

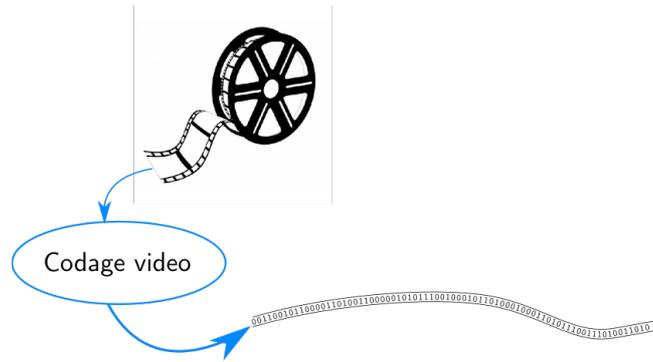
# Compression et transmission video : Concepts et Technologies.

*UE « Systèmes d'Information Tactiques »  
Master 2 Génie Logiciel  
ISTIC, Université de Rennes 1  
Année 2016-2017, second semestre*

Emmanuel Caruyer  
CR CNRS  
Emmanuel.Caruyer@irisa.fr



# Le codage de vidéos



## Plan du cours

### Pourquoi et comment compresser ?

- Calcul mental : transmission d'un flux vidéo brut
- Le codage de Huffman

### Algorithmes spécialisés pour la compression d'images

- Modèles de prédiction (PNG)
- Le modèle colorimétrique YCbCr (JPEG, ...)
- Transformées en cosinus discrète et ondelettes (JPEG, JPEG2000)

### Algorithmes spécialisés pour la compression de vidéos

- Mesures de prédiction de mouvement (MPEG2, MPEG4)
- Quantification vectorielle

## Pourquoi et comment compresser ?

- Calcul mental : transmission d'un flux vidéo brut
- Le codage de Huffman

## Description d'une image numérique

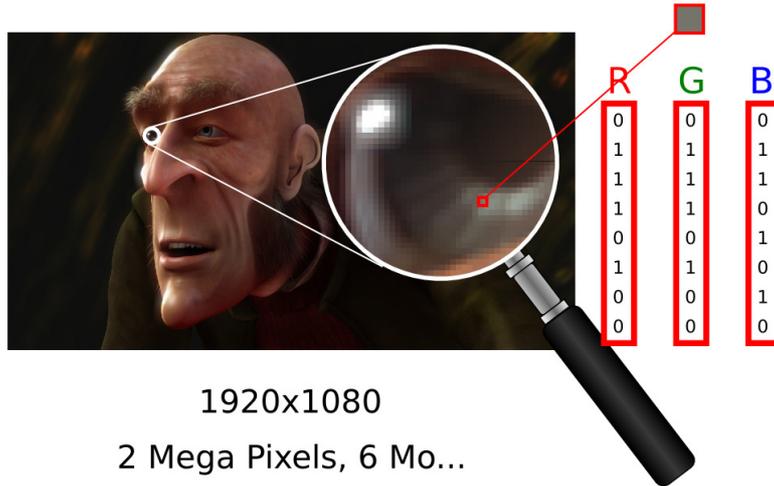


## Description d'une image numérique



1920x1080

## Description d'une image numérique



## Description d'une vidéo numérique



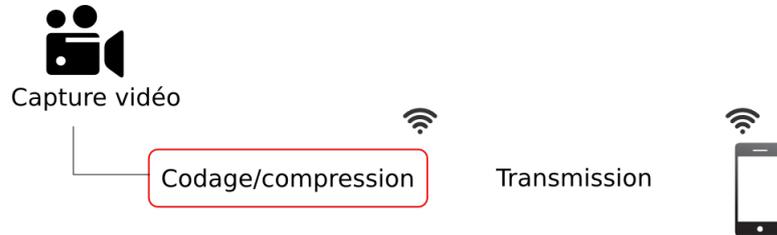
24 images / seconde : pour 7 secondes... **1 Go !!!**

# Stockage et transmission d'images et de vidéos

## Images et vidéos brutes sont très lourdes

- 6 Mo pour une image de 2 Megapixels
- 1 Go pour une vidéo de 7 secondes.
- **Débit nécessaire de 1.2 Gbit/s** pour transférer une vidéo *full HD*.

## Nécessité de compresser la source



## Code à longueur variable

### Représentation standard

- Un texte au format ASCII code chaque caractère sur 8 bits
- Une image brute code chaque pixel par 3 couleurs sur 24 bits
- Une vidéo brute code chaque seconde par 24 images...

### Représentation adaptée à la source : texte en anglais

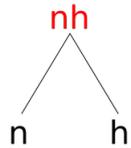
- Certaines lettres sont plus fréquentes que d'autres
- Si on codait de façon plus « économique » les caractères les plus fréquents ?

## Codage de Hauffman (1952)

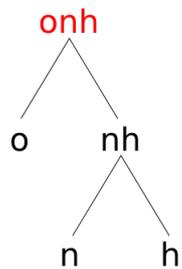
e	45
t	20
a	15
o	10
n	7
h	5

## Codage de Hauffman (1952)

e	45
t	20
a	15
nh	12
o	10

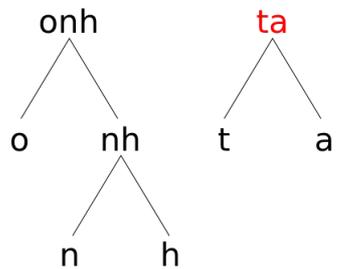


## Codage de Hauffman (1952)



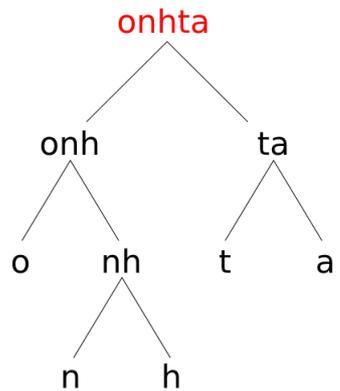
e	45
onh	22
t	20
a	15

## Codage de Hauffman (1952)



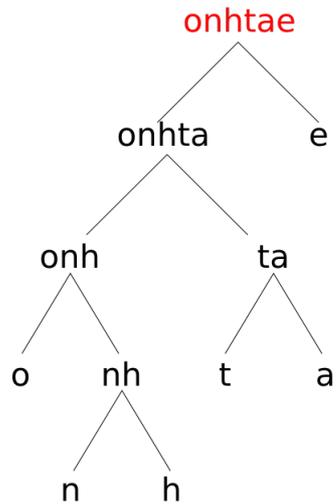
e	45
ta	35
onh	22

## Codage de Hauffman (1952)



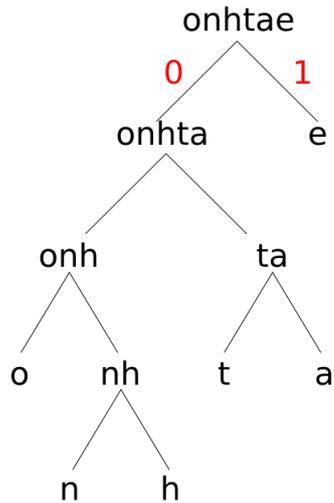
onhta	57
e	45

## Codage de Hauffman (1952)



onhtae 102

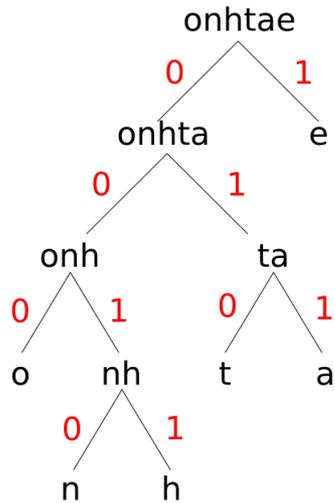
# Codage de Hauffman (1952)



1  
0  
0  
0  
0  
0  
0

e	45
t	20
a	15
o	10
n	7
h	5

## Codage de Hauffman (1952)



1  
010  
011  
000  
0010  
0011

e	45
t	20
a	15
o	10
n	7
h	5

## Codage de Hauffman : et pour une image ?

### Codage adapté pour un texte en langue naturelle

- Peu de symboles différents
- Certains symboles plus fréquents que d'autres

### Et pour une image ?



## Codage de Hauffman : et pour une image ?

### Codage adapté pour un texte en langue naturelle

- Peu de symboles différents
- Certains symboles plus fréquents que d'autres

### Et pour une image ?



## Algorithmes spécialisés pour la compression d'images

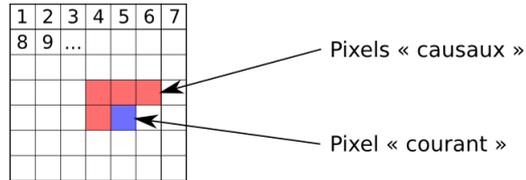
Modèles de prédiction (PNG)

Le modèle colorimétrique YCbCr (JPEG, ...)

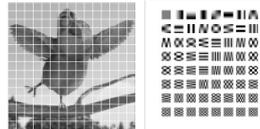
Transformées en cosinus discrète et ondelettes (JPEG, JPEG2000)

## Transformer avant de compresser

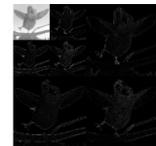
### Prédire la couleur d'un pixel en fonction des voisins



### Utiliser une transformation mathématique de l'image



Transformée en  
cosinus discrète



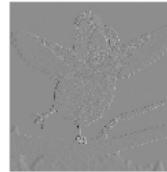
Transformée en  
ondelettes discrète

# Prédiction de la couleur d'un pixel

## Utilisation du pixel voisin de gauche

- Transmission du premier pixel
- Transmission de la différence avec le voisin pour les pixels suivants

1	2	3	4	5	6	7
8	9	...				



- Compression sans perte
- Un taux de compression modeste

## Le modèle colorimétrique YCbCr

Le modèle rouge vert bleu



R



G



B



# Le modèle colorimétrique YCbCr

Le modèle **luminance chrominance**



Y



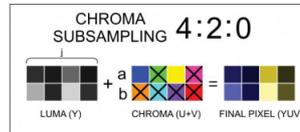
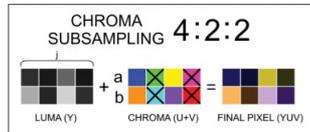
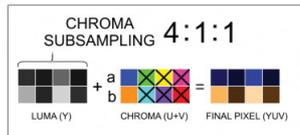
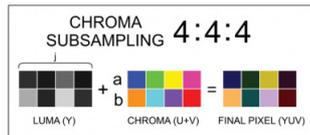
Cb



Cr



# Sous-échantillonnage de la chrominance



## Sous-échantillonnage de la chrominance (4:4:4)



## Sous-échantillonnage de la chrominance (4:2:2)



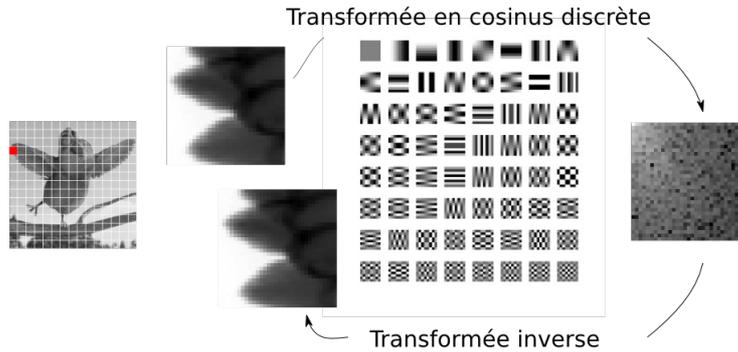
## Sous-échantillonnage de la chrominance (4:2:0)



