

Examen

Exercice 1 : (04 points)

- 1 - Citez les différentes représentations d'une image ?
- 2 - Que signifie le format vectoriel ? Donnez un exemple de tel format.
- 3 - Est-il possible de compresser une image en format vectoriel ? justifiez
- 4 - Quels sont les critères de comparaison des algorithmes de compression
- 5 - Donnez trois exemples d'algorithmes de compression sans perte.
- 6 - Donnez un exemple d'algorithme de compression avec perte.
- 7 - L'algorithme GIF pour la compression d'images est-il sans ou avec perte ? Justifiez
- 8 - Quel est l'algorithme de compression utilisé dans le format GIF ?

SVG (Scalable Vector Graphic)
EPS (Encapsulated Post Script)
ADOBE

Exercice 2 : (4 points)

Le codage avec l'algorithme LZW a donné le codage suivant :

69 88 65 77 69 78 0 82 83 68. Sachant que le code de code a=65, ..., code(z)=90, code de derniers caractères du dictionnaire :{ | } ~ sont 91, 92, 93, 94 et le code de espace est 0.

Donnez la séquence de caractères qui a été codée.

Exercice 3 : (4 points)

Ecrire un algorithme pour la compression de vidéo par compensation de mouvement. Il est demandé d'écrire et d'utiliser une fonction pour la recherche du correspondant d'un bloc d'une image dans l'image de référence.

Exercice 4 : (4 points)

Une vidéo à résolution de 352x288 et à 30 frames par seconde est compressée en utilisant les frames de type I et P. Le calcul de chaque vecteur de mouvement nécessite en moyenne 3ms. Entre 2 frames de type I on insère 10 frames de type P.

1. Quel est le temps (en secondes) qui sera consommé dans le calcul des vecteurs de mouvement pour la compression de 2 secondes de vidéo ?
2. Peut-on utiliser ceci pour le codage et distribution en temps réel ?
3. Que peut-on faire pour que le codeur calcule plus vite les vecteurs de mouvements ?

Exercice 5 : (4 points)

La compression audio est basée sur deux principes de base. Enoncez ces principes, expliquez et justifiez leur utilisation.

Examen

✓ Exercice 1 : (03 points)

- ✓ Citez les différentes représentations d'une image ?
- ✓ Quels sont les critères de comparaison des algorithmes de compression
- ✓ Donnez trois exemples d'algorithmes de compression sans perte.
- ✓ Donnez un exemple d'algorithme de compression avec perte.
- ✓ L'algorithme GIF pour la compression d'images est-il sans ou avec perte ? Justifiez
- ✓ Quel est l'algorithme de compression utilisé dans le format GIF ?

✓ Exercice 2 : (5 points)

✓ Donnez le résultat du codage avec l'algorithme LZW de la chaîne : aabbaabbaabbabb.
Le code de a=65, ..., code(z)=90.

(X) Exercice 3 : (5 points)

Ecrire un algorithme pour la compression de vidéo par compensation de mouvement. Il est demandé d'écrire et d'utiliser une fonction pour la recherche du correspondant d'un bloc d'une image dans l'image de référence.

Exercice 4 : (5 points)

Une vidéo à résolution de 352x288 et à 25 frames par seconde est compressée en utilisant les frames de type I et P. Le calcul de chaque vecteur de mouvement nécessite en moyenne 3ms. Entre 2 frames de type I on insère 23 frames de type P.

- ✓ 1. Quel est le temps (en secondes) qui sera consommé dans le calcul des vecteurs de mouvement pour la compression de 2 secondes de vidéo ?
- ✓ 2. Peut-on utiliser ceci pour la codage et distribution en temps réel ?
- ✓ 3. Que peut-on faire pour que le codeur calcule plus vite les vecteurs de mouvements ?

(X) Exercice 5 : (2 points)

La compression audio est basée sur deux principes de base. Enoncez ces principes, expliquez et justifiez leur utilisation.

Exercice ②:

Input: aabbbaabbbaabb

Dict

W	K	WK	Edict?	insert wto in dict	out put ced(w)	:
φ	a	a	Oui	/	/	a 65
a	a	aa	Non	aa=91	c(aa)=65	b 66
a	b	ab	Non	ab=92	c(ab)=65	z
b	b	bb	Non	bb=93	c(bb)=66	90
b	a	ba	Non	ba=94	c(ba)=66	ca 91
a	a	aa	Oui	/	/	ab 92
aa	b	aab	Non	aab=95	c(aab)=91	bb 93
b	b	bb	Oui	/	/	ba 94
bb	a	bba	Non	bba=96	c(bba)=93	aab 95
a	a	aa	Oui	/	/	bba 96
aa	b	aab	Oui	/	/	aabb 97
aab	b	aabb	Non	aabb=97	c(aabb)=95	bba 98
b	a	ba	Oui	/	/	aabb 97
ba	a	bba	Non	bba=98	c(bba)=94	abb 99
a	b	ab	Oui	/	/	
ab	b	abb	Non	abb=99	c(abb)=92	
b	/	/	/		c(b)=66	

Exercice ③:

Voir la photo de la

dernière page de ce PDF.

Rq: c'est seulement une partie
de l'algorithme.4. Alg de compression avec Perte:

- JPEG

5. L'Alg GIF est sans perte à
condition d'avoir une image avec
moins de 256 couleurs, au delà,
on perd la qualité de l'image.6. GIF utilise LZW pour la
compression de l'image.

Exercice 4: EMD 2012

Résolution = 352x288.

$$\Rightarrow \text{Nb. Blocs} = \frac{352 \times 288}{16 \times 16} = 396 \text{ Blocs/frame}$$

1) $\rightarrow 30 \text{ frames.}$

2) $\rightarrow k = 30 \times 2 = 60 \text{ frames.}$

I 10P I ...

$\Rightarrow 5 \text{ (I)}$

$\Rightarrow 55 \text{ (P)}$

* Nb Vecteur = $396 \times 55 = 21780$
en mvt

$$\Rightarrow \text{Temps de calcul} = 21780 \times \frac{1}{3} = 65,340 \text{ s}$$

2) On ne peut pas utiliser Ceci pour le codage et distribution en temps réel car temps calcul \gg durée de la Vidéo
($65,340 \text{ s} \gg 2 \text{ s}$)

3) Pour que le codage calcul plus vite les vecteurs de mouvement on peut utiliser le principe de DivX, on divise l'espace de recherche d'un bloc qui a bougé par 2, puis pour chaque partie on vérifie si elle a bougé, on divise encore par 2 et ainsi de suite... sinon on la laisse telle qu'elle est.

Bon : coder uniquement la portion qui a bougé au lieu du bloc entier.

Exercice 4: EMD 2013

Résolution: 352x288

$$\text{Nb. blocs} = \frac{352 \times 288}{16 \times 16} = \frac{101376 \text{ blocs}}{396}$$

1) $\rightarrow 25 \text{ frames}$

2) $\rightarrow 50 \text{ frames}$

I 23P I ...

$\Rightarrow 3 \text{ frames (I)}$

$$\Rightarrow 46 + 1 = 47 \text{ frame (P)}$$

* Nb Vecteur = $47 \times 396 = 18612$
en mvt

* Tpm = $18612 \times 3 = 55,836 \text{ ms}$
 $= 55,836 \text{ s.}$

Exercice 5:

Compression audio :

- compression sans pertes :

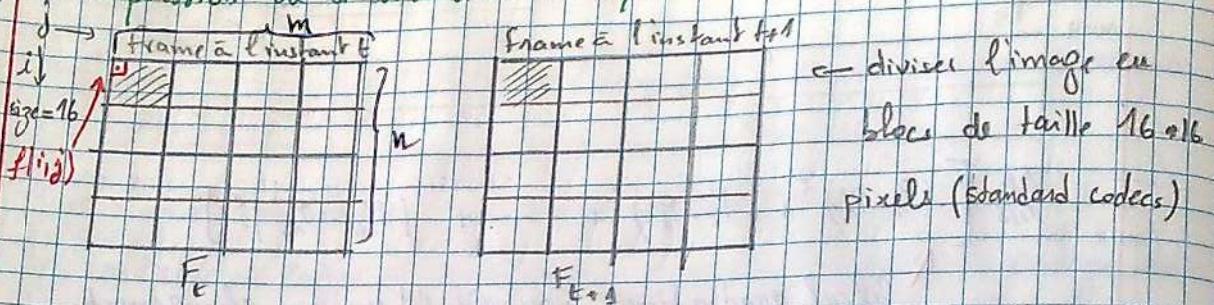
utilise ZIP ou Huffman, recherche des séquences afin de réduire la quantité de données qui doivent être stockées.

- compression avec pertes : Le Modèle psycho-acoustique, l'encodeur supprime l'information qu'on ne peut pas entendre.

- on utilise la compression car :
- on nécessite un stockage facile sur des supports mobiles (Baladeurs) mais limités en Volume.

- Développement de la musique en ligne

1.1.4/ny Compression de la vidéo numérique



- en plus de ça, il y a d'autres qui utilisent une taille adaptative $16 \times 16, 8 \times 8, 4 \times 4$

- après la décomposition de l'image, on applique un algorithme d'algorithmie :

F_t, F_{t+1} ;

// chercher les blocs en mouvement

(comparer le 1^{er} bloc de F_t avec le 1^{er} bloc de F_{t+1} , et déterminer si ils sont identiques ou non).

$i = 0;$

$j = 0;$

// B_{ij} : représente le bloc qui commence au pixel $F(i,j)$.
while ($i \leq n - size$)

Do while ($j \leq m - size$)

Do // traiter le bloc B_{ij} / des frames F_t, F_{t+1})

if ($B_{ij}^t \neq B_{ij}^{t+1}$) then B_{ij}^{t+1} a bougé.

// (\neq signifie : MSE > seuil)

$j = j + size$

End

$i = i + size$ // $size = \text{taille du bloc}$.

End.