ▶ Ecrire une fonction qui estime le mouvement « backward » de «  $I_2$  vers  $I_1$  » par la méthode « Block Matching »
  - Découpage en blocs de  $8 \times 8$  pixels
  - Recherche exhaustive sur des fenêtres de taille  $15 \times 15$
  - Critère SAD
- ▶ Afficher le mouvement
  - Images (composantes horizontale et verticale, amplitude, direction)
  - Vecteurs (flèches) en surimpression (**quiver**) de l'image traitée

# Compensation de mouvement (1/3)

---

- ▶ Lire trois images consécutives  $I_1$ ,  $I_2$  et  $I_3$
- ▶ Calculer et afficher les différences simples  $I_2 - I_1$  et  $I_2 - I_3$
- ▶ Calculer les mouvements « backward » de «  $I_2$  vers  $I_1$  » et « forward » de «  $I_2$  vers  $I_3$  »
- ▶ Calculer et afficher les images
  - $I_1^C$  obtenue après compensation de mouvement « backward »
  - $I_3^C$  obtenue après compensation de mouvement « forward »
- ▶ Calculer et afficher les différences « compensées »  
 $I_2 - I_1^C$ ,  $I_2 - I_3^C$  et  $I_2 - (I_1^C + I_3^C)/2$

# Compensation de mouvement (1/3)

---

- ▶ Réitérer les calculs (mouvements, compensation et différences) avec des mouvements  $\frac{1}{2}$  pixel
- ▶ Comparer les entropies et les énergies (écart-type de l'intensité) de
  - $I_1, I_2, I_3$
  - $I_2 - I_1, I_2 - I_3$
  - $I_2 - I_1^C, I_2 - I_3^C$
  - $I_2 - (I_1^C + I_3^C)/2$
  - $I_2 - (I_1^C + I_3^C)_{\text{subpixel}}/2$
- ▶ Calculer et afficher les entropies et les énergies sur tous les triplets d'images successives de la séquence  
« *football* »