# Utilisation de la transformée de Fourier 2D

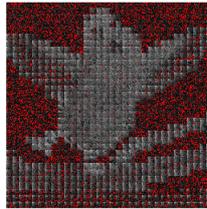
## Transformée en cosinus discrète

- Découpage de l'image en blocs (16x16, 32x32)
- Transformation de Fourier des blocs



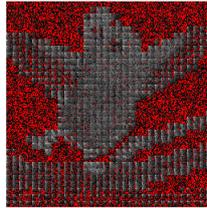
## Compression d'images « avec pertes »

Forçage à 0 de 10% des coefficients



## Compression d'images « avec pertes »

Forçage à 0 de 20% des coefficients



## Compression d'images « avec pertes »

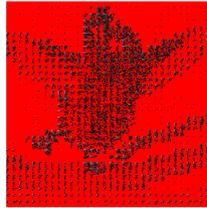
Forçage à 0 de 50% des coefficients



Taux de compression : x2

## Compression d'images « avec pertes »

Forçage à 0 de 80% des coefficients



Taux de compression : x5

## Compression d'images « avec pertes »

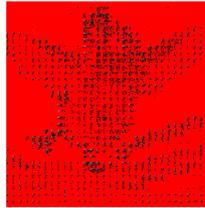
Forçage à 0 de 90% des coefficients



Taux de compression : x10

## Compression d'images « avec pertes »

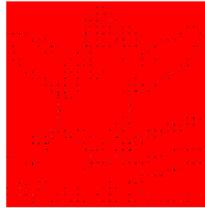
Forçage à 0 de 95% des coefficients



Taux de compression : x20

## Compression d'images « avec pertes »

Forçage à 0 de 99% des coefficients



Taux de compression : x100

# Algorithmes spécialisés pour la compression de vidéos

## Compression de vidéos image par image

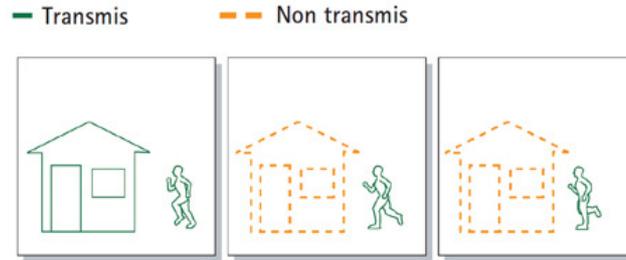
Chaque image est compressée en JPEG



- Peut-on faire mieux ?

# Compression de vidéos : redondance temporelle

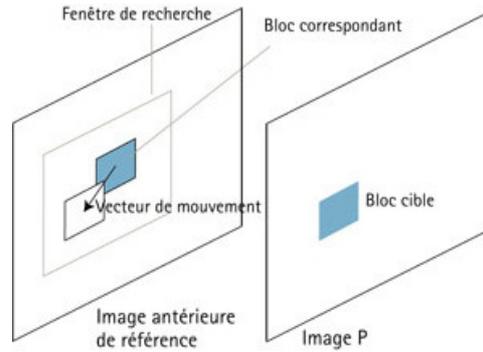
## Un grand nombre d'éléments immobiles dans l'image



- On envoie la différence entre 2 images consécutives
- Peut-on faire mieux ?

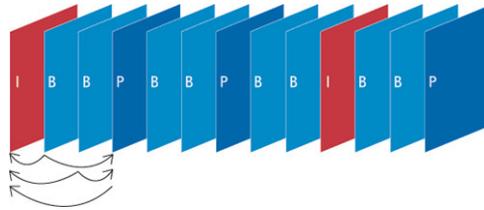
## Compression de vidéos : redondance temporelle

### Compensation du mouvement d'une image à une autre



- On cherche dans une image qu'on connaît déjà les éléments de la nouvelle image à transmettre.

## Compression de vidéos : organisation du flux de données



Cela fait apparaitre 3 types d'images

- I pour Intra
- P pour Predit
- B pour Bi-directionnel

Questions ?