HMIN322 - Codage et compression multimedia

Table des matières

Informations

Examens

- **Note finale** : 60% Exam + 40% TP

Ressources

- Cours et TDs-TPs
- **Mail** :
  - sebastien.beugnon@lirmm.fr
  - wpuech@lirmm.fr

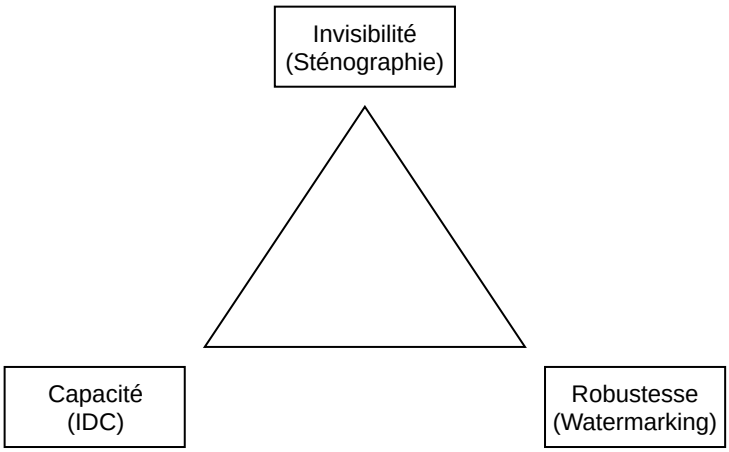
Sécurisation des données

- **Compression** : suppression des redondances en réduisant la taille de l'image
- **Chiffrement** : suppression des redondances sans réduire la taille de l'image

Remarque : on ne peut pas chiffrer une image puis la compresser (plus de redondances). Les algorithmes utilisé sont donc des hybridations d'algorithmes de compression et de chiffrement.

Insertion de données cachées

- Invisible
- Résiste aux transformations
- Contrainte sur la longueur maximum du message
- Résiste aux attaques
- Applicable dans des applications en temps réel



Hachage

- Chiffrement de n'importe quelle donnée en une petite signature de taille fixe
- À la moindre différence entre les deux donnée la signature deviens totalement différente

Caractéristiques d'une image

- Forme
- Texture
- Couleurs
- **Symmetric encryption by block** : DES, TEA, AES laisse des traces de l'ancienne image

Format des images et colorimétrie

Grande diversité d'image ⇒ difficulté à créer un compresseur efficace universel

Codage source

- Compression
- Chiffrement
- Insertion de données cachées

Format d'un fichier image

- **Header**
  - Code : Magic number
  - Format d'image en pixel : L x H
  - Taille d'un pixel : 1 bit (binaire), 8 bits (monochrome), 24 bits (3 x 8 bits : vraies couleurs)
- **Données images**
  - Données des pixels
  - Ordre de lecture des pixels
  - Structures complexes
- **Footer**
  - Informations supplémentaires de l'image

Exemple : Header pour le format PNM (Portable Any Map)

```
P5
512 512
255
# Commentaires
```

Compression sans perte

- **TGA** : Algorithme RLE
- **GIF** : Algorithme LZW
- **PNG** : Algorithme LZ77
- **TIFF** : Algorithme RLE + Codage prédictif
- **JPEG** : Compression sans perte possible : Codage prédictif
- **JPEG2000** : JPEGLS

Mode de transmission

a) Transmission séquentielle

b) Transmission progressive

Entrelacement

Données transmises en un certain nombre de passes.

$2 \to n$ 2 : lignes paires, lignes impaires

7 passes : (Adam 7)

$$\begin{pmatrix} 1 & 1110.1 & a \\ 2 & 10.1 & b \\ 3 & 23.113231 & c \end{pmatrix} \tag{1}$$

$$\begin{pmatrix} 1 & & 4 & & 1 & 4 & & \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ & & & & & & & \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ 3 & "" & 4 & "" & 3 "" & 4 & "" & \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\ "" & "" & "" & "" & "" & "" & "" & "" \\ 7 & 7 & 7 & 7 & 7 & 7 & 7 & 7 \end{pmatrix} \tag{2}$$

1ere passe : 1 pixel (1/64)

2eme : 1 pixel (1/32)

3eme passe : 2 pixels : 1/16

4eme passe : 4 pixels

...

6eme passe : 16 pixels

7eme passe : 32 pixels

*Par plan de bits

(Schéma)

*Pyramide par bloc

image de 512x512 pixels $\to$ 16x16 blocs de 32 pixels

II - Théorie de l'information

1) Quantité d'information

Principes de la compression de données

claude shannon

Théorème fondamental de Shannon : pour une source d'information donnée et un canal d'information il existe toujours un code permettant de transmettre ce message à la capacité du canal avec un taux d'erreur binaire (**TEB**) fixé

Message, signal : séquence d'événements finie ou non

Évènements : valeurs dans un alphabet (texte, signal numérique : bits, octets), images, pixels, son (pressions accoustique, échantillons).

Séquence finie : nombre d'evenements longueur du message

Tous les événements d'un message sont :

- Identiques : source constante
- Indépendants (les uns des autres) : source aléatoire

Si la production d'un évenement est condition des évenements précédents :

- Les évenements corrélés
- La source est Markovienne d'ordre la longueur du message

$\Rightarrow$ Extraction de corrélations

On découpe l'image en bloque

Mesure d'information

information $\rightarrow$ probabilité

émetteur $\rightarrow$ réception

Exemple Météo : L'information à transmettre est un choix éffectué par l'emetteur entre un certain nombre d'évenements plus ou moins probables

Source fini : probabilité $\rightarrow$ ocurence

source : m evenements

symbole $\alpha \rightarrow k$ fois

- $O(\alpha) = k$: occurence
- $O(\alpha) = k/m$: probabilité

Fonction du message à transmettre ~~(et pas l'alphabet)~~ indépendant de la taille de l'alphabet

Exemple : m = "compression_de_donnees"

- Evenements : 22
- Alphabet : 11

$$P(e) = \frac{4}{22} \tag{3}$$

$$\sum p(\alpha_i) = 1 \tag{4}$$

l'information véhiculée par un evenement est inversement proportionnelle à son occurence.

si $proba(\alpha) decrease \Rightarrow information(\alpha) increase$

Exemple : Scrable : W,E

Message : ensemble d'evenements

- l'informaiton véhiculée par un message dont les evenents sont indépendants les uns des autres = somme informations de chaques évenements pris indépendaments.
- l'informaiton véhiculée par un message dont les evenents sont dépendants les uns des autres < somme informations de chaques évenements pris indépendaments.

Ex : scrable Q,U

- cinq, coq, piqûre

$\Rightarrow$ associations d'evenements : réduire la quantité d'information = compression de données

Information : $I(\alpha_i) = f(p(\alpha_i))\ 1 \leq i \leq n$

Evenements indépendants :

$$I(\alpha_i + \alpha_j) = I(\alpha_i) + I(\alpha_j) \tag{5}$$

$$f(`p(\alpha_i) + p(\alpha_j)) = f(p(\alpha_i)) + f(p(\alpha_j)) \tag{6}$$

f : fonciton décroissante de $p(\alpha_i)$

$$I(\alpha_i) = \log_b(\frac{1}{p(\alpha_i)}) = -\log_b(p(\alpha_i)) \tag{7}$$

b = 2 signaux binaires (bits)

Sources binaires

$$P(0) = P(1) = \frac{1}{2} \tag{8}$$

$$I(0) = I(1) = -\log_2(\frac{1}{2}) = 1bit \tag{9}$$

Exemple image 8 niveaux de gris / pixels en 500x600 px

$N = 8^{3.10^5}$ images différentes

$$I = \log_2 8^{3.10^5} \simeq 10^6 bits \tag{10}$$

Texte : vocabulaire : 100 000 mots

message de 1000 mots parmi 100 000)

$$I = 1000.\log_2 10^5 \simeq 2.10^4 bits \tag{11}$$

$$I(\alpha_i) = -\log_2(p(\alpha_i)) \tag{12}$$

- signal aléatoire : alphabet de N symboles

$$p(\alpha_i) = \frac{1}{N} \Rightarrow I(\alpha_i) = \log_2(N) \tag{13}$$

- source markovienne

$$I(\alpha_i) >> \log_2(N) \tag{14}$$

$$I(\alpha_j) << \log_2(N) \tag{15}$$

Exemple S = "9876543210000000000"

$P(9) = P(8) = \ldots = P(1) = \frac{1}{20}$

$I(9) = I(8) = \ldots = I(1) = 4.32bits$

$P(0) = \frac{11}{20} \rightarrow I(0) = 0.86bit$

répétition : $\log_2 10 = 3.32bits$

Nombre de bits par symbole :

- Moins de bits possible pour coder "0"
- Codage à longueur variable : VLC

2) Entropie

Quantité sans dimension $\mathscr{H}(m) = 3.17bits/symbole$

Claude Shannon

Mesurer l'incertitude sur la nature d'un message donné par rappor tà un message qui le précède

- Si aucune incertitude : Entropie nulle
- signal aléatoire : Entropie maximale

**Exemple image 256 couleur (8bits)

- Image noir : 0 bit/pixel
- Image avec du contenu structuré : $5bit/px \leq entropie \leq 7bit/px$
- Image aléatoire 8bit/pixel

Signal S de taille m

$$\alpha_i \quad 0 \leq i \leq N \tag{16}$$

$$\mathscr{H}(S) = \sum_{i=0}^{N-1} p(\alpha_i) I(\alpha_i) \tag{17}$$

$$\mathscr{H}(S) = -\sum_{i=0}^{N-1} p(\alpha_i) \log_2 p(\alpha_i) \tag{18}$$

$$0 \leq \mathscr{H} \leq \log_2 N \tag{19}$$

Extension d'une source

Source discrete sans mémoire (SDSM)

alphabet A: N symboles $\alpha_i \quad 0 \leq i \leq N$

n evenements : blocs de k evenements (k : ordre markovien)

Ex S = "0101111001101111011001111100"

N = 2, A = {0, 1}

Extension d'ordre 3 (k = 3) :

- B = {000, 001, 010, ..., 111}
- N' = 8 $\quad \beta_j$

$$S : p(\alpha_i) \tag{20}$$
$$S' = S^3 : p(\beta_j) = \pi . p(\alpha_i) \quad \mathscr{H}(S^k) = k . \mathscr{H}(S)$$

Si dépendance des evenements: $\mathscr{H}(S^k) < k . \mathscr{H}(S)$

Exemple image de 64x64 pixels.

- Source Markovienne d'ordre $64^2 = 4096$
- Blocs de 4x4 pixels : 16 pixels
- Ordre 16 : $\mathscr{H}(Img^{16}) = 16 . \mathscr{H}(Img)$

Exemple Passage de l'ordre 1 à l'ordre 2

$$\mathscr{H}(S^2) = 2 . \mathscr{H}(S) \tag{21}$$
$$S : \alpha_0 = 0 \quad \alpha_1 = 1$$

Extension d'odre 2

$$S^2 : \beta_0 = 00 \quad \beta_1 = 01 \quad \beta_2 = 10 \quad \beta_3 = 11 \tag{22}$$

$$\mathscr{H}(S^2) = -\sum_{j=0}^{3} p(\beta_j) \log_2 p(\beta_j) \tag{23}$$
$$p(\alpha_0) = p_o \quad p(\alpha_1) = p_1$$

$$= -(p(\beta_0) \log_2 p(\beta_0) + p(\beta_1) \log_2 p(\beta_1) + p(\beta_2) \log_2 p(\beta_2)) \tag{24}$$
$$= -(2p_0^2 \log_2 p_0 + 2p_0 . p_1 \log_2 p_0 . p_1 + 2p_1^2 \log_2 p_1)$$
$$= -2(p_0 + p_1)^2 (p_0 \log_2 p_0 + p_1 \log_2 p_1)$$
$$= 2\mathscr{H}(S)$$

