

# Quantification Vectorielle

Marco Cagnazzo

Département Traitement du Signal et des Images  
TELECOM ParisTech

14 Décembre 2012

# Plan

- 1 Introduction
- 2 Construction du dictionnaire
- 3 Performances
- 4 Techniques de QV
  - QV gain-forme

# Plan

- 1 Introduction
- 2 Construction du dictionnaire
- 3 Performances
- 4 Techniques de QV
  - QV gain-forme

## Rappel : Quantification scalaire (QS)

### Définition

$$Q : x \in \mathbb{R} \rightarrow y \in \mathcal{C} = \{\hat{x}^1, \hat{x}^2, \dots, \hat{x}^L\} \subset \mathbb{R}$$

- Puissance de l'erreur de quantification (cas idéal)

$$\sigma_Q^2 = c_X \sigma_X^2 2^{-2R}$$

- $c_X$  est une constante qui dépend de la distribution de  $X$  (facteur de forme)
  - $c_X = 1$  dans le cas uniforme
  - $c_X = \frac{\sqrt{3}}{2} \pi$  dans le cas gaussien

# Rappel : Quantification scalaire prédictive (QP)

## Performances

- On quantifie la différence  $Y$  entre la valeur de  $X$  et la prédiction linéaire basée sur  $P$  échantillons précédents
- Gain de prédiction : rapport entre la puissance de l'erreur sans et avec prédiction.

$$G_P(P) = \frac{\sigma_X^2}{\sigma_Y^2} = \frac{\sigma_X^2}{[\det \mathbf{R}_P]^{1/P}}$$

- Valeur asymptotique du gain de prédiction

$$G_P = \lim_{P \rightarrow +\infty} \frac{\sigma_X^2}{[\det \mathbf{R}_P]^{1/P}}$$

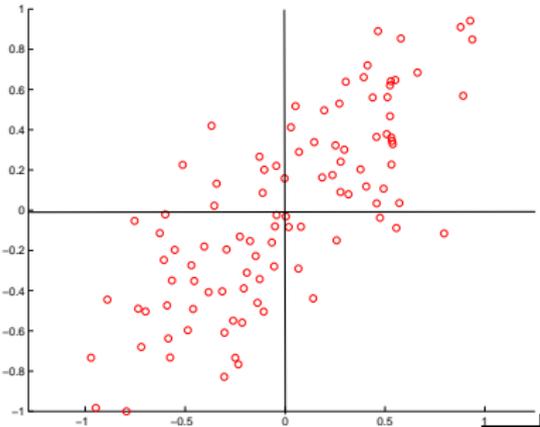
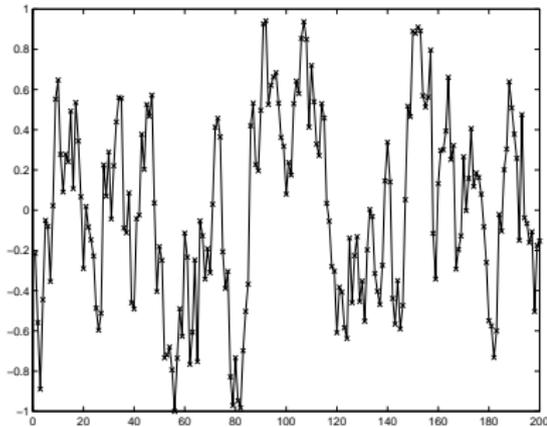
# Quantification vectorielle (QV)

## Introduction

- La quantification est normalement utilisée sur *séquences* d'échantillons
- Idée : quantifier conjointement plusieurs échantillons (un *vecteur*)
- L'espace  $\mathbb{R}^N$  est donc partitionné en régions de décision
- À chaque région on fait correspondre un représentant (un vecteur)

# Exemple: signal corrélé

## Signal AR(2)



# Quantification vectorielle

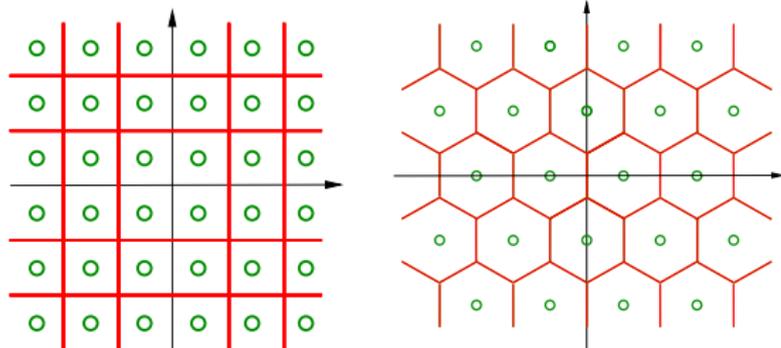
## Introduction

- Généralisation de la QS à l'espace  $\mathbb{R}^N$
- La QS est un cas particulier de la QV
  - Les régions de décisