Traitement signal numérique
Parole & Image

4TC-TIP

A. Baskurt, K. Idrissi, C. Garcia
E. Lombardi, R. Kechichian, S. Duffner

Introduction

Compréhension des techniques de traitement du signal pour le codage de la parole et le traitement d'images

- I. Rappels
  - 1. Filtrage numérique
  - 2. Décomposition en sous-bandes
  - 3. Quantification (2h de TD Matlab)
- II. Parole & son (6h)
  - 1. Signal de parole (+TD)
  - 2. Codage de la parole
  - 3. Codage Audio
- III. Image & vidéo (12h)
  - 1. Représentations (+TD)
  - 2. Pré-traitements & Segmentation (+TP)
  - 3. Compression d'images et de vidéos
- IV. Computer Vision & Machine Learning (16h)
  - 1. Indexation des images (+ TD)
  - 2. Reconnaissance de visage (+ TD)
  - 3. Apprentissage profond (Deep Learning) (+TP)

I. Rappels et +

I.1. Filtrage numérique

a) TZ
b) Filtres numériques
c) Modèles ARMA

I.1 Filtrage numérique

a) La transformée en z

la transformée en Z est aux signaux ***échantillonnés*** *ce que la transformée de Laplace est aux signaux continus*

- **Déf.** $$X(z) = \sum_{n \in Z} x(n) z^{-n}$$

X(z) est définie comme une série relative aux échantillons temporels x[n]

$$X(z) = \sum_{n \in Z} x(n) z^{-n} = \sum_{n \in Z} x(n) (\rho e^{jw})^{-n} = \sum_{n \in Z} \left\{ x(n) \rho^{-n} \right\} e^{-jwn} = TF\left\{ x(n) \rho^{-n} \right\}$$

- **Prop.**

cf. cours 3TC-SIS

Linéarité

Décalage temporel

$$Z\{x(n-m)\} = z^{-m} X(z)$$

Convolution temporelle

$$Z\{x(n) * y(n)\} = X(z) Y(z)$$

- **Liens avec la transformée de Fourier discrète**

si on restreint l'espace de z au cercle unité, $z=e^{(jw)}=e^{(j2\pi f)}$
on retrouve la transformée de Fourier :

$$X_d(f) = X(z = e^{j\omega}) = \sum_{n=-\infty}^{+\infty} x[n] \cdot e^{-j2\pi fn}$$

spectre périodique de f=1 ou $f=f_e$

La périodicité du spectre apparaît naturellement, en fonction de la fréquence d'échantillonnage

- **Fonction de transfert numérique H(z)**

Comme pour la TFD, le passage dans le domaine des z, permet de remplacer l'opération de **convolution** par une opération de **multiplication** => mathématiquement très intéressant

I.1 Filtrage numérique

b) Filtres numériques

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1z^{-1}+\cdots+b_qz^{-q}}{1+a_1z^{-1}+\cdots+a_pz^{-p}}=b_0.\frac{\prod_{j=1}^{q}(1-z_jz^{-1})}{\prod_{i=1}^{p}(1-z_iz^{-1})}$$

✓ Les coefficients a_i et b_j sont réels

✓ Fonction de transfert *H(z)* à p **pôles** et q **zéros** réels ou en paires complexes conjuguées

✓ Filtre stable ssi la réponse impulsionnelle h(t) est absolument sommable $\int_{n=-\infty}^{\infty}|h(n)|<\infty$

✓Filtre **stable** si les pôles sont à l'intérieur du cercle unité

- **Implantation : équation aux différences**

$$y(n)+a_1y(n-1)+\ldots+a_py(n-p)=b_0x(n)+b_1x(n-1)+\ldots+b_qx(n-q)$$

$$y(n)=\left(b_0x(n)+b_1x(n-1)+\ldots+b_qx(n-q)\right)-\left(a_1y(n-1)+\ldots+a_py(n-p)\right)$$

expression dans le domaine temporel

$$Y(z)=X(z)H(z)=X(z)\frac{B(z)}{A(z)}=X(z)\frac{b_0+b_1z^{-1}+\cdots+b_qz^{-q}}{1+a_1z^{-1}+\cdots+a_pz^{-p}}$$

expression dans le domaine spectral

$$H(f)=H(z=e^{j\omega})=\frac{b_0+b_1\cdot e^{-j\omega}+\ldots+b_q\cdot e^{-jq\omega}}{1+a_1\cdot e^{-j\omega}+\ldots+a_p\cdot e^{-jp\omega}}$$

- **Filtre FIR : que des zéros**

$$H(z) = \frac{B(z)}{A(z)} = b_0 + b_1 z^{-1} + \cdots + b_q z^{-q} = \sum_{j=0}^{q} b_j z^{-j}$$

- La Réponse Impulsionnelle est de durée Finie (RIF)
- Les coefficients b_j forment la réponse impulsionnelle
- Toujours stable (durée finie + valeurs b_j bornées)
- la sortie est une moyenne pondérée glissante des échantillons d'entrée

$$y(n) = \left(b_0 x(n) + b_1 x(n-1) + \ldots + b_q x(n-q)\right)$$

• Filtre FIR : filtres numériques non récursifs

On a la relation entrée-sortie suivante : $y(n)=\frac{1}{4}\left(x(n)+2x(n-1)+x(n-2)\right)$

▪ Réponse impulsionnelle

$$h(n)=\frac{1}{4}\left(\delta(n)+2\delta(n-1)+\delta(n-2)\right)$$

▪ Réponse fréquentielle

$$H(f)=e^{-j2\pi f}\cos^2(\pi f)$$

• Module $|H(f)|=\cos^2(\pi f)$

• Phase $\arg(H(f))=-2\pi f$

Phase linéaire par rapport à f

• Filtre à moyenne mobile, ou filtre MA (*Moving Average)*

• Phase linéaire possible avec la symétrie de la rép. impuls.

• Filtre IIR : des pôles non nuls

- La Réponse Impulsionnelle est de durée Infinie (RII)
- Un pôle correspond à une réponse impulsionnelle exponentielle
- Stable si pôles à l'intérieur du cercle unité

$$H(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} \left(az^{-1}\right)^n = \frac{1}{1 - az^{-1}}$$

Pôle en z=a

$$converge \text{ si } \left|az^{-1}\right| < 1 \Leftrightarrow |z| > |a|$$

$$\text{et si } |a| < 1, \text{ alors TF converge}$$

$$h(n) = a^n \quad \text{pour} \quad n = [0, \infty[$$

- Si le filtre RII n'a que des pôles : Filtre AR (auto regressive)

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}}$$

$$= b_0 \frac{1}{\prod_{i=1}^{p} (1 - p_i z^{-1})}$$

- Si le filtre RII a des pôles et des zéros : Filtre ARMA (auto regressive moving average)

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}}$$

- **Modélisation MA** (moving average)

  - Modèle 'tout zéros'
  - Spectres doux

$$H(z) = \frac{B(z)}{A(z)} = b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}$$

- **Modélisation AR** (auto regressive)

  - Modèle 'tout pôles'
  - Algorithmes d'estimation très rapides
  - Spectres présentant des pics de résonance
  - Attention : pas toujours stable, phase non linéaire
  - Nécessite moins de place mémoire et de calcul que MA

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}}$$

- **Modélisation ARMA** (auto regressive moving average)

  - Zéros > partie 'moyenne mobile'
  - Pôles > partie 'autorégressive'

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_q z^{-q}}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}}$$

I.1 Filtrage numérique ou modélisation

c) Analyse (filtrage) versus Synthèse (modélisation)

- **Problème de filtrage**
- **Analyse du signal**

définition d'un gabarit

- **Problème de modélisation**
- **Synthèse du signal**

II. Parole

II.1. Signal de parole

a) Production naturelle de la parole
b) Analyse spectrale
c) Synthèse de la parole par une modélisation paramétrique ARMA

II.2. Codage de la parole

a) Quantification
b) Codage direct (PCM)
c) Méthode prédictive (codage différentiel (DPCM)
d) Codage différentiel adaptatif (ADPCM)
e) Codage linéaire prédictif (LPC, CELP)
f) Codage GSM (FR,EFR)

II.3. Codage du son

a) Les sons
b) Modèle psycho-acoustique
c) MPEG Layer 3 audio : MP3

Production de la parole & Analyse temporelle

Représentation simplifiée du conduit vocal

Signature temporelle différente pour chaque son

II.1 Etude du signal de parole

a) Production naturelle du signal de parole

➸ **Un peu de physiologie**

« La parole est le résultat de l'action volontaire et coordonnée des appareils respiratoire et masticatoire, sous le contrôle du système nerveux central. »

Le larynx

- Voisé (partiellement fermé) ou non voisé (ouvert)
- Fréquence fondamentale (pitch)
  - . (150-450 Hz) voix féminine
  - . (80-200 Hz) voix masculine
  - . (200-600 Hz) voix enfantine

Le rôle des cordes vocales : sons voisés

Un son voisé est défini par :

- sa fréquence fondamentale (=hauteur)
- son timbre = rapport entre fondamental et harmonique

C'est un signal déterministe

Sons non voisés

- un son non voisé ressemble à du bruit blanc, donc il faut le traiter comme un signal aléatoire
  - ✓ hypothèse de stationnarité (cf. plus loin)
  - ✓ hypothèse d'ergodicité (cf. plus loin)
  - ✓ étude de la DSP (cf. plus loin)

Représentation simplifiée du conduit vocal

Signature temporelle différente pour chaque son

Les différents types de sons (phonèmes)

Phonème : plus petite unité présente dans la parole et susceptible de changer la signification d'un mot

Phonèmes

- *Voyelles*
  - *Orales*
  - *Nasales*
- *Semi-consonnes*
- *Consonnes*
  - *Liquides*
  - *Nasales*
  - *Frictives*
    - *Voisées*
    - *Non-voisées*
  - *Occlusives*
    - *Voisées*
    - *Non-voisées*

Orales		Nasales		Semi-consonnes		Liquides		Nasales		Frictives: Voisées		Frictives: Non-voisées		Occlusives: Voisées		Occlusives: Non-voisées	
i	(I)	ɛ̃	(IN)	j	(Y)	l	(L)	m	(M)	v	(V)	f	(F)	b	(B)	p	(P)
e	(EI)	œ̃	(UN)	w	(W)	R	(R)	n	(N)	z	(Z)	s	(S)	d	(D)	t	(T)
ɛ	(AI)			y	(UI)			ɲ	(GN)	ʒ	(J)	ʃ	(CH)	g	(G)	k	(K)
a	(A)	ã	(AN)														
ɔ	(O)	ɔ̃	(ON)														
u	(OU)																
y	(U)																
ø	(EU)																
œ	(Œ)																
ə	(E)																
o	(AU)																
a	(Â)																

Les 36 phonèmes du français

* Les voyelles (voisées)

- Orales

[A, E, I, O, U, OU...]

- Nasales

[IN, UN, AN, ON]

* Les consonnes

- Liquides

[R,L]

- Nasales

[M,N,GN]

- Fricatives non voisées

[F (labiale), S (dentale), CH (palatale)]

- Fricatives voisées

[V, Z, J]

- Occlusives non voisées

[P, T, K]

- Occlusives voisées

[B, D, G]

Analyse fréquentielle de la parole

II.1 Etude du signal de parole

b) Analyse spectrale

- **Sons non voisés**

Bruit (souffle)
Signal aléatoire

- **Sons voisés**

signal harmonique
Signal déterministe

Analyse du spectre d'un signal déterministe (sons voisés)

- On peut calculer la DFT (Discrete Fourier Transform)
- Mais on travaillera avec la PSD
  - PSD **(Power Spectral Density)** représente la répartition fréquentielle de la puissance d'un signal suivant les fréquences
  - PSD = DSP (Densité Spectrale de Puissance)
  - PSD = $(TFD)^2$ pour les signaux déterministes

Analyse du spectre d'un signal aléatoire (sons non voisés)

- On ne peut pas analyser le spectre avec une DFT (car chaque réalisation donnera un spectre différent)
- PSD ≠ (TFD)2
- Pour caractériser le spectre des signaux aléatoires, on est obligé de passer par les propriétés statistiques
    - Calcul de la fonction d'auto-corrélation
- PSD = DFT (fonction d'auto-corrélation du signal aléatoire)

Si le signal aléatoire est un bruit blanc

Le son non voisé est un signal aléatoire coloré (n'est pas un bruit blanc parfait)

- la fonction d'auto-corrélation du son « ch » n'est pas un simple dirac, elle est étalée ➔ sa PSD n'est donc pas une constante

Calcul du PSD sous hypothèses

- ✓ Hypothèse de stationnarité du signal sonore :
  - Les propriétés statistiques ne changent pas durant un phonème

- ✓ Hypothèse d'ergodicité du signal sonore :
  - l'évolution d'un signal aléatoire au cours du temps apporte la même information qu'un ensemble de réalisations

$$PSD(X) = mean\left(\left[DFT(X_{\text{différentes réalisations}})\right]^2\right)$$

Durant un phonème, le signal est stationnaire et ergodique

On fait la moyenne des TFD² calculés sur des fenêtres glissantes

$I^{(i)}_N$: i^e TFD² calculé sur N points

S : DSP estimée du signal x

Spectrogramme : un outil puissant pour l'analyse temps fréquence

Non stationnarité :

un spectre par fenêtre temporelle (20 ms)
Short Time Fourier Analysis

« Saurez-vous répondre à ces questions si difficiles ? »

Matlab : spectrogram(phrase_exemple.data,4096,2048,4096,44100,'yaxis')

Formants : résonances caractérisant un phonème

- **Sons voisés**
  - Fondamentale (pitch)
  - Harmonique
  - Formant

Femmes

	F1	F2	F3
i	306	2456	3389
y	305	2046	2535
u	311	804	2485
e	417	2351	3128
ø	469	1605	2581
o	461	855	2756
ɛ	660	2080	2954
œ	647	1690	2753
ɔ	634	1180	2690
a	788	1503	2737

Hommes

	F1	F2	F3
i	308	2064	2976
y	300	1750	2120
u	315	764	2027
e	365	1961	2644
ø	381	1417	2235
o	383	793	2283
ɛ	530	1718	2558
œ	517	1391	2379
ɔ	531	998	2399
a	684	1256	2503

Association voyelles – formant

Synthèse de la parole

$$T(z) = \frac{T_0\left(1 - z^{-1}\right)}{\left(1 - \alpha z^{-1}\right)\left(1 - \beta z^{-1}\right)\prod_{i=1}^{n}\left(1 + a_{1i} z^{-1} + a_{2i} z^{-2}\right)}$$

II.1 Etude du signal de parole

c) Modélisation ARMA du signal de parole

- **La source**
  - Bruit blanc (sons non voisés ou chuchotés)
  - Train d'impulsions périodiques pour les sons voisés.

- **Le conduit Vocal**
  - Tubes acoustiques (pharynx, cavités) de type résonateurs

- **L'émetteur** (lèvres ou narines)
  - Charge acoustique pour l'onde sonore qui y arrive

Source pour les sons voisés

$$\sum_k \delta(n - kT) \text{ avec } f_0 = {}^{1}\!/_{T}, \text{ la fréquence fondamentale}$$

Source pour les sons non voisés

Comment modéliser une résonance

VIII.5 Evaluation géométrique de la Transformée de Fourier

Diagramme des pôles et des zéros de H(z) représenté dans le plan-z permet d 'estimer graphiquement la réponse fréquentielle du système (quand elle existe)

Exemple

$$h[n] = a^n\, u[n] \overset{TZ}{\leftrightarrow} H(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z-a} \qquad |z| > |a|$$

Soit les vecteurs : $v_2 = z - a$

$v_1 = z = e^{j\omega}$

Réponse fréquentielle : $H(e^{j\omega}) = \dfrac{1}{1 - ae^{-j\omega}}$

$|v_2| = |z - a| = \rho \qquad |v_1| = 1$

$$\boxed{|H(j\omega)| = \frac{1}{\rho}} \qquad \boxed{Arg\{H(j\omega)\} = \omega - \varphi}$$

- **Exemple de modèles AR, 1 pôle réel**

réponse impulsionnelle

Module du spectre

Diagramme des pôles

$$|H(z)| = \frac{1}{|z-p|}$$

- **Exemple de modèles AR, 2 pôles conjugués**

Diagramme des pôles

$$|H(z)| = \frac{1}{|z-p||z-p^*|}$$

Une base voisée modélisée par un train d'impulsion filtré

$$G(z) = \frac{G_0}{\left(1+\alpha z^{-1}\right)\left(1+\beta z^{-1}\right)}$$

Conduit vocal > tube acoustique

Un tube acoustique ~= un résonateur

Fonction de transfert d'un résonateur
> Modèle AR ordre 2

$$V_i(z) = \frac{A_i}{1 + a_{1,i} z^{-1} + a_{2,i} z^{-2}}$$

Un résonateur = Un 'formant' à la fréquence :

$$F_i = \frac{1}{2\pi} f_E \cos^{-1}\left[\frac{-a_{1,i}/2}{\sqrt{a_{2,i}}}\right]$$

Le conduit vocal > cascade de n résonateurs > **modèle AR, ordre 2n**

$$V(z) = \frac{A}{\prod_{i=1}^{n} 1 + a_{1,i} z^{-1} + a_{2,i} z^{-2}}$$

Résonateur : exemple

Réponse impulsionnelle

Réponse spectrale

Position des pôles

L'émetteur (lèvres ou narines)

La pression observée à une certaine distance
des lèvres est proportionnelle
à la dérivée du volume des lèvres :

$$\mathbf{R(z) = C \cdot \left(1 - z^{-1}\right)}$$

Modèle MA d'ordre 1

Synthèse d'un son voisé

$$T(z) = \frac{T_0\left(1 - z^{-1}\right)}{\left(1 - \alpha z^{-1}\right)\left(1 - \beta z^{-1}\right)\prod_{i=1}^{n}\left(1 + a_{1i} z^{-1} + a_{2i} z^{-2}\right)}$$

Rque : T(z) : **modèle AR d'ordre 2n+1**, si un des pôles de G(z) est proche de l'unité

Modélisation du spectre de « e » avec un AR d'ordre 9

TF inverse : signal original en rouge et estimé en vert

En résumé

- Complexité de la production de la parole (sons voisés, occlusifs etc...).

- Principales propriétés spectrales du signal de parole (électrique ou acoustique) : fondamentale, harmonique, formants, bande passante vocale, non stationnarité, stationnarité locale dans la durée d'un phonème, signal stochastique

- Modèles ARMA d'un processus physique

- Modèle AR peut suffire dans la majorité des cas

II. Parole

II.1. Signal de parole

a) Production naturelle de la parole
b) Analyse spectrale
c) Synthèse de la parole par une modélisation paramétrique ARMA

II.2. Codage de la parole

a) Quantification
b) Codage direct (PCM)
c) Méthode prédictive (codage différentiel (DPCM)
d) Codage différentiel adaptatif (ADPCM)
e) Codage linéaire prédictif (LPC, CELP)
f) Codage GSM (FR,EFR)

II.3. Codage du son

a) Les sons
b) Modèle psycho-acoustique
c) MPEG Layer 3 audio : MP3

Quantification

Quantification

a) Généralités

- Espace infini de valeurs ➔ espace fini de valeurs
- La dynamique du signal est divisée en intervalles
- Chaque intervalle est défini par des seuils s : **partition P**

Quantification

a) Généralités

- Écart entre 2 niveaux consécutifs : **pas de quantification** Δ
- q : mot code (codeword)
- Ensemble des q : **dictionnaire D** (codebook)

Types de quantification

- Quantification Scalaire (QS) : échantillon par échantillon
  - méthodes simples, surtout QS uniforme
  - ne tient pas compte du contexte (temporel pour un signal, spatial pour une image) : l'échantillon est considéré comme « seul au monde »

- Quantification Vectorielle (QV) : groupe d 'échantillons
  - Vecteur pour un signal, bloc pour une image
  - L'idée est d'utiliser l'existence de motifs qui se répètent dans un signal ou une image : exemple en langue française : « que » ; « et » ; « ion »
  - QV Uniforme ➔ QV sur treillis
  - Optimale > Partitionnement P et codebook D adaptés à la source

Quantification

b) Quantification scalaire

- QS Uniforme Linéaire (QSUL)
  - Partition et dictionnaire uniformément répartis

- QS Uniforme Optimale
  - Partition uniformément répartie et dictionnaire non uniformément réparti

- QS Optimale (algorithme Lloyd Max)
  - Partition et dictionnaire non uniformément répartis

Quantification Scalaire Uniforme Linéaire (QSUL)

- Pas de quantification constant : $\Delta = s_{i+1} - s_i = \dfrac{\max - \min}{Nnq}$
- Mot code (codeword) q = milieu des segments
- Nombre de niveaux : *Nnq* = *dyn*/Δ
- Erreur de quantification : $-\Delta/2 \leq \varepsilon(t) < +\Delta/2$
- Ne tient pas compte de la probabilité de la source
- Plus exactement : hypothèse d'équiprobabilité des messages
  - ➢ densité de probabilité uniforme

$$q_i = \frac{s_{i+1} + s_i}{2}$$

Codeword q au milieu des segments

Partition P : seuils de quantification uniformément répartis

- QSUL :
  - pas besoin de transmettre le codebook et la partition
  - on transmet min, max et le nombre de bits b pour quantifier (une seule fois pour toutes les valeurs d'une source à quantifier)
  - pour chaque valeur à quantifier, on transmet l'indice i du codeword q_i

- QSUL :
  - optimal pour une pdf (probability density function) uniforme
  - ne l'est pas pour d'autres pdf :
    - ✓ **Solution 1 (simple) : modifier le signal pour que son pdf devienne uniforme**
    - ✓ **Solution 2 : adapter la partition P et le codebook D à chaque pdf : algorithme lourd**

Solution 1 : loi de compression pour changer la pdf du signal

Compression (loi A ou μ) → Quantification QSUL

Pré-traitement des valeurs pour tendre vers un signal avec une **densité de probabilité uniforme** et conserver le quantificateur scalaire uniforme linéaire

$$F(x) = \mathrm{sgn}(x) \begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \leq |x| \leq 1 \end{cases},$$

où A est le paramètre de compression. En Europe $A = 87,7$.

Loi de compression logarithmique

Les faibles amplitudes sont « **amplifiées** »

ou « **favorisées** » par rapport aux fortes valeurs

Exemple avec un signal sonore

Compression (loi A ou μ)

Quantification QSUL

histogramme du son :
Proche d'une gaussienne

histogramme du son
après la loi de compression :
proche d'une densité uniforme

Solution 2 : adapter la partition et le codebook au signal : Quantification Scalaire Uniforme Optimale (QSUO)

- Partition P : seuils de quantification uniformément répartis
- Codewords adaptés à pdf (noté p(n) dans la suite)

  ➔ barycentre des plages si p(n) connu

- Diminution de l'erreur de quantification par plage
- non-linéaire > coût : transmission du codebook

$$q_i = \frac{\sum_{n=s_i}^{s_{i+1}} np(n)}{\sum_{n=s_i}^{s_{i+1}} p(n)}$$

Solution 2 (encore mieux) : Quantification scalaire optimale : Algorithme Lloyd-Max 1960

- Minimise l'erreur de quantification globale : $Min\left(\sum_i (x_i - \hat{x}_i)^2\right)$
- Partition et codewords adapté à pdf
- Non-linéaire > coût : transmission du codebook et de la partition

Quantification

c) Quantication vectorielle QV

- Groupe d'échantillons (vecteur) à quantifier
- Idée de QV : exploiter la probabilité conditionnelle entre les échantillons pour mieux faire que si on les quantifiait séparément par un QS : p(« u »)=0,99 si « q » est arrivé avant
- Comme pour la QSUL, on peut
  - partitionner uniformément l'espace de dimension n
  - désigner le milieu de chaque région comme son représentant

représentant : codeword

Exemple de partition uniforme du plan (n=2)

Quantification

d) QV optimisée

- Le partitionnement s'adapte à la répartition des points dans l'espace (un vecteur est un point dans un espace de dimension n)
- Le représentant de chaque région ou classe est le barycentre des points de cette classe : on peut utiliser par exemple une méthode de classification non supervisée : « kmeans algorithm » proposé par MacQuenn en 1967

Codage de la parole

Un monde de normes …

b) Codage direct (PCM)

Norme CCITT G711 (1972) : Modulation par Impulsions Codées (MIC, 64kbits/s) /*(Pulse Code Modulation, PCM)*

- Fenêtre temporelle élémentaire : 20ms
- fe=8kHz (Te=125µs)
- 160 échantillons sur 20ms

- En utilisant ce type de loi, on arrive a une qualité équivalente à du 8k x 12bit (96 kbit/s) en n'utilisant que 8k x 8 bits (64 kbit/s)

c) Codage différentiel (DPCM)

➸ **Principe**

HYPOTHESE : 2 échantillons successifs se ressemblent

On quantifie la différence entre l'échantillon courant et le précédent

Prédiction d'ordre 1 pour réduire la redondance

➸ Relation entre variances du signal et de l'erreur de prédiction

$$\sigma_r^2 = 2\sigma_x^2\left[1-\rho_x(1)\right]$$

Où $\rho_x(1)$ = coeff. d'autocorrelation défini par : $\rho_x(k) = \dfrac{\Phi_{xx}(k)}{\sigma_x^2}$

$$\Phi_{xx}(k) = \int\int_{-\infty}^{+\infty} E\left[X^*(n)X(n+k)\right] = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N} x(n)x(n+k)$$

Stationnaire, ergodique ☺

$$\mu_x = \int_{-\infty}^{+\infty} X(n) f_X(x,n)dx = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{-N}^{N} x(n)$$

$$\sigma_x^2 = \int_{-\infty}^{+\infty} X^2(n) f_X(x,n)dx = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{-N}^{N} x^2(n)$$

$$G_p = \frac{\sigma_x^2}{\sigma_r^2} = \frac{1}{2\left(1-\rho_x(1)\right)}$$ gain de prédiction

➸ Gain en RSB ou en débit

$$G_p = \frac{\sigma_x^2}{\sigma_r^2} \text{ gain de prédiction}$$

• Si $\rho_x(1)>0.5$, $\sigma_r^2 < \sigma_x^2$, prenons le cas où $\rho_x(1)=0.9$ *(cela veut dire une source très corrélée)*

$$G_p = \frac{\sigma_x^2}{\sigma_r^2} = \frac{1}{2(1-\rho_x(1))} = 5$$

• si débit constant ➔ même nombre de niveaux de quantification pour représenter le signal erreur de prédiction avec une dynamique du message à coder qui a diminué ➔ gain en qualité puisqu'on diminue l'erreur de quantification

• si distorsion (ou qualité) constante ➔ comme la dynamique a diminué, on peut réduire d'autant le nombre de niveaux de quantification ➔ gain en nombre de bits pour coder les messages (gain en débit binaire)

Mise en œuvre codage-décodage

$r_q(n) = r(n) + q(n)$

$r(n) = x(n) - x(n-1)$

$r_q(n) = x(n) – (x(n-1) – q(n))$

$y(n) = y(n-1) + r_q(n)$

$y(n) = y(n-1) + x(n) – (x(n-1) - q(n))$

$y(n)-x(n) = y(n-1) – x(n-1) + q(n)$

$e(n) = y(n) - x(n)$

$\boldsymbol{e(n) = e(n-1) + q(n)}$

Intégration de l ’erreur !!!!

➸ *Idée* : coder la différence par rapport à la sortie

autrement dit : prédiction avec des valeurs quantifiées

$r_q(n) = x(n) - y(n-1)$

$r_q(n) = x(n) – (y(n-1) – q(n))$

$y(n) = y(n-1) + x(n) – (y(n-1) - q(n))$

$y(n)-x(n) = y(n-1) – y(n-1) + q(n)$

e(n) = q(n)

Schéma prédictif analysé

Valeur prédite ou estimée qui servira
- à calculer le résidu suivant
- à reconstruire l'échantillon quantifié suivant

échantillon quantifié reconstruit qui va servir à prédire

Schéma complet en généralisant la prédiction à l'ordre p

Prédiction d'ordre p avec $P(z)=a_1z^{-1} + a_2z^{-2} + \ldots + a_pz^{-p}$

Codage de la parole

d) Codage différentiel adaptatif (ADPCM)

Modulation par Impulsions Codées Différentielle Adaptative (MICDA)
Adaptive Differential Pulse Code Modulation (ADPCM) : G726

32 – 16 – 8 – 4 niveaux de quantification ➔ 40, 32, 24 et 16kbps
Forward adaptive prediction : estimer les paramètres de prédiction en utilisant la fenêtre temporelle précédente

➸ Résumons

- Codeur PCM : fe=8kHz, 8 bits/ech 64kbs^{-1}
- Codeur ADPCM : fe=8kHz, 2< x <8 bits/ech 16-40kbs^{-1}

Objectif : moins de 16 kb/sec

Moins de 2bits/éch. !!!!!!!!

Un monde de normes …

Codage de la parole

e) Se servir de la connaissance sur le système phonatoire

Une approche radicalement différente pour réduire le débit : utiliser le schéma de synthèse de la parole pour le codage

Schéma de codage :

On transmet :

- B quantifié et codé
- Paramètres pertinents

Schéma de décodage :

➸ Contraintes pour la prédiction linéaire (estimation)

- Stationnarité : la parole n'a qu'une stationnarité locale
  ➔ fenêtres de 20 ms pour analyser et synthétiser
- Quel ordre p ?
- Quels paramètres AR pour la synthèse ?
- Quel gain ?
- Algorithme de Schur pour déterminer les paramètres du modèle AR

Techniques utilisées :

- LPC (Linear Predictive Coding) : quantification scalaire du résidu ➔ entre 2 et 8 kbps *(qualité compréhensible à 2.4 kbps)*

- CELP (Code Excited Linear Prediction) : quantification vectorielle du résidu ➔ 4.8 kbps (voix naturelle de bonne qualité)

Schéma bloc LPC

On transmet :

- B quantifié et codé
- paramètres du modèle AR
- Gain G
- si voisé, la fondamentale F (pitch)

Mais comment on estime le modèle AR ?

Modèle AR, équation aux différences

$$y(n) = -(a_1 y(n-1) + \cdots + a_p y(n-p))$$

Modèle AR, fonction de transfert en z

$$H(z) = \frac{1}{1 + a_1 z^{-1} + \cdots + a_p z^{-p}}$$

En fait, $y(n)$ est une estimation (prédiction) $y_{estimé}(n)$ avec :

$$y(n) = y_{estimé}(n) + e(n)$$

- on est en présence d'un signal corrélée ➔ ce qui justifie une méthode prédictive AR
- les coeffs a_i doivent être liés à la fonction d'auto-corrélation de y

Montrons la relation entre les coeffs a et les coeffs de corrélation ρ

Estimation du modèle AR d'ordre 2

Etape 1

On multiplie l'équation suivante par $y(n-1)$:

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + e(n)$$

$$y(n)y(n-1) = a_1 y(n-1)y(n-1) + a_2 y(n-1)y(n-2) + y(n-1)e(n)$$

On prend l'espérance :

$$\underbrace{E[y(n)y(n-1)]} = a_1 E[y(n-1)y(n-1)] + a_2 \underbrace{E[y(n-1)y(n-2)]} + \underbrace{E[y(n-1)e(n)]}$$

- Auto-corrélation d'une variable centrée (parole : moyenne quasi nulle)
- Coefficient qui mesure la corrélation entre 2 échantillons successifs : $\boldsymbol{\rho(1)}$
- Erreur d'estimation : bruit blanc ➔ **corrélation nulle entre y et e**

$$\boxed{\rho(1) = a_1 \rho(0) + a_2 \rho(1)}$$

Estimation du modèle AR d'ordre 2

Etape 2

On multiplie l'équation suivante par $y(n-2)$:

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + e(n)$$

$$y(n)y(n-2) = a_1 y(n-1)y(n-2) + a_2 y(n-2)y(n-2) + y(n-2)e(n)$$

On prend l'espérance :

$$E[y(n)y(n-2)] = a_1 E[y(n-1)y(n-2)] + a_2 E[y(n-2)y(n-2)] + E[y(n-2)e(n)]$$

$$\Rightarrow \quad \boxed{\rho(2) = a_1 \rho(1) + a_2 \rho(0)}$$

Estimation du modèle AR d'ordre 2

Etape 3 : résoudre les équations de Yule Walker

$$\rho(1) = a_1\rho(0) + a_2\rho(1)$$
$$\rho(2) = a_1\rho(1) + a_2\rho(0)$$

$$\Rightarrow \begin{pmatrix}\rho(1)\\ \rho(2)\end{pmatrix} = \begin{pmatrix}\rho(0) & \rho(1)\\ \rho(1) & \rho(0)\end{pmatrix}\begin{pmatrix}a_1\\ a_2\end{pmatrix}$$

$R = MA$ avec M : matrice de Toeplitz d'ordre 2, inversible

Résolution par les méthodes de Levinson ou de Schur

$$\Rightarrow \quad A = M^{-1}\, R$$

Les coefficients a_i du modèle de prédiction (modèle AR) s'écrivent donc comme une combinaison linéaire des coefficients de corrélations $\rho(k)$.
En généralisant à un ordre p , $i = 1$ à p :

$$\boldsymbol{a_i = b_1\rho(1) + b_2\rho(2) + \cdots + b_p\,\rho(p)}$$

Equivalent du schéma bloc LPC vu dans le spectre

f) Codeurs GSM : LINEAR PREDICTIVE CODING LPC

- Norme GSM FR (Full Rate) (1988) 13 kbits/sec.
- Norme GSM HR (Half Rate) (1995, Motorola) 5,6 kbits/sec. (qualité comparable à FR)
- Norme GSM EFR (Enhanced FR) (1996, Nokia) 12,2 kbits/sec (meilleure qualité que FR et HR)

Schéma GSM FR

➸ Norme GSM, FR (Full Rate)

(RPE-LTP, *Regular Pulse Excitation with Long Term Prediction*)

1- Pré-filtrage passe-haut (> 80 Hz)

2- **Prédiction à court terme LPC** ordre 8, algorithme de Schur sur N=160échantillons.

Codage des coefficients sur 36bits/fen de 20 ms 1.8kbit/s

3- **Prédiction à long terme** (2 coefs b et Q) sur fenêtres N=40éch.

Codage des coefficients (7+2)bits/fen de 5 ms 1.8kbit/s

4- **Quantification du résidu de 40 échantillons** :

- sous échantillonnage par 3
- on choisit la séquence de 13 éch. la plus énergétique E : 2 bits pour le choix de la séquence S_i
- normalisation de chaque éch. par l'énergie E
- quantification non uniforme sur 3bits des 13 éch.
- quantification non uniforme sur 6bits de E

Codage (2+3*13+6)bits /fen40ech 9.4kbit/s

13 kbit/s en tout

2- Prédiction à court terme : LPC ordre 8, N=160 échantillons

Filtre de blanchiment

$$\mathbf{H(z) = A(z)}$$

On obtient en sortie un résidu (bruit) = *S(z).A(z)*

Paramètres AR définis pour des fenêtres de 20ms (160 éch.)

Algorithme de Schur

$A(z)=1+ a_1.z^{-1} + a_2.z^{-2} + \ldots + a_8.z^{-8}$

Transmission et codage des paramètres
sur 36bits / fenêtre de 160 échantillons (20ms)

Reste une certaine périodicité >>> Prédiction long terme

3. Long Term Prediction (*LTP*)

Ajouter un 2ème étage de prédiction pour prendre en compte la périodicité des sons voisés.... « Pitch synthesis filter »

⇨ **r'(n)=r(n) – br(n-Q)**

$$H_{LT}(z) = \frac{1}{B(z)} = \frac{1}{1 - bz^{-Q}}$$

La périodicité résiduelle de période Q sera détectée par une méthode d'intercorrélation entre les 40 échantillons actuels et les 120 échantillons précédents de la même fenêtre d'analyse de 20 ms

Le résidu est encore « blanchi »

Paramètres b,Q définis pour des fenêtres de 40 éch. ⇨

Pour b : 2bits

Pour Q, 7 bits

➸ Norme GSM, FR (Full Rate)

(RPE-LTP, *Regular Pulse Excitation with Long Term Prediction*)

1- Pré-filtrage passe-haut (> 80 Hz)

2- **Prédiction à court terme LPC** ordre 8, algorithme de Schur sur N=160échantillons.

Codage des coefficients sur 36bits/fen de 20 ms 1.8kbit/s

3- **Prédiction à long terme** (2 coefs b et Q) sur fenêtres N=40éch.

Codage des coefficients (7+2)bits/fen de 5 ms 1.8kbit/s

4- **Quantification du résidu de 40 échantillons** :

- sous échantillonnage par 3
- on choisit la séquence de 13 éch. la plus énergétique E : 2 bits pour le choix de la séquence S_i
- normalisation de chaque éch. par l'énergie E
- quantification non uniforme sur 3bits des 13 éch.
- quantification non uniforme sur 6bits de E

Codage (2+3*13+6)bits /fen40ech 9.4kbit/s

13 kbit/s en tout

HUMAN SPEECH PRODUCTION
QUASI-PERIODIC EXCITATION SIGNAL
VOICED EXCITATION
LUNGS
a
MOUTH, NOSE
SOUND PRESSURE
VOCAL CORDS
ARTI-CULATION
SPEECH
1-a
NOISE-LIKE EXCITATION SIGNAL
UNVOICED EXCITATION
VOICE CODER SPEECH PRODUCTION
VOICED/ UNVOICED DECISION
TONE GENERATOR
SPEECH
ENERGY
FUNDAMENTAL FREQUENCY
VARIABLE FILTER
ENERGY VALUE
NOISE GENERATOR
FILTER COEFFICIENTS
EXCITATION
ARTI-CULATION

➸ En résumé, le codage de la parole

• Codage de base (PCM) 64kbit/s. qualité téléphonique (pas audio).

• Codage adaptatif différentiel (ADPCM), même qualité à 32kbit/s. Acceptable jusqu'à 16kbit/s.

• Codage LPC ou CELP ou GSM : se servir du modèle phonatoire pour coder la parole : très bonne qualité à 10kbit/s.

• Augmentation des performances ➔ complexité des méthodes

II. Parole

II.1. Signal de parole

a) Production naturelle de la parole
b) Analyse spectrale
c) Synthèse de la parole par une modélisation paramétrique ARMA

II.2. Codage de la parole

a) Quantification
b) Codage direct (PCM)
c) Méthode prédictive (codage différentiel (DPCM)
d) Codage différentiel adaptatif (ADPCM)
e) Codage linéaire prédictif (LPC, CELP)
f) Codage GSM (FR,EFR)

II.3. Codage du son

a) Les sons
b) Modèle psycho-acoustique
c) MPEG Layer 3 audio : MP3

II.3 Codage du son

a) Le son

- La parole (cf. partie II.1)
- La musique
- Les autres sons

➸ La parole en résumé

Propriétés connues

- f_{max}=3,4kHz (Fe=8kHz)
- Processus de formation (formants, pitch, harmoniques)
- Modélisation AR performante.
- Loi de distribution ~ Laplace (>> A-law, μ-law)

Objectif : transmission d 'un message compréhensible et reconnaissance du locuteur

Mise en œuvre de codeurs spécifiques très performants

➸ La musique

1° Les notes (pitch)

fondamental, harmoniques, octaves différentes

1 octave = x2

2° Les harmoniques

3° **Le timbre** (forme d ’onde; *waveform*)

4° Musique

- Somme de plusieurs notes (pitch)
- Différentes formes d'ondes (les instruments)
+ chant =parole modulée.

IL FAUT COUVRIR
TOUT LE SPECTRE AUDIBLE

➸ Les autres sons

Applications : TVHD, Vidéo, Multimédia ….

Les bruits ambiants
les onomatopées (miaou, miam miam, oups…)
Bruits sourds (chute d ’eau…)
Bruits brusques (bris de glace…)

TRANSMETTRE TOUT CE QUI EST AUDIBLE

➢ Codage parole : on s'inspire du mécanisme de la production de la parole

➢Codage audio : tout type de son capté par l'oreille ➔ on s'inspire du mécanisme de l'audition

N 'autoriser que les pertes non perceptibles

Codage audio : analyse du signal

- Caractéristiques du signal audio
  - structure harmonique très riche
  - forte variation de la puissance à court terme
  - dynamique spectrale très importante

signal audio

densité spectrale de puissance

Codage audio : oreille comme capteur

- Caractéristiques de l ’oreille adaptées à la nature du signal capté :
  - sensibilité entre 20 Hz et 20kHz : large gamme de sensibilité
  - plus sensible que l’œil en terme de résolution fréquentielle
  - peut supporter une grande dynamique de signaux (130 dB)

Puissance exprimée en dB et seuils de tolérance hebdomadaire

II.3 Codage du son

Modèle psycho-acoustique : seuil d'audition :

masquage fréquentiel 1 : masque en silence

- existe une énergie minimale que doit avoir une composante fréquentielle du signal pour être audible
- **seuil minimal d'audition** : bruit intrinsèque du corps dont le cerveau fait abstraction et donc n'entend pas
- sensibilité maximale entre 1 kHz et 7 kHz

Modèle psycho-acoustique : bandes critiques

- Pour l'oreille, l'échelle des fréquences n'est pas continue, mais découpée en 24 bandes critiques.
- l'oreille est sensible à la puissance acoustique par bande
- 24 sous-bandes de fréquence de largeurs différentes et de sensibilités différentes (bandes critiques : échelle de Bark) :
  - 5 bandes de 100Hz jusqu'à 500Hz
  - ensuite, des bandes plus larges : ~ largeur = fréq. centrale / 5
  - donc la largeur augmente avec la fréquence
- Ainsi, nous différencions plus aisément les sons graves ou moyens que les aigus : plus facile de distinguer deux sons s'ils appartiennent à des bandes différentes.

Modèle psycho-acoustique : bandes critiques

numéro	début	centre	fin
1	20	50	100
2	100	150	200
3	200	250	300
4	300	350	400
5	400	450	510
6	510	570	630
7	630	700	770
8	770	840	920
9	920	1000	1080
10	1080	1170	1270
11	1270	1370	1480
12	1480	1600	1720

numéro	début	centre	fin
13	1720	1850	2000
14	2000	2150	2320
15	2320	2500	2700
16	2700	2900	3150
17	3150	3400	3700
18	3700	4000	4400
19	4400	4800	5300
20	5300	5800	6400
21	6400	7000	7700
22	7700	8500	9500
23	9500	10500	12000
24	12000	13500	15500

Modèle psycho-acoustique : masquage fréquentiel 2

- Un signal puissant dans une bande de fréquence cache d'autres signaux de bandes de fréquence voisines si ces signaux sont moins puissants
- Ce signal crée donc un masque dont le seuil détermine quelles bandes voisines sont masquées (inaudible)

Modèle psycho-acoustique : masquage fréquentiel 2

- Exemple de masquage par une bande de bruit [1100 à 1300]Hz à divers niveaux d'intensité
- l'effet masquant du bruit augmente avec l'intensité
- Si on considère le bruit rouge à 60 dB : un son pur de 1000 Hz ne sera perçu qu'à une intensité de 45dB au lieu de 3dB en l'absence de bruit

Modèle psycho-acoustique : masquage fréquentiel 2

- l'effet masquant augmente avec la fréquence (à l'image des bandes critiques)
- L'excitation à 9kHz est audible, celle à 5kHz ne l'est pas
- Le masque 2 et le masque 1 ne s'ajoutent pas

Vers de nouvelles stratégies de compression

Principe : utiliser les propriétés de masquage de certaines zones de fréquence pour réduire le débit nécessaire.

→ Pertes non perceptibles

→ Passage dans le domaine de Fourier nécessaire

↓

CODAGE SOUS-BANDES

principe proche des bandes critiques

Enjeux du codage audio : les normes

- Musique
  - ΔB=0.02 - 20 kHz, fe=44,1 kHz
  - débit : 16 bits x 44,1 kHz = 705 kbit/s (débit CD), le double en stéréo... ➔ **1,4 Mb/s***60s*60min. = 630 Moctets à stocker
- Pour la diffusion et le stockage, il faut réduire le débit :
  - 705 kbits/s ==> de 192 kbits/s à 64 kbits/s
  - MPEG 1-2 Layer I : 192 kbits/s, algo. MUSICAM, utilisation DCC (Digital Compact Cassette)
  - MPEG 1-2 Layer II : 128 kbits/s, algo. MUSICAM, utilisation DAB (Digital Audio Broadcasting) et DVB (Digital Video Broadcasting)
  - MPEG 1-2 Layer III (MP3) : **96 kbits/s**, algo MUSICAM et ASPEC

Codage audio : quantification et modèle de quantification

- Quantification scalaire uniforme (intervalles identiques)
- Exemple 3 bits par échantillon

signal original

signal quantifié

signal erreur

Codage audio : quantification et modèle de quantification

- Relation entre la variance du signal et celle du bruit de quantification lorsqu ’on quantifie sur b bits :

$$\sigma_S^2 = \sigma_Q^2 2^{+2b}$$

- Rapport signal à bruit :

$$SNR = 10 \log_{10} \frac{\sigma_S^2}{\sigma_Q^2} \approx 6b$$

- Ajouter 1 bit, augmente SNR de 6 dB

Codage audio : quantification scalaire

- exemple avec une dynamique de 70 dB

==> 12 bits par échantillon

exemple avec une dynamique de 100 dB

==> 16 bits par échantillon (CD) : n'oublions pas qu'au départ, le signal est analogique : on l'a numérisé sur 16 bits, cette quantification a introduit une erreur ➔ 16 bits nous garantit un PSNR de 96dB…

Codage audio : quantification avec le modèle psychoacoustique

- estimation par bande des dynamiques signal et seuil de masquage ➔ rapport signal masque (SMR : Signal to Mask Ratio)
- 4, 2, 2 et 0 bits en fonction des bandes
- l'erreur de quantification est masquée

puissance du signal dans cette bande : 64 dB

puissance estimée du seuil de masquage dans cette bande : 40 dB

➸ Principe général de MPEG-1 *(et suivants)*

Codage audio MPEG 1 - 2

- 32 bandes
- pour chaque bande de fréquence, cacher (masquer) le bruit de quantification en dessous du seuil de masquage

Schéma bloc codeur MPEG-1 Layer.III
Fe = 48000 Hz
32 sous bandes de 750 Hz chacune

Résumons …

Codage audio
(musique)
Méthodes basées système d'audition

Codage parole
(téléphonie)
Méthodes basées systèmes phonatoires

Codage :

- **Compromis débit distorsion**
- **propriétés du signal source + propriétés du récepteur**

III. Image

III.1. Représentations

a) Continue, échantillonnée
b) Voisinage, connexité, distance
c) Acquisition : échantillonnage, quantification, bruit
d) Représentations fréquentielles
e) Représentations pyramidales
f) Représentation de la couleur

III.2. Pré-traitements & Améliorations

a) Opérations pixel à pixel
b) Opérations sur le voisinage (Filtrage)
c) Transformations géométriques

III.3. Compression d'images

a) Approches directes
b) Approches par transformation
c) Compression de séquences

III.4. Ouvertures

a) Segmentation
b) Indexation
c) Tatouage

Recherche dans BD
Tatouage
Image numérique
Indexation
Compression
Traitements simples pour l'analyse
Segmentation
Transmission
Reconnaissance de formes
Décision

III.1 Représentation continue

a) Représentation continue

- Image = fonction d'au moins deux variables réelles
  - Image : *f(x,y)* *image 2D*
  - Volume : *f(x,y,z)* *«image» 3D*
  - Séquence d'image : *f(x,y,t)*
  - Séquence de volumes : *f(x,y,z,t)* *«image» 4D*
- Les valeurs prises par *f(.)* peuvent être
  - Scalaires (intensité lumineuse)
  - Vectorielles (couleur (RVB, ..), imagerie multispectrale, image de paramètres...)
  - Réelles ou complexes

- Une image 2D *f(x,y) scalaire réelle* peut être vue comme une surface en 3D :

Interprétation altimétrique des images, bassin versant, détection de ligne de crêtes, dénivellation ...

- Si *f(.)* représente une intensité lumineuse

Cette représentation est utilisée
quel que soit le paramètre représenté par *f(.)*
(Température, pression,....)
Correspondance entre niveau de gris et grandeur physique.

- Opérations sur les images continues :

  Toutes opérations réalisables «sur le papier» sur les fonctions continues à variables réelles

  Transformée de Fourier ***bidimensionnelle (2D)***

  Filtrage, convolution, corrélation, intégration, dérivation, traitements non linéaire...

- On utilisera souvent la notation «continue» pour représenter et manipuler des images numériques *(discrètes, échantillonnées, quantifiées)*

- Le traitement numérique de l'image sera parfois une «discrétisation» d'une opération en continu

b) Représentation échantillonnée

- Echantillonnage d'une fonction *f(x,y)*

$$\tilde{f}(i\Delta x, j\Delta y) = f(x,y).\Sigma_i \Sigma_j \delta(x - i\Delta x, y - j\Delta y)$$

Δx pas d'échantillonnage dans la direction *x*

Δy pas d'échantillonnage dans la direction *y*

$\Sigma_i \Sigma_j \delta(x - i\Delta x, y - j\Delta y)$ Peigne de Dirac 2D

- Le poids $\tilde{f}(i\Delta x, j\Delta y)$ de chaque Dirac est :
  - Soit la valeur de *f(x,y)* en *x* = *i* Δx et *y* = *j* Δy
  - Soit la valeur «moyenne» de *f(x,y)* dans une région entourant *(i* Δx *, j* Δy*) (f(x,y) est pondérée et intégrée dans la région R)*

Caméra CCD

- Dans un ordinateur, l'image (numérique) sera représentée par une matrice (tableau 2D) : $f[i,j] = \tilde{f}(i\Delta x, j\Delta y)$

- $f[i,j]$ est appelé «valeur du PIXEL (i,j) »
  (Pixel: PICture ELement)

- Pour visualiser une image, on remplit une région rectangulaire (Pixel) avec un niveau de gris (ou de couleur) correspondant à la valeur du pixel. En général les niveaux de gris (ou de couleur) utilisé pour la visualisation sont compris entre 0 et 255 (code de longueur fixe sur 8 bits).

c) Voisinage, connexité, distance

- Beaucoup de traitements font intervenir la notion de **voisinage**
- Un pixel possède plusieurs voisins (4 ou 8)

- On parlera de **connexité** 4 ou 8

La région grise forme :
UN seul objet en connexité 8
DEUX objets en connexité 4

➸ **Distance spatiale** entre deux pixels f [i,j] et f ’[k,l]

- <u>Distance Euclidienne</u> $$d_e(f,f') = \sqrt{(i-k)^2\Delta x^2 + (j-l)^2\Delta y^2}$$

- Distance *City-Block* $$d_c(f,f') = |i-k|\Delta x + |j-l|\Delta y$$

  longueur du chemin en connexité 4

- Distance de l’échiquier $$d_b(f,f') = \max(|i-k|\Delta x, |j-l|\Delta y)$$

➸ **Distance radiométrique** entre deux images f et f ’

- Distance Euclidienne (l_2)

$$d_e(f,f') = \left[\sum_i \sum_j \left(f(i,j) - f'(i,j) \right)^2 \right]^{1/2}$$

- Distance (l_1)

$$d_c(f,f') = \sum_i \sum_j \left| f(i,j) - f'(i,j) \right|$$

- Distance max (l_∞)

$$d_\infty(f,f') = \max_{i,j} \left(\left| f(i,j) - f'(i,j) \right| \right)$$

d) Acquisition : échantillonnage / quantification

➸ Effets de l'**échantillonnage** : **pixelisation**

256 x 256 pixels

64 x 64 pixels

16 x 16 pixels

- Contours en marche d'escalier
- Perte de netteté
- Détails moins visibles/ moins précis
- Perte de résolution

Effets de la **quantification** à l'acquisition

- CAN sur les systèmes d'acquisition d'images
- Codage de la valeur de chaque pixel sur N bits (En général 8 bits)

8 bits (256 niv.)

4 bits (16 niv.)

2 bits (4 niv.)

- Apparition de faux contours
- Bruit de quantification
- Effet visible à l'œil en dessous de 5 bits
- Quantification sur 8 bits pour l'affichage

e) Représentations fréquentielles

- Notion de fréquence spatiale
- Transformée de Fourier
- Transformée Cosinus

➸ Notion de **fréquence** spatiale

- Basses fréquences

  >> zones homogènes, continues

- Hautes fréquences

  >> détails, contours

L'image est un signal bidimensionnel non stationnaire. Les propriétés statistiques changent spatialement

Transformée de Fourier Discrète 2D (**DFT**)

Image échantillonnée (M x N) pixels, la DFT est donnée par :

$$X(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m,n] \exp(-2j\pi(\frac{mu}{M} + \frac{nv}{N}))$$

Transformation séparable :

- TFD 1D calculé en ligne sur x ➔ X_{ligne}
- TFD 1D calculé en colonne sur X_{ligne} ➔ X(u,v)

Variation sinusoïdale ± rapide

des niveaux de gris

dans une direction donnée

$$x(m,n) = Asin(2\pi f_m m + 2\pi f_n n) + \varphi_m + \varphi_n)$$

Images sinusoïdales → Impulsions de Dirac

Haute fréquence

Basse fréquence

Exemple TFD 2D

$$\left(\frac{1}{2\Delta}, \frac{1}{2\Delta}\right)$$

Si la période d'échantillonnage spatiale est $T = \Delta$

- $f_e = \frac{1}{2\Delta}$

- la bande observable va jusqu'à $f = \frac{1}{2\Delta}$ dans chacune des 2 directions

Exemple TFD 2D

➸ Propriétés de la DFT 2D

- Identiques au 1D
- Périodique en u,v (période M,N)
- X(0,0) = composante continue = moyenne des NG
- Conservation de l 'énergie ➘ $\Sigma\Sigma\ |x(m,n)|^2 = \Sigma\Sigma\ |X(u,v)|^2$
- x réelle ➘ X symétrique conjuguée (mod. pair, arg. impair)
- Séparable : DFT 1D en ligne et en colonne
- Algorithme rapide (FFT) :
  - $(n - 1)^2$ produits complexes et $n(n - 1)$ sommes complexes
  - $(n / 2)(\log2(n) - 2)$ produits et $n\log2(n)$ sommes pour la FFT
  - ~ 100 fois plus rapide pour n=1024

➸ Echantillonnage & Aliasing

• Si le théorème de **Shannon** n'est pas respecté lors de l'échantillonnage d'une image continue, il y a repliement de spectre

• Ceci se traduit dans les images par des figures de Moiré, des formes fausses qui n'existaient pas dans l'image d'origine

• Les caméras matricielles types CCD induisent systématiquement du repliement de spectre. L'image d'entrée ne devra donc pas contenir trop de hautes fréquences (*Ne passez pas à la télé avec un costume rayé !*)

Transformée **Cosinus** Discrète

$$C(u,v) = \frac{4.c(u).c(v)}{M.N}.\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} x(i,j).\cos\left(\frac{(2i+1)u.\pi}{2N}\right).\cos\left(\frac{(2j+1)v.\pi}{2M}\right)$$

$$\text{Avec}\begin{cases} c(u) = \sqrt{2/N} & \text{pour } u \neq 0 \\ c(u) = \sqrt{1/N} & \text{pour } u = 0 \end{cases}$$

Image originale

Transformée Cosinus Discrète

TCD de l'image

DCT par bloc

Lena

DCT par bloc 8x8

Fonctions de base DCT 2D 8x8

Pour calculer la valeur moyenne

Fonction de base de fréquence max. dans les 2 directions

Projection orthogonale de blocs 8x8 d'image sur ces fonctions de base (cf. MAS)

➸ Propriétés de la DCT 2D

- Linéaire, séparable
- **Coefficients réels**
- C(0,0) = composante continue = moyenne des NG
- Concentration d ’énergie en basse-fréquence
- Algorithme rapide (via la FFT) : $N/2.\log_2 (N)$

➢ compression d ’images (JPEG, MPEG)

f) Décomposition et analyse en sous-bandes / ondelettes

- Esteban/Galland 1977 - Woods/O 'Neil 1986 - … - Mallat (1989)
- Filtres **FIR 1D**, 2D

- Décomposition / Reconstruction sans pertes

Pyramidale
(itérée en octave)

Adaptative

- Réversible
- Concentration d ’énergie

- Spatio - fréquentiel

➢ Analyse & Compression & débruitage

g) Représentations de la couleur

- RGB
- Y C_b C_r

- Rouge Vert Bleu (RGB)

- Perception humaine dans le visible

Rouge Vert Bleu (RGB)

- Couleur primaire
- Perception par synthèse additive
- Objectif : perception d ’une source :
  - Moniteur, Carte graphique …
- Technique de pointillisme (G. Seurat, 1889)

- sur les écrans

➸ Rouge Vert Bleu (RGB) : redondance des 3 canaux

- plus un canal est clair, plus la couleur correspondante est active (présente)

Changement d'espace couleur

- Chaque image RGB est transformée dans un espace Y-Cb-Cr

$$\begin{cases} Y = 0.3R + 0.6V + 0.1B \\ Cb = \dfrac{B - Y}{2} + 0.5 \\ Cr = \dfrac{R - Y}{1.6} + 0.5 \end{cases}$$

L'idée est de décorréler l'information redondante contenue dans RGB et obtenir un espace dans lequel on sépare la luminance (sous espace d'une dimension) et la chrominance (sous-espace de deux dimensions)

YUV (PAL) / YIQ (NTSC)

- Objectif : distinguer luminance et chrominance
- Y = intensité lumineuse = TV N&B
- UV / IQ = information chrominance

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & -0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.522 & 0.311 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- YUV décorrélation de l 'information meilleure que RGB

➢ **Compression d 'images couleur**

$$\begin{bmatrix} R:33.2 \\ V:36.2 \\ B:30.6 \end{bmatrix} \leftrightarrow \begin{bmatrix} Y:93 \\ U:5.3 \\ V:1.7 \end{bmatrix}$$

Exemples de transformations couleur

III.2 Pré-traitements & Amélioration

- a. Opérations pixel à pixel
- b. Opérations sur un voisinage : filtrage
- c. Transformations géométriques

- Pourquoi pré-traiter une image ?
  - ➸ Corriger les effets de la chaîne d ’acquisition
    - Correction radiométriques et/ou géométriques
    - Réduire le bruit : Restauration, Déconvolution
  - ➸ Améliorer la visualisation
  - ➸ Améliorer les traitements ultérieurs (segmentation, compression …)

a) Opérations pixel à pixel

- Modification d'un pixel indépendamment de ses voisins

- Histogramme des niveaux de gris
  - Comptage des pixels ayant un niveau de gris (NG) donné
  - Histogramme ↘ densité de probabilité des niveaux de gris

Modifications d 'histogramme

- Transformation des niveaux de gris : f

  - *b=f(a)* *avec a niv. gris de départ, b niv.gris d'arrivée*

  - *f* peut prendre une forme quelconque

• Recadrage linéaire des niveaux de gris

- **Seuillage binaire** >> segmentation

- **Négatif**

• Egalisation d'histogramme

b) Opérations sur un voisinage : filtrage

- Modification d'un pixel en fonction des ses voisins

- Filtrage linéaire
  - Domaine spatial : filtres FIR 2D (masque), filtres IIR
  - Domaine fréquentiel dans le plan de Fourier

$$y(i,j) = h(i,j) * x(i,j) \quad \text{(convolution bidimensionnelle)}$$

$$Y(u,v) = H(u,v) \,.\, X(u,v)$$

- Filtrage non-linéaire dans le domaine spatial

Filtrage spatial FIR 2D : masque de convolution

- Convolution par une réponse impulsionnelle finie appelée ***Masque de Convolution***

$$y(i,j) = \sum_{(k,l)\in W}\sum h(k,l)x(i-k,j-l)$$

x est l'image de départ
h est le masque de convolution
W défini un voisinage

- Un pixel *x(i,j)* est remplacé par une somme pondérée de lui-même et des pixels de son voisinage

- <u>Exemple</u> : Filtre moyenneur

$$h(k,l) = \left\{ \begin{array}{c} \dfrac{1}{(2M+1)(2N+1)} \; si \; -M < k < M \; et \; -N < l < N \\ 0 \; sinon \end{array} \right\}$$

W: voisinage 2x2 ↘ k=0,1 l=0,1

$h(k,l)$ = 1 /4 pout tout (k,l)

l \ k	0	1
0	1/4	1/4
1	1/4	1/4

0	1	2	2
1	1	2	1
1	2	0	0

→

3/4	6/4	7/4	x
5/4	5/4	3/4	x
x	x	x	x

→

En arrondissant à la valeur entière la plus proche

1	2	2	x
1	1	1	x
x	x	x	x

Filtre moyenneur (lissage) le plus utilisé : filtre 3x3

$$1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(zoom)

Filtre moyenneur (lissage) le plus utilisé : filtre 3x3

Image originale

Image originale + bruit gaussien

Image lissée par un filtre 3x3

- Remarques
  - Utilisation de voisinages très divers
    - Rectangulaires 2x2, 3x3, 4x4, 5x5, 7x7, 1x2, 2x1, 1x3, 3x1...
    - En croix, «Circulaires»...
  - Gestion des bords : zéro padding, périodisation (FFT), symétrie …
  - Filtrage séparable, tps de calcul …
  - Valeurs des coefficients
    - Constants(Moyenneur), Gaussiens…
  - Effets de filtrage passe-bas : image plus «flou», contours moins précis mais réduction du bruit haute fréquence

  - *Le principe du masque de convolution sera utilisé pour d'autres traitements (Détection de contours)*
  - *L'utilisation d'un voisinage entourant un pixel est un principe très général en traitement de l'image*

➸ Filtrage passe haut et détection de contours

- Pour chaque pixel *(i,j)*, on mesure le gradient dans deux directions orthogonales : D_x *et* D_y ➔ vecteur gradient $\vec{\nabla} f = D_x \vec{x} + D_y \vec{y}$

- Calcul de l'amplitude du gradient max. : $M = \sqrt{D_x^{\,2} + D_y^{\,2}}$

- Calcul de la direction du gradient max. : $\vec{\nabla} f = \theta = \mathrm{Arctan}\left(\dfrac{D_y}{D_x}\right)$

➸ Convolution avec les filtres FIR : masques

<u>Masques de convolution, exemples :</u> H_1 H_2

Roberts $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Prewitt $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$

Sobel $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

➸Exemple de détection de contours

« Roberts »

128x128

H_1

Amplitude

Zoom

(inhomogénéité du contour)

H_2

Direction

blanc=$\pi/4$

...

gris = $\pi + \pi/4$

H_1

Amplitude

H_2

Direction

Il existe de nombreuses méthodes de détection de contour:

- Dérivation au premier ordre
  Prewitt, Sobel, Roberts, Kirsh, Compass, dérivateurs...
- Dérivation au second ordre
  Laplacien, Marr et Hildreth,...
- Filtrage optimal
  Canny-Deriche, Shen
- Modélisation des contours
  Hueckel, Haralick
- Morphologie mathématique
  gradient morphologique, ligne de partage des eaux...

Caractéristiques:
Complexité, précision de localisation, sensibilité au bruit, création de faux contours

- <u>Exemple</u> : rehaussement de contours

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix} = \text{Image d'origine} + \text{Laplacien} \begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix}$$

?

Filtrage non linéaire 2D : filtre Médian

- Remplacer le pixel central par la valeur médiane du voisinage

bruit poivre et sel

Comparaison avec le filtrage moyenneur

bruit poivre et sel

Filtrage moyenneur

Filtrage médian

- Avantage par rapport au filtrage linéaire
  - ↘ les bords sont conservés

III.3. Compression

- a. Introduction
- b. Approches directes
- c. Approches par transformation
- d. Compression de séquences d'images

a) Introduction

Objectifs

Réduction du volume occupé par les images numériques pour faciliter leur transfert et/ou leur stockage

Historique

- 1952 : Codeur entropique (Huffman)
- 1978 : DCT (Pratt)
- 1980 : Vectoriel (Linde-Buzo-Gray)
- 1986 : Sous-bandes (Woods)
- 1986 : Vectoriel sur treillis (Fisher)
- 1989 : JPEG
- 1989 : MPEG-2
- 1989 : Ondelettes (Mallat, Daubechies)
- 1990 : Fractales (Jacquin)
- 1996 : SPIHT
- 1996 : MPEG-4
- 1997 : MPEG-7
- 1998 : $JPEG_{2000}$

Applications

- Imagerie médicale → Télémédecine
- Imagerie spatiale
- Imagerie sous-marine
- Archivage divers (Musée, BNF, Empreintes ...)
- Vidéo conférence / visiophone (64 kb/s)
- Télésurveillance
- Video On Demand
- Télévision numérique (150 Mb/s)

...

Classification des méthodes de compression

image → **Transformation** → **Quantification** → **Codage** → bits

❶ Sans pertes / avec pertes contrôlées

- ❑ **Sans pertes** (Huffman, Quadtree)
  - image originale = image comprimée ➪ TC limité (#3)
- ❑ **Avec pertes** contrôlées
  - On perd l'information qui se voit peu ➪ TC augmente
  - Recherche d'un **compromis** Tc / Qualité

❷ Directe / Transformation

- ❑ **Directe** ➪ Quantification & codage des pixels de l'image
- ❑ **Transformation** ➪ Quantification & codage des coeff. transformés

Evaluation d'une méthode compression

➪ **Dépend de l'application**

- **Taux de compression (Tc)**

$$Tc = \frac{\text{Volume image originale}}{\text{Volume du fichier comprimé}}$$

- **compression ratio**
- **bitrate : 1/compression ratio**

Ex : image (512x512x8bpp) avec Tc=10
➪ 512x512x8/10=26215 bits → 0.8 bpp

- **Qualité**
  - Critère mathématique (RSB)

$$RSB_{dB} = 10\log_{10}\left(\frac{\left(2^b - 1\right)^2}{EQM(X,\hat{X})}\right)$$

Avec l'erreur quadratique moyenne : $EQM(X,\hat{X}) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\frac{\left(X_{ij} - \hat{X}_{ij}\right)^2}{M.N}$

  - Critères subjectifs
    - Courbes ROC (médecine)
    - Notations subjectives (TV)

Codage sans perte

- Codage Huffman
- Codage arithmétique
- Codage type dictionnaire

} Codeurs de source (Th. Information)

- Codage par longueur de plage

➸ Codage par **longueur de plage** (Run length coding RLC)

⇨ Coder le nombre de symboles identiques

00000111110000000000000000	⇨	5w5b17w
00000000000111111000000000	⇨	11w5b11w
A B C C C C C C A B C A B C	⇨	A B !6C A B C A B C

- CCITT, Fax groupe III
  - ↳ Huffman sur les plages de 0 précédant les 1
- JPEG
  - ↳ Huffman sur les plages de 0 précédant les coeff. DCT

Méthodes de codage sans perte : synthèse

- RLC (Run Length Coding)

- Codage Huffman : au moins 1 bit par symbole
  - ☞ proba=0,9 ➔ information = 0,15 bits, mais au moins 1 bit avec Huffman

- Codage arithmétique : moins d'1 bit par symbole, mais sans mémoire, peut être contextuel

- Codage à base de dictionnaire : modèle avec mémoire : LZW

c) Approches par transformation

☺ Représentation différente de l'image

⇨ Décorrélation ⇨ Gain en performances

☹ Temps de calcul supplémentaire

- Une Transformation
  - ⇨ **Réversible** (sans perte)
  - ⇨ **Orthogonale** (énergie conservée)
  - ⇨ **Rapide**

➸ DCT ⇨ JPEG

➸ Ondelettes ⇨ SPIHT, JPEG2000

➸ Compression DCT bloc : **JPEG** (1989)

- DCT bloc 8x8
  - ⇨ Pixels corrélés dans un voisinage 8x8 ==> obtenir des coefficients DCT quasi décorrélés (blanchiment)
  - ⇨ Information concentrée sur très peu de coefs DCT
  - ⇨ Coefficients concentrés majoritairement en BF
  - ⇨ Calcul rapide, besoin de peu de mémoire

Fonctions de base DCT 2D 8x8

Pour calculer la valeur moyenne

Fonction de base de fréquence max. dans les 2 directions

Projection orthogonale de blocs 8x8 d'image sur ces fonctions de base (cf. MAS)

Produits scalaires entre le bloc 8x8 et fonctions de base

- Calculons le coefficient DCT(4,2)

- Produit scalaire 2D :

•< , > = DCT(4,2)

- Schéma général

- Matrice de **normalisation**
  - ⇨ allocation des bits aux coeffs avant quantification par **arrondi**

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Matrice luminance

Matrice chrominance

$$\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

- Lecture **zig-zag**
  ⇨ prise en compte de la répartition spatiale de l'énergie pour faire apparaître de longues plages de coeffs nuls

- Codage du coeff DC
  ⇨ DPCM d'ordre 1 + Huffman

JPEG : codage des coefficients

26	-3	6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Plan DCT quantifié

- Séquence 1D après le balayage zigzag :

26 -3 1 -3 -2 6 2 -4 1 -4 1 1 5 0 2 0 0 1 2 0 0 0 0 0 -1 -1 EOB

- DC coeff. : prédiction avec le bloc voisin
- AC coeffs. : codage du symbole (plages, amplitude)

- Codage des coefficients AC

  ⇨ Codage hybride : runlength de zéros + amplitude du coefficient non nul qui suit la plage de zéros

  - Huffman = Code (plage de 0 + catégorie)

  162 codes : $10_{cat} x 16_{lp} + 2_{(EOB+16)}$

Cat.	Intervalle des coefficients AC
1	-1 _ 1,
2	-3, .. ,-2 _ 2, .. ,3
3	-7, .. ,-4 _ 4, .. ,7
4	-15, .. ,-8 _ 8, .. ,15
5	-31, .. ,-16 _ 16, .. ,31
6	-63, .. ,-32 _ 32, .. ,63
7	-127, .. ,-64 _ 64, .. ,127
8	-255, .. ,-128 _ 128, .. ,255
9	-511, .. ,-256 _ 256, .. ,511
10	-1023, .. ,-512 _ 512, .. ,1023

AC ⇨ | huffman | signe | k-1 bits |

- Exemple

  0 -2 -1 0^2 -1 0^{57} ⇨ **111001** 0 0 / **00** 0 / **11011** 0 / **1010**

- Extrait de la table d'Huffman des AC

Plage de Zéros	Catégorie	Code
0	1	00
0	2	01
0	3	100
0	4	1011
.	.	.
1	1	1100
1	2	111001
1	3	1111001
1	4	111110110
.	.	.
2	1	11011
2	2	11111000
.	.	.
3	1	111010
.	.	.
16		11111010
EOB		1010

Examples of rate/distortion with JPEG

Original image

CR=10, PSNR=34.9 dB

Examples of rate/distortion with JPEG

CR=32, PSNR=30.25 dB

CR=60, PSNR=27.3 dB

- Remarques
  - ⇨ JPEG = méthode générale → à adapter ...
  - ☺ Très performant à taux faibles (#10)
  - ☹ Effets de blocs à taux élevés
  - ☹ On ne maîtrise pas la taille du fichier binaire

Tc = 20 / RSB = 28.7 dB

Traitement des images couleur

L'œil est de l 'ordre de trois fois plus sensible aux variations de luminance par rapport aux variations de chrominance

Traitement des images couleur

- Chaque image RVB est transformée dans un espace luminance - chrominance Y-Cb-Cr

$$\begin{cases} Y = 0.3R + 0.6V + 0.1B \\ Cb = \dfrac{B - Y}{2} + 0.5 \\ Cr = \dfrac{R - Y}{1.6} + 0.5 \end{cases}$$

- Les composantes de chrominance sont sous-échantillonnées par 2 dans les deux directions (4:1:1)

Traitement des images couleur

- Différentes qualités avec JPEG

90 % quality JPEG

Image erreur

- Différentes qualités avec JPEG

25 % quality JPEG

Image erreur : significant JPEG artifacts are visible around all edges. Most appear as ripples and echo lines

d) Compression de séquences d'images

- Supprimer la redondance spatiale ou intra-image

  >> Approches 2D

- Supprimer la redondance temporelle ou inter-image

  >> Utiliser le « déjà vu » et le mouvement

Evolution des normes de codage vidéo

- MPEG 1 (1988 – 92) : 1,5 Mbs : qualité VHS
- MPEG 2 (1990 – 94) : qualité DVD à 8 Mbs
- H.264 (Advanced Video Coding AVC (2003) :
  - ☞ qualité DVD à 2 Mbs au lieu de 8 Mbps (MPEG2)
  - ☞4 times more complexe than MPEG2
  - ☞ format Play Station 3 ou Blu-Ray
  - ☞adaptée au format Full HD (1920 x 1080)

Evolution des normes de codage vidéo

- H.265 (High Efficiency Video Coding HEVC) (2013)
  - ☞ Même qualité que H.264 avec un débit divisé par 2
  - ☞ Mais bcp plus complexe que H.264
  - ☞ adaptée pour des images de grande taille, 4K (4096 × 2160) (cinéma) ou UHD (3840 x 2160) (TV) jusqu'à 8K (8192 x 4320)
  - ☞ ce codec permet à la vidéo HD de se généraliser sur tous les terminaux (tablettes et smartphones) et toutes les plateformes (TNT, Internet, VoD)
  - ☞ H.264 : 9 GO à stocker pour 2h de vidéo Full HD (1920 x 1080)
  - ☞ H.265 : 5 GO à stocker pour la même vidéo

Evolution des normes de codage vidéo

- H.266 (Versatile Video Coding VVC) (juillet 2020)
  - ☞ En 2020, 80% du trafic internet : de la vidéo
  - ☞ Fraunhofer heinrich hertz institute et apple ericsson intel huawei microsoft qualcomm sony
  - ☞ - 50% en débit que H.265 pour la même qualité
  - ☞ Sans codage : 2h de vidéo Full HD (1920 x 1080 x 8 bits x 50 Hz) ==> pour une transmission temps réel, il faut un canal de 1,24 Gbs
  - ☞ Avec codage H.266 : un canal de 5Mbs suffit (0,4% des données originales à transmettre)
  - ☞ Standard propriétaire, donc payant
  - ☞ Toutes ces normes ont une base commune : **MPEG 1, TCD, travail par bloc**

MPEG 1

- 3 types d'images : 3 codages

- 2 paramètres de réglage
  - N : distance inter-I (#12)
  - M : distance inter-P (#3)

GOP

MPEG 1 : Group of Planes GOP

- Distance between P : m=3 ; distance between I : n=12
- 1 et 13 (I) are reference images I
- 4 is predicted from 1, 7 is predicted from 4
- 2 is predicted from 1 and 4
- Transmission order : 1 4 2 3 7 5 6 10 8 9 13 11 12 16 14 ...