4) Mesure de distorsion

Image originale

(schéma)

$EQM = \frac{1}{N} \sum (p(i,j) - p'(i,j))^2$ N taille de l'image

$SNR : \frac{S}{N} \Rightarrow \frac{S}{B} = 10 . log_{10} \frac{S}{N} dB$

Puissance max d'un pixel : valeure crète : P_{each}

$PSNR = 10 \log_{10} \frac{NDC_{image}^2}{EQM}$

$PSNR = 10 \log_{10} \frac{255^2}{EQM}$

si psnr > 50 db => image très haute qualité (THQ)

30 < psnr <= 50 db => image de bonne qualité (BQ)

20 < psnr <= 30 => moyenne qualité (MQ)

PSNR <= 20 dB => Médiocre

Image couleur : 3 images en NdG

RGB : 3 N pixels en NdB

(Schéma distorsion)

$$Z = \frac{NbBitsImgOriginale}{NbBitsImgCompressée} \tag{25}$$

Z : taux de compression : Z strictement croissant et Z >= 1

distorsion PSNR

WPSNR = PSNR pondéré

$$EQM_w = \frac{1}{N} \sum \frac{(p(i,j) - p'(i,j))^2}{1 + variance(p(i,j))} \tag{26}$$

- Zone homogène : variance ~= 0 et EQM_w ~= EQM
- Zone torturée : variance
- $WPSNR = 10 \log_{10} \frac{255^2}{EQM_w}$

Protection des média visuels

Protection des données en insérant des données cachées

Exemple : Disney protège ses modèles 3D pour éviter la contre façon

11 septembre : plan d'attaques cachés dans des images de chats

cinema : données caché en fonction de la salle qui permet de reperer la place d'ou les images ont été pris

radio : données cachés inserer (texte)

Cryptographie : Transforme les données originales de façon intelligible

Insertion de données cachées : L'art de cacher des données de façon imperceptible au sein d'un média

Phase d'insertion

le média est marqué avec un message secret à l'aide d'une clé secrète

Mode d'insertion

- **Injection** : Message insérer directement dans le média
  - **Problème** : augmentation de la taille du support
- **Substitution** : Le message est inséré de façon à remplacer l'information redondante du support (technique la plus utilisée)
  - **Problème** : Altération du média
- **Distorsion** : Analyse des différences entre objets supports et marqués.

Evaluation robustesse

$$NE = |m| - |m'| + \sum_{i=0}^{|m|} \tag{27}$$

- **Métriques subjectives**

    - MOS (score d'opinion moyenne)
    - Distance
    - Perceptuelle
- **Métriques objectives**
    - PSNR
    - RMSE (Root Mean Square Error)

Sécurité

- Le secret dépend de la clé et non de la méthode

Maillage 3D

- 2-variété (2-manifold) : assure que le maillage est fermé

Compression sans perte

1) Synchronisation - VLC (Variable Length Coding)

Soit une image avec :

- 1 pixel : 3 bits
- 1 pixel 9 bits

le nombre de bit par pixels est variable, cela génère des **problème de synchronisation** comment fait-on pour lire les données il faudra une méthode spéciale pour lire les données de manière cohérentes ?

Objectif : Attribuer les codes les plus court possible aux valeurs les plus probables

$$\mathscr{H}(pixel) \leq longueur \leq \mathscr{H}(pixel) + \alpha \tag{28}$$

VLC : Dictionnaire : mot <----> intensité lumineuse

Solutions :

- bits de signalisation sur chaques mots ()
- Codes à longueur fixes (=> longueur moyenne l est bien plus grande que l'entropie H(pixel))
- Codes à longueur variables préfixés. $l \approx \mathscr{H}(pixel)$

Un code est dit préfixé si il n'est pas le début d'un autre code

Exemple : {a, b, c, d, e, f}

- a : 0
- b : 100
- c : 101
- d : 110
- e : 1110
- f : 1111

Message : 110(d) 0(a) 100(b) 0(a) 110(d) 101(c) 0(a) 0(a) 1111(f) 0(a) (deterministe)

6 symboles : CLF : $\log_2(6) = 3bits/symbole$ (2 symboles avec 4 bits)

2) Algorithme de Shannon-Faro

VLC préfixé, source d'ordre 1 : L = 1

Théoreme du codage de source sans bruit

- L : ordre d'extension d'un bloc B
- li : longueur d'un bloc B, en nombre de bits
- $\bar{l}$: la moyenne par pixel
    - $ \overline{l} = { 1 \over L } \sum p(li)li)

$$\forall \alpha > 0, \exists L, \forall i, \mathscr{H}(I) \leq \bar{l} \leq \mathscr{H}(I) + \alpha \tag{29}$$

$$li = \lceil -log_2(p(i,j)) \rceil \tag{30}$$

Algorithme :

1. Trier les symboles par probabilité décroissante
2. Séparer les symboles en 2 sous-groupes tel que $p(G_1) \approx p(G_2)$
3. - $G_1 \Rightarrow$ Concatener avec un 0 sup
   - $G_2 \Rightarrow$ Concatener avec un 1 sup
4. Pour chaque sous-groupe retour en 2.
5. Arrêt d'un sous-groupe s'il ne contient qu'un element

Exemple : $\{p(\alpha_0) = 0.6, p(\alpha_1) = 0.3, p(\alpha_2) = 0.05, p(\alpha_3) = 0.05\}$

$$\mathscr{H}(S) = \sum_{i=0}^{3} p(\alpha_i) \log(\frac{1}{p(\alpha_i)}) = 1.4 bits/symbole \tag{31}$$

(Shéma Algo)

nul	CLF	truc	li	shanon-faro
α_0	00	1	0	0
α_1	01	2	10	10
α_2	10	5	11000	110
α_3	11	5	11001	111
	2 bits/symbole		1.7 bits/symbole	1.5 bits/symbole

Efficacité d'un codage :

$$eff = \frac{EntropieMessage}{longueurCode} = \frac{1.4}{1.5} \tag{32}$$

3) Algorithme de huffman

Algorithme:

1. Trier les symboles par probabilité décroissante

2. ◦ Regrouper les 2 symboles avec les probabilité les + grandes
   ◦ Remplacer les 2 symboles par le nouveau (avec sa pb)
3. Retour en 2
4. Arrêt quand il ne reste qu'un élément

Image télé :

Résolution : $[720 \times 576 + 2(720 \times 288)] \times 0.21 \approx 200.10^3 bits$

Image originale(entiers) => décorrélation => flottants => entiers => Codeur entropique => image compressé => décodeur entropique => transformée des intensité décorrélation => Image décompressé $I' \approx I$

4) Codage par plage

a) RLE (Run Length Encoding)

Principe : regrouper les pixels voisin ayant la même valeur

RLE : VLC préfixé, ordre 2

Couples : $Q_i = (p_i, l_i) = (Ndg, LongueurDePlage)$

Exemple : Ligne de pixels

50 50 50 50 52 52 52 50 50 50 48 48 50

$$\begin{matrix} Q_0(50,4) \\ Q_1(52,3) \\ Q_2(50,3) \\ Q_3(48,2) \\ Q_4(50,1) \end{matrix} \tag{33}$$

Images de synthèse :

- $0 \leq p_i \leq 255$: 256 couleurs
- $1 \leq l_i \leq M \times N$: taille de l'image

longueur min / max de plages

Codage entropique : les mots qui apparaissent souvent => codage court (Huffman)

b) Blan binaire + RLT

$Q_i = P_i, L_i$

Image ndg $\rightarrow$ décomposition n plans binaires $\begin{cases} MSB \\ \vdots \\ LSB \end{cases}$

$Q_i = l_i \begin{cases} 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{cases}$

$l_i < L_p$ déterministe

RLE + décomposition

$$\left.\begin{array}{l} {}_7 = 64bits \\ L_6 = L_5 = 32bits \\ L_4 = 16bits \\ L_3 = 8bits \\ 4 \geq L_2 \geq 0bits \end{array}\right\} Huffman \leftarrow (\text{LSB, LSB\_0 : 00,01,10,11 non utilité}) \quad (34)$$

$$L_0 = L_1 = 0 \text{ (non utilisé)}$$

Codage binaire : $p(i,j) = 127\ (0111\ 1111) \quad p(i,j+1) = 128\ (1000\ 000)$

Code de gray : valeurs voisines $\Rightarrow$ codage voisins

Exemple : 4 bits

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		■		■		■		■		■		■		■		■
1			■	■			■	■			■	■			■	■
2					■	■	■	■					■	■	■	■
3									■	■	■	■	■	■	■	■

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0		■	■			■	■			■	■			■	■	
1			■	■	■	■					■	■	■	■		
2					■	■	■	■	■	■	■	■				
3									■	■	■	■	■	■	■	■

5) Codage à base de dictionnaire

a) Codage statique

Constructino d'un dictionnaire

$\rightarrow$ Liste de motifs : longueur fixe ou

Exemple : Mise en service de véhicules (fichier texte)

MMAA : (0917)

4 évènements alphabet 10 symboles (0-9)

véhicules de moins de 10 ans

1 symbole : $4bits \rightarrow 2^16 \approx 65536$ mots

10 symboles et 4 événements $\rightarrow 10^4$

$$\begin{array}{l} 2^{13} < 10^4 < 2^{14} \\ 16bis \rightarrow 14bits \\ 01 \leq MM \leq 12 \quad (12 \text{ mois}) \\ 10 \leq AA \leq 19 \quad (10 \text{ dernières années}) \end{array} \quad (35)$$

10^4 mots possibles $\rightarrow 12 \times 10 = 120\, mots \rightarrow 7\, bits$

véhicules plus ancien : bit de signalisation : 0 si mot $\in$ dictionnaire 1 sinon

$$\begin{array}{lll} 0101 \quad (janvier2010) & \Rightarrow & 0000\ 0000 \\ 0210 \quad (fevrier2010) & \Rightarrow & 0000\ 0001 \\ \vdots & & \vdots \\ 1219 \quad (decembre2019) & \Rightarrow & 1111\ 0111 \end{array} \tag{36}$$

b) codage à fenêtre glissante

LZ77 : lesupel et ziv 1977

taille de fenêtre : mémoire

c) Codage dynamique

LZ, LZ77, LZ78 $\rightarrow$ LZW (Welch)

Dictionnaire quasi vide au départ $\rightarrow$ remplir au fur et à mesure

Dictionnaire :

	adresses	valeurs
0	000	00
⋮	⋮	⋮
255	0FF	FF
⋮	100	20, 31
⋮	101	31, 0A
⋮	102	0A, 20
⋮	103	20310A
⋮	⋮	⋮
⋮	FFF	⋮

Algorithme

```
1. Initialisation:
    i <- 0
    c <- lireCaractèreSuivant()
2.  i++; j <- i
    C_ij <- c
3.  c <- lireCaractèreSuivant()
    chaine + c \in Dictionnaire ?
    Oui :
        j++
```

```
        C_ij <- c
        chaine <- chaine + C_ij
        retour en 3.
    Non :
        code <- &chaine
        1er@ libre <- chaine + c
        retour en 2.
```

Exemple : 20 31 0A 20 31 0A 20 41

chaine + c	$\in$ Dictionnaire ?		Nlle @
20, 31	non	@20 = 020	@20, 31 $\leftarrow$ 100
31, 0A	non	@31 = 031	@31, 0A $\leftarrow$ 101
0A, 20	non	@0A = 001	@0A20 $\leftarrow$ 102
20, 31	oui	@2031 = 100	
20, 31, 0A	non		@20310A $\leftarrow$ 103

