

Informatique multimédia

✓ Compression d'image non réversible

- Algorithme JPEG
- Algorithme JPEG 2000

Institut Charles Gros
SISAR

Benoît Piranda
Équipe SISAR
Université de Marne La Vallée

Algorithme JPEG

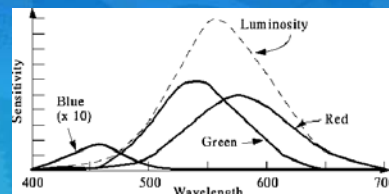
- JPEG (Joint Photographic Expert Group) 1990
- Principe
 - suppression des données peu significatives
 - l'œil est un filtre passe bas
 - suppression des hautes fréquences (peu visibles)
 - Compromis Taille/Qualité
 - Compression jusqu'à 90%

Algorithme JPEG

- Algorithme en 5 étapes
 - Changement de domaine de couleurs
 - Transposition amplitude-fréquence : DCT
 - Quantification
 - Codage RLE
 - Codage de Huffman

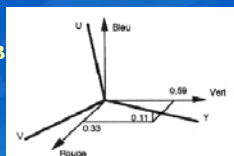
Changement d'espace de couleurs

- Passage du modèle RVB au modèle YUV
 - Raison : la perception de la luminance par l'œil humain
 - La luminance mieux perçue que la chrominance
 - Codage adapté : down-sampling



Changement d'espace de couleurs

- YUV
 - Y : luminance dépend de RVB
 - Base orthogonale
 - U normal au plan RY
 - V normal au plan BY
- Règle de conversion
 - Matrices de conversion

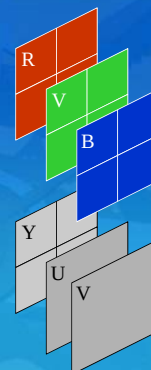


$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0.402 \\ 1 & -0.344136 & -0.714136 \\ 1 & 1.772 & 0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

Changement d'espace de couleurs

- Down sampling
 - Pour 4 pixels voisins
 - 12 octets de codages en RVB
 - On code en YUV
 - la luminance Y sur 4 octets (1 par pixel)
 - la chrominance UV sur 2 octets
 - ✓ U moyen sur les 4 pixels
 - ✓ V moyen sur les 4 pixels
 - Division de la taille de codage par 2
 - Perte de précision
 - Codage YUV sur des entiers
 - Moyenne sur la chrominance



Transformée en cosinus discrète

- Élément centrale de la méthode JPEG
- Transformée en cosinus discrète
 - Méthode comparable à la transformée de Fourier
 - Passage du domaine spatial au domaine fréquentiel
 - Intérêt dans le cas d'une image
 - ✓ Cohérence géométrique
 - ✓ Exploite la cohérence géométrique dans l'image
 - Variations fortes assez rares (hautes fréquences)
 - Peu de coefficients (de fréquences) pour représenter une image
- Méthode inversible IDCT
 - Passage du domaine fréquentiel au domaine spatial

Transformée en cosinus discrète

- Formule de la DCT pour une matrice NxN

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P(x, y) \cos\left(i \times \frac{(2x+1)\pi}{2N}\right) \cos\left(j \times \frac{(2y+1)\pi}{2N}\right)$$

$$\begin{cases} C(0) = \frac{1}{\sqrt{2}} \\ C(x) = 1 & x > 0 \end{cases}$$

- Formule de la IDCT pour une matrice NxN

$$P(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i) C(j) DCT(i, j) \cos\left(i \times \frac{(2x+1)\pi}{2N}\right) \cos\left(j \times \frac{(2y+1)\pi}{2N}\right)$$

Transformée en cosinus discrète

- DCT : projection sur fonctions cos

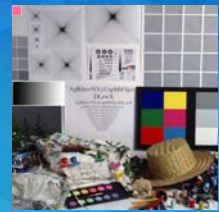
$$\sum_{x=0}^{N-1} \cos\left(i \times \frac{(2x+1)\pi}{2N}\right)$$