- Estimation du mouvement par block matching
  - Blocs 8x8
  - Compromis simplicité / efficacité
  - Rapide : algorithme logarithmique

➸ Le codage des images **P**

1- Calcul toutes les distances entre blocs père et le bloc fils

2- Choisir la distance minimale ➔ choix bloc « vrai père »

3- Calcul du bloc erreur entre le bloc vrai père et le bloc fils

4- Code binaire est formé de :

$$\begin{cases} \text{bloc père connu } \hat{I}(n-1) : \text{déjà connu} \\ \text{vecteur de translation entre père et fils (codé par DPCM)} \\ \text{bloc erreur codé par JPEG *} \end{cases}$$

Codage de chaque type d'image

- Images **I** (intra)
  - Codées JPEG'
  - Point d'accès séquence (0.5s)
  - Tc faible
- Images **P** (Prédites)
  - Prédites à partir de I ou P
  - Codage DPCM des vecteurs mvt
  - Codage JPEG* de l'erreur de prédiction
  - Tc élevé
  - Propagation de l'erreur
- Images **B** (Bidirectionnelles)
  - Interpolées à partir des I P
  - Tc le plus élevé

Block based bidirectional interpolation

Gain du à Motion Compensation

Image N

Différence entre image N et N-1

Différence après compensation de mouvement

Motion Compensation : minimisation erreur de prédiction père - fils

Norm H.264

- H.264 : standard for video compression (2003)
- H.264 is equivalent to MPEG-4 Part 10, or MPEG-4 AVC (for Advanced Video Coding)
- **des varies nouveautés par rapport à MPEG1 et MPEG2**
- Bit rate 2-3 times lesser than MPEG2
  - ☞ **DVD quality at 2 Mbps instead of 8 Mbps (MPEG2)**
  - ☞ 4 times more complexe than MPEG2
- Adopted per Sony for Play Station 3
- Blu-Ray format uses H.264

Norm H.264 : coding scheme

Advanced Video Coding (AVC)

Norm H.264 : decoding scheme

Norme H.264 : intra prediction

Original macroblock

4x4 luma block to be predicted

Norme H.264 : intra prediction

9 prediction modes for 4x4 luminance blocks

Norme H.264 : intra prediction

Criterium: minimize the sum of the absolute values of errors

Norme H.264 : intra prediction

Image source d'origine

Image intra-prédite

Image résiduelle

Image finale

Norme H.264 : inter prediction

Partition with macroblocks

Subpartition of 8x8 partitions

- **a motion vector for each partition or subpartition**
- **this partition scheme allows to adapt the coding process to the local image content and leads to a energy reduction for error blocks**

Norme H.264 : inter prediction

example of partition

Norme H.264 : « deblocking » filter

- **Vertical et horizontal filtering:**
  - ☞ **to reduce the blocking effects**
  - ☞ **to obtain less energetic residual blocks when applying block matching**
  - ☞ **without filtering and loosing the real contours**

original image

decoded image without filtering

decoded image with filtering

Norme H.264 : « deblocking » filter

III.4 Ouvertures

- a. Introduction à la segmentation
- b. Introduction à l'indexation
- c. Introduction au watermarking

a) Introduction à la segmentation

✓La segmentation est un traitement bas-niveau qui consiste à créer une **partition** de l'image I en sous-ensembles R_i appelés **régions** telles que :

$$\forall_i \quad R_i \neq 0$$

$$\forall_{i,j} \; ; i \neq j \;\; R_i \cap R_j = 0$$

$$I = \bigcup_i R_i$$

✓Une région est un ensemble de pixels connexes ayant des **propriétés communes** qui les **différencient** des pixels des régions voisines.

Segmentation techniques

- Adaptive Thresholding
- Contour Approaches
  - Differential Methods
    - Contours detection + Contours closing
  - Active Contours
- Template Matching
- Region Approaches
  - Texture
    - Analysis and classification
  - Markov Methods
  - Structural Approaches

Exemples d'images segmentées par 2 approches

Contour (edge) segmentation :
Approche contour

Region segmentation :
Approche région

Segmentation contour : dérivées ➔ contours

Segmentation région : seuillage de l'histogramme

<u>Idea:</u> If each object present in the image has an uniform and distinct color, then pics related to the objects will appear in the histogram.

Segmentation par seuillage adaptatif d'histogramme

- Détection de vallées, en prenant le minimum de l'histogramme situé entre les 2 pics
- Optimisation du seuil S par modélisation Gaussienne $p_1(x)$ et $p_2(x)$ et en minimisant l'expression basée sur les fonctions de répartition

Exemple : Méthode Fisher ou méthode OTSU

Objectif : Trouver le seuil S qui minimise la somme des moments centrés d'ordre 2 (somme des Variances) des 2 classes

h(x) : histogramme de l'image

Centre de gravité G d'une classe

$$G_i(S) = \frac{\sum_{x \in C_i} x h(x)}{\sum_{x \in C_i} h(x)}$$

Variance Var d'une classe

$$Var_i(S) = \sum_{x \in C_i} (x - G_i)^2 h(x)$$

Trouver S qui minimise la somme des variances :

$$S_{opt} = \underset{S}{Min}\left(Var_1(S) + Var_2(S)\right)$$

En simplifiant les termes en carrés, cela revient à maximiser la fonctionnelle J(S) :

x varie de 0 à S

x varie de (S+1) à 255

$$J(S) = \frac{\left(\sum_{x \in C_1} xh(x)\right)^2}{\sum_{x \in C_1} h(x)} + \frac{\left(\sum_{x \in C_2} xh(x)\right)^2}{\sum_{x \in C_2} h(x)}$$

Le problème de seuillage ou de partitionnement revient à chercher S dans {0,255} qui maximise J(S)

Approche région par partitionnement quadtree
SPLIT & MERGE

Structure de l'arbre

Exemple

Activité du bloc :
- si variance > seuil, alors on ”split”

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 1

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Example of splitting process

Example of merging process

Each color represent a merged region

What means to segment an image?

Partitioning the image in homogeneous areas related to a certain criteria: color, texture, gray level, motion,…

And then…
Indexing each region

Indexation d'une région

- Modéliser les caractéristiques texture, couleur, forme, mouvement d'une région
  - ☞ texture raillée, jaune, circulaire, sans mouvement
  - ☞ construire un vecteur caractéristique : une sorte de signature de la région
  - ☞ autrement dit : indexer la région

Rechercher des régions similaires

- Etant donnée une base d'images :
  - ☞ on segmente chaque image
  - ☞ chaque région est indexée par un vecteur caractéristique de N dimensions
  - ☞ création d'une base de vecteurs caractéristiques
- Etant donnée une région requête (mot clé) :
  - ☞ on mesure la similarité entre le vecteur requête et les vecteurs de la base
  - ☞ par exemple, la distance quadratique
  - ☞ on ordonne par ordre croissant, les distances obtenues
  - ☞ si on veut 10 régions similaires dans la base, on prend les 10 distances les plus petites
  - ☞ Image retrieval (en anglais)

Rechercher des régions similaires

- Les régions peuvent être sans sémantique
- Les régions peuvent être des visages détectés
  - ☞Reconnaissance de visages
- Les régions peuvent être des humains en mouvement
  - ☞Détection et reconnaissance d'actions humaines
- Une fois les vecteurs calculés, on est dans un espace à N dimensions
  - ☞Un vecteur : un point dans l'espace N dimensions
  - ☞Des millions de vecteurs : un nuage de points
  - ☞Problème d'analyse et de classification de données

Morphologie mathématique

Objectifs

- Inspiré des problèmes de traitement d'images, domaine qui constitue son principal champ d'application
- Etudier ou de traiter un ensemble (ici : image) à l'aide d'un autre ensemble de géométrie connue : appelé **élément structurant**
- Propriétés : la non-linéarité ; la non-inversibilité
- Très souvent appliquée sur des images binaires
- Applications : filtrage, amélioration de la qualité, segmentation, remplissage des trous, fermeture des contours

Schéma bloc

Image binaire originale
Morpho
Math.
éléments structurants :
Géométrie et taille à choisir
(une sorte de sonde)
Image érodée

Dilatation

- Hypothèse : fond noir, objet blanc (ici, jaune)
- Lorsque l'intersection de l'élém @TC1 Bon WE, à la prochaine séance. ent structurant avec l'objet blanc n'est pas vide, le centre de l'élément structurant prend la valeur « 1 » (donc le centre

- Rép. positive : centre = objet
- Rép. négative : centre = fond

Dilatation

Image binaire originale

Image après dilatation

Dilatation

- Bouche les trous plus petits que l'élément structurant
- Soude les formes proches
- Elargit les pointes
- Comble les canaux étroits
- Augmente la taille des objets

Erosion

- Hypothèse : fond noir, objet blanc (ici, jaune)
- L'érosion est l'opération duale de la dilatation
- Lorsque l'élément structurant n'est pas complétement inclus dans l'objet blanc, le centre de l'élément structurant est enlevé de l'objet

- Rép. positive : centre = objet
- Rép. négative : centre = fond

Erosion : rétrécissement de l'objet

Image binaire originale

Image après érosion

Erosion

- Elimine les composantes connexes plus petites que l'élément structurant
- Elimine les pointes étroites
- Elargit canaux et trous
- Transforme une presqu'île en île
- Diminue la taille des objets