6) Codage prédictif

Idée : entre 2 pixels voisins il y a peu de différences

$$\begin{aligned} & p(i,j) \ : \ pixel \\ & \hat{p}(i,j) \ : \ \text{pixel voisins } (i \pm h, j \pm l) \\ & \varepsilon = p(i,j) - \hat{p}(i,j) \end{aligned} \tag{37}$$

$$\begin{aligned} \hat{p}(i,j) &= A &&= \tfrac{3(A+C)-2B}{4} \\ &= C &&= \cdots \\ &= \tfrac{A+C}{2} &&= \cdots \\ &= \tfrac{A+B}{2} &&= \cdots \\ &= A + C - B \end{aligned} \tag{38}$$

JPEG sans perte : JPEGLS

Codage prédictif avec préanalyse de l'image DPCM.

chaque pixel p(i,j) :

$$\begin{aligned} & si \quad |A-B| < |B-C| \\ & alors \quad \hat{p}(i,j) = C \\ & sinon \quad \hat{p}(i,j) = A \end{aligned} \tag{39}$$

7) Quantification

a) Scalaire

Protection des données visuelles

- Chiffrement (Cryptanalyse)
- Tatouage / Stéganographie (Steganalyse)
- Biométrie
- Forensiques (Détection de manipulations, identification de capteurs)

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

TP - Chiffrement II

a) Dans le cas d'un chiffrement parfait, quelle est la valeur de l'entropie pow :

- Image 512x512 pixels, 256 niveaux de gris $\rightarrow$ **8 bpp**
- Bloc de 16 pixels, 40 niveaux de gris $\rightarrow$ **4 bpp**
  - $P(\alpha_i) = \frac{1}{16}$

b) Démonstration avec

- k : nombre de pixels
- l : nombre de niveaux de gris

Rappel :

$$\begin{aligned} H(I) &= -\sum_{i=0}^{l} p(\alpha_i) log_2(p(\alpha_i)) \\ &= -\sum_{i=0}^{15} \frac{1}{16} log_2(\frac{1}{16}) \\ &= -\log_2(\frac{1}{2^4}) \\ &= -(\log_2(1) - \log_2(2^4)) = \log_2 2^4 = 4bpp \end{aligned} \tag{40}$$

Projets à choisir

- Compression universelle
- Normalisation
- IDC dans données chiffrée
- Super pixels
- Détection d'images falsifiée CNN
- Partage d'images secrètes

Stage Puech

- Détecter des images modifié
- Suivre poissons

Compression d'images JPEG

Principe du codage arithmétiques

On transforme une chaine binaire en une valeur flottante comprise entre 0 et 1

On fait cela en calculant les probabilité d'avoir un 0 et celle d'avoir un 1

EXAMEN : quantification JPEG

algorithme

3 images reconstruite

image médicale | Girafe | Couleur

PSNR

^

| *

| *

| *

----------> debit bits/pixel

plus les points sont vers la gauche mieux se sera

Examen

savoir différence

- stégano : communication secrete, dissimuler un message dans un medium (image, video, son...)
- tatouage :
- chiffrement :

capacité d'insertion : nb maximale de bit a utilisé (en bit per pixels)

Image originale (I) -> chiffrement (cle Ke) -> Image chiffré (Ie) -> Insertion (message + clé KW) -> Image chiffrée marqué (Iew)

Naive I -> o -> Ie -> lsb substitution -> Iew

homomorphisme : €(IoM) = €(I) o €(M) (RSA, Paillier

15 minutes par soutenances (10 presentation + 5 de question)

Super pixels aller plus loins que la methode de super pixels (variance + algor naifs) supers pixels -> carte binaires -> couleurs rgb -> compresser au maximum compresser en RLE

et on fait des experimentations courbe (debit selon qualité d'image) TRÈS IMPORTANT LES EXPERIMENTATIONS

foutage de visage réversible (chiffrement séléctif) appliqué dct ou transformé en ondelettes puis faire du chiffrement selectif la-dedans appliquer algo de detection de visage deja présent (critère par rapport à l'apparence de la peau pour chiffré)

meilleur compromis psnr + compression image

introduction / context / motivation

methode

resultats

conclusion

super pixels (magnier thibault)