Graphique avec N=8, $0 \leq i \leq 8$



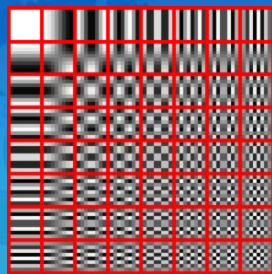
Transformée en cosinus discrète

- Complexité de l'algorithme en $O(N^4)$
 - Pas applicable sur une image complète
 - Découpage en bloc 8x8
 - réalisé en 1987 (ordinateurs à 12Mhz et 1Mo de mémoire)
- Chaque bloc est ensuite traité indépendamment



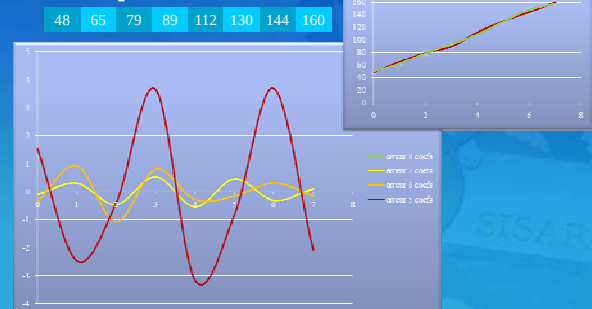
Transformée en cosinus discrète

- Codage de 8x8 fréquences
 - « Images des fréquences » de même taille que l'image initiale
 - Basses fréquences en haut à gauche
 - Hautes fréquences en bas à droite



Transformée en cosinus discrète

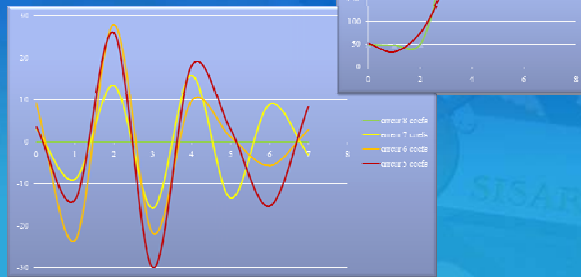
- Exemple avec une ligne d'une image sans hautes fréquences



Transformée en cosinus discrète

- Exemple avec une ligne d'une image avec hautes fréquences

48 49 50 201 200 198 200 197



Simplification de l'algorithme de calcul

- Mémorisation matricielle

- Soit C la matrice mémorisant la transformée en cosinus

$$C_{ij} = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left(\frac{(2j+1)i\pi}{2N}\right) & \text{si } i > 0 \end{cases}$$

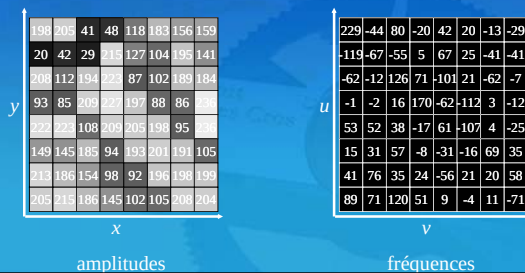
- On en déduit la DCT par le produit matriciel

$$DCT = C \times P(x, y) \times C^t$$

Algorithme JPEG

- Etapes

- Niveau des échantillons centré sur 0 : $-128 < v < 128$
- Transformée en cosinus discrète (DCT)



Quantification

- Filtrage

- élimination des informations non ou peu perçues
- atténuer ou supprimer les fréquences hautes
- utilisation d'une matrice de quantification

- Quantification

- Étape la plus dégradante de l'image

- But :

- Faire apparaître des valeurs nulles dans les hautes fréquences
- Rendre ces valeurs plus probables pour le codage à longueur variable

Quantification

- Matrice de quantification (q : qualité)

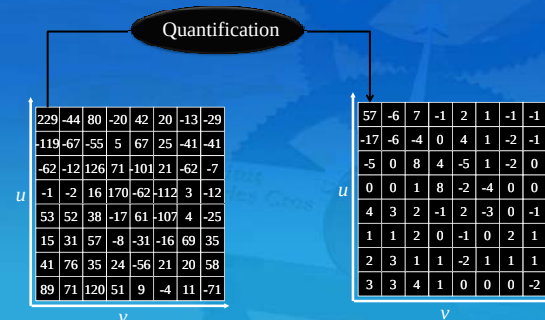
$$Q_{ij}(q) = 1 + q \times (i + j + 1)$$

$$Q(3) = \begin{pmatrix} 4 & 7 & 11 & 13 & 16 & 19 & 22 & 25 \\ 7 & 11 & 13 & 16 & 19 & 22 & 25 & 28 \\ 11 & 13 & 16 & 19 & 22 & 25 & 28 & 31 \\ 13 & 16 & 19 & 22 & 25 & 28 & 31 & 34 \\ 16 & 19 & 22 & 25 & 28 & 31 & 34 & 37 \\ 19 & 22 & 25 & 28 & 31 & 34 & 37 & 40 \\ 22 & 25 & 28 & 31 & 34 & 37 & 40 & 43 \\ 25 & 28 & 31 & 34 & 37 & 40 & 43 & 46 \end{pmatrix}$$

- Ensuite la matrice de DCT est quantifiée :

$$DCT_Q(i, j) = \text{round}\left(\frac{DCT(i, j)}{Q_{ij}}\right)$$

Quantification



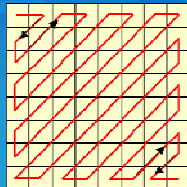
Codage du bloc

- Parcours des éléments de la matrice en zigzag
 - Basses fréquences d'abord
 - Valeurs maximales
 - Hautes fréquences filtrées ensuite
 - Beaucoup de petits entiers

u

57	-6	7	-1	2	1	-1	-1
-17	-6	-4	0	4	1	-2	-1
-5	0	8	4	-5	1	-2	0
0	0	1	8	-2	-4	0	0
4	3	2	-1	2	-3	0	-1
1	1	2	0	-1	0	2	1
2	3	1	1	-2	1	1	1
3	3	4	1	0	0	0	-2

v



Post-traitement

- Codage de Huffman
 - Pour chaque bloc
 - Étude statistique de l'apparition de chaque élément
 - Codage à longueur variable des éléments
 - Fort gain sur les hautes fréquences très dégradées par la quantification

Algorithme de décompression JPEG

- Mêmes opérations à l'envers
 - Décodage Huffman
 - Reconstruction de la matrice de fréquences
 - Application de la transformée en cosinus inverse
 - Placement des blocs dans l'image
 - Conversion en RVB

Exemple de compressions JPEG



Algorithme JPEG 2000

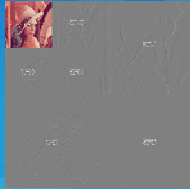
- JPEG 2000 conserve de JPEG
 - Prétraitement : RVB -> YUV
- Répondre aux limites de JPEG
 - Complexité de calcul de la DCT
 - Blocs 8x8
 - Taux de compression limité
- Solutions
 - Autre méthode de compression fréquentielle (DWT)
 - Travailler sur l'image entière
 - Ajoute un mode de compression réversible

Compression JPEG 2000

- On étudie tout d'abord le cas non réversible
- Normalisation de l'image
 - Les valeurs de pixels sont modifiées linéairement
 - Centrée sur 0 et normée
 - De 0-255 à -0.5 à 0.5
 - Avantages
 - Optimisation pour la compression
 - Valeurs indépendantes du codage initial de l'image
- Conversion de l'espace des couleurs
 - Identique à JPEG

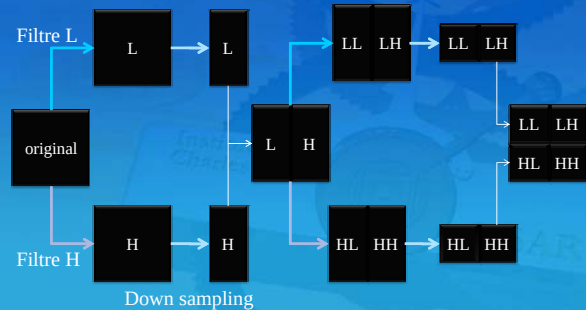
Compression par DWT

- Discret Wavelet Transform (DWT)
 - Transformée par ondelettes discrètes
- Principe de l'algorithme récursif de Mallat
 - Subdiviser l'image en 4 zones (de taille $\frac{1}{4}$)
 - 1 zone conservant les basses fréquences
 - 3 zones mémorisant les hautes fréquences
 - Répéter l'opération récursivement sur la zone de basse fréquence



Algorithme de Mallat

- Construction des sous-images



Compression par DWT

- Outils de calcul des sous-images
 - Deux filtres d'ondelettes
 - Cohen-Daubechies-Fauveau 9/7
 - Filtres de convolution
 - Passe bas pour les basses fréquences (9 coefficients) (0.61, 0.27, -0.08, -0.02, 0.03, 0.27, -0.08, -0.02, 0.03)
 - Passe haut pour les hautes fréquences (7 coefficients) (-0.59, -0.06, 0.10, 1.12, -0.59, -0.06, 0.09)
 - Appliqué horizontalement puis verticalement

Rappels sur la convolution d'une image

- Principe mathématique
 - Convolution de x par h

$$y(t) = \int_{-\infty}^{+\infty} h(v) \times x(t-v) dv$$

Convolution

- Solution numérique
 - Application d'une matrice M_{ij} ($0 \leq i < n, 0 \leq j < m$) à chaque pixel (x,y) de l'image I à transformer
- $$I'(x,y) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} M_{ij} \times I\left(x+i-\frac{n}{2}, y+j-\frac{m}{2}\right)$$
- Le résultat est ensuite borné entre 0 et 255

Convolution

- Exemple de convolution 33

```

ubyte8_t convolution33(ubyte8_t *imageNB, int lx, int ly, float *tab)
{
  unsigned char *ptr1=imageNB, *ptr2=ptr1+lx, *ptr3=ptr2+lx,
    *res = new ubyte8_t[lx*ly],
    *ptrRes=res+lx+1;

  int i=ly-2, j;
  while (i--)
  {
    j = lx-2;
    while (j--)
    {
      *ptrRes++ = ubyte8_t(
        tab[0]*(*ptr1) + tab[1]*(*ptr1+1) + tab[2]*(*ptr1+2) +
        tab[3]*(*ptr2) + tab[4]*(*ptr2+1) + tab[5]*(*ptr2+2) +
        tab[6]*(*ptr3) + tab[7]*(*ptr3+1) + tab[8]*(*ptr3+2));
      ptr1++;
      ptr2++;
      ptr3++;
    }
    ptrRes+=2; ptr1+=2; ptr2+=2; ptr3+=2;
  }
  return res;
}

```

Filtrage de suppression du bruit

- **Filtre passe bas**
 - réduire les points parasites (bruits de mesure).
 - moyenne sur un voisinage
 - supprime donc les détails de la taille inférieure au filtre

Exemple de filtres passe bas

- **Dimensions impaires**
 - Éviter les décalages de l'image traitée

- **Filtre 3x3**

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

- **Filtre 5x5**

$$\frac{1}{25} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

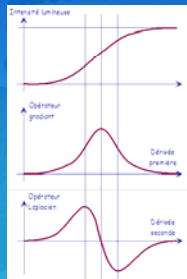
Filtres passe bas

Filtre appliqué 5 fois



Filtrage pour l'extraction de contour

- **Filtre passe haut**
 - augmentent le contraste
 - mettent en évidence les contours.
- **Extraction des hautes fréquences**
 - Variations rapides des valeurs entre pixels voisins
 - Les techniques :
 - de filtres gradients
 - ✓ les contours seront les lignes de maximum locaux
 - de filtres Laplacien
 - ✓ les contours seront les suites de points à 0



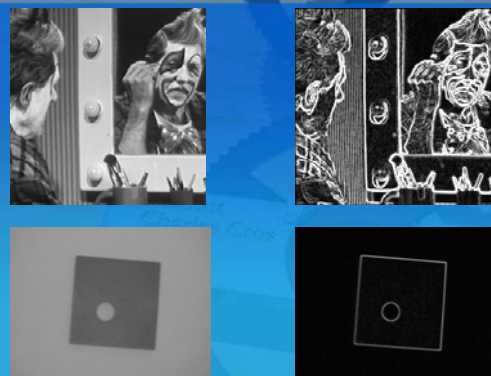
Filtrage par convolution

- **Filtre de Sobel**
- **Principe :**
 - Calculer la norme de l'application de plusieurs filtres
 - Détection des verticales
 - Détection des horizontales
 - Détection des diagonales
 - **Seuillage sur la norme**

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

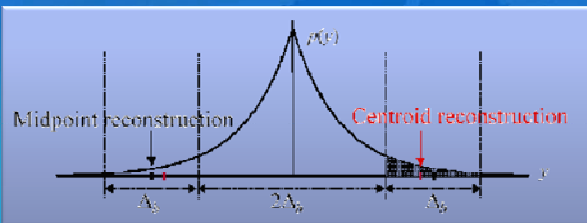
$$\begin{pmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{pmatrix} \quad \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{pmatrix}$$

Seuillage de Sobel



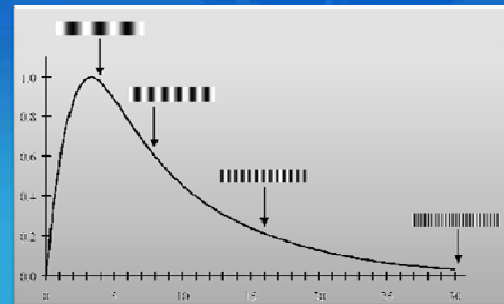
Quantification JPEG 2000

- Comme pour JPEG, mise à 0 des coefficients les moins importants
- Fonction adaptée à la DWT



Quantification JPEG 2000

- Prise en compte de la sensibilité de l'œil humain
– Contrast Sensibility Function



Compression JPEG 2000

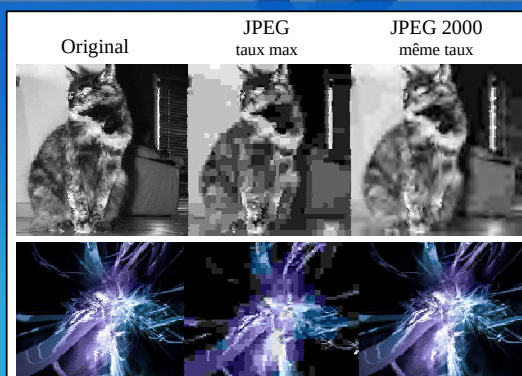
- Étape de codage
 - Utilisation du codage arithmétique adaptatif
- Différences du mode de compression réversible :
 - Modification de la conversion de l'espace des couleurs
 - Pas de normalisation (pour éviter la perte de précision)
 - Ondelettes réversibles « spline 5/3 » de LeGall

Comparaisons JPEG / JPEG 2000

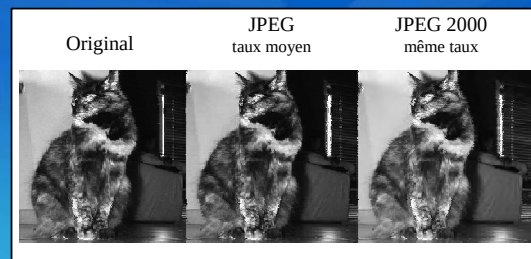
- Les plus de JPEG 2000
 - Décodage progressif et accès aléatoire aux différentes zones de l'image
 - Codage avec et sans perte
 - Flexibilité : adapté à une grande variété de type d'image (NB, gris, photos, images sur 4 octets)
 - Gère la transparence
- Inconvénients
 - Temps de calcul
 - Breveté donc payant

Temps en (ms)	Codage	Décodage
JPEG	100	30
JPEG 2000 irréversible	180	160
JPEG 2000 réversible	180	90

Comparaisons en images



Comparaisons en images



Sources : <http://ljk.imag.fr/membres/Valerie.Perrier/SiteWeb/node22.html>

Compression MPEG

✓ MPEG 1 & 2 Audio

Institut Charles Gros
SISAR

Benoît Piranda
Équipe SISAR
Université de Marne La Vallée

Compression MPEG

- MPEG : Motion Picture Expert Group
 - Codage et compression Audio et Vidéo
 - MPEG-1 : Vidéo niveau VHF
 - MPEG-1 Audio :
 - Layer I : format cassette numérique DCC (Philips)
 - Layer II : format professionnel, qualité CD
 - Layer III : fort taux de compression MP3
 - MPEG-2 : Vidéo qualité DVD
 - MPEG-4 : Fort Taux de compression (Divx)

Compression MP3

- Méthode proche de la compression JPEG
 - Codage en sous-bandes
 - Filtre séparateur de bandes
 - Codage à transformée MDCT
 - ✓ 384 coefficients (échantillons d'entrée)
 - Quantification
 - Huffman

MDCT

- MDCT = DCT modifiée
 - Admet $2N$ valeurs x_n en entrée mais seulement N en sortie X_k

$$X_k = \sum_{n=0}^{2N-1} x_n \cos\left(\frac{\pi(2n+N+1)}{2N}\left(k+\frac{1}{2}\right)\right)$$

- IMDCT

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cos\left(\frac{\pi(2n+N+1)}{2N}\left(k+\frac{1}{2}\right)\right)$$

Compression MPEG Audio

Layer	Complexité		Débit (kbits/s)			Compression maximale (qualité CD)
	encodeur	décodeur	mono	stéréo	qualité CD	
I	1,5 ... 3	1,0	32 ... 448	32 ... 448	256 ... 384	5,5
II	2 ... 4	1,25	32 ... 192	64 ... 384	192 ... 256	7,3
III	> 7,5	2,5	8 ... 64	56 ... 256	112 ... 128	12,6

Compression MPEG

✓ MPEG 1, 2 et 4 Vidéo

Institut Charles Gros
SISAR

Benoît Piranda
Équipe SISAR
Université de Marne La Vallée

Compression d'animation avec perte

- Exploitation de la cohérence spatio-temporelle
- Méthode MPEG 1-2
 - Exploitation de la cohérence spatiale
 - même procédé que le JPEG
 - Exploitation de la cohérence temporelle
 - enregistrer les différences avec une image de référence

La compression MPEG

- MPEG-1
 - 352x282 en 25 images/seconde ou
 - 352x240 à 30 images/seconde
- MPEG-2
 - 720x576 à 25 images/seconde
 - 720x480 à 30 images/seconde

La compression MPEG

- MPEG-1 (norme ISO/CEI 11172)
- Trois parties
 - couche système
 - couche de codage vidéo
 - couche de codage audio

La compression MPEG

- La couche système
 - synchronisation audio / vidéo
 - entrelacement de plusieurs trains binaires comprimés
 - gestion de la mémoire tampon avant restitution de l'image
 - initialisation
 - gestion au cours de la restitution
 - détermination du temps (datation)

La compression MPEG-1

- La couche vidéo
 - désentrelacement vidéo
 - écrite pour traiter les interventions de type magnétoscope
 - lecture, arrêt sur image, accéléré avant / arrière, ralenti...
- Compression : utilisation de plusieurs techniques
 - sous-échantillonnage de la chrominance
 - lié à la sensibilité de l'œil humain
 - quantification : signal = signal moyen + bruit
 - codage prédictif (codage des différences)
 - compensation du mouvement (MC)
 - transformation dans le domaine fréquentiel (DCT)
 - codage VLC
 - interpolation d'images

Codage de la vidéo

- Format standard « 601 » ou 422
 - format YUV
 - 1 octet par pixel
 - formule 422
 - 1 octet / 2 : 4 bits Y, 4 bits U
 - 1 octet / 2 : 4 bits Y, 4 bits V



Compression MPEG

- Quantification

- image

- couleur uniforme
 - ✓ codage facile
 - bruit
 - ✓ œil peu sensible (hautes fréquences)
 - ✓ peut être fortement compressé



Compression MPEG

- Compensation du mouvement

- utilisation de la cohérence temporelle,
 - grande partie de l'image fixe,
 - détermination d'un vecteur de déplacement sur certaine zone,
 - prédiction de la position d'un bloc par translation

Décodeur MPEG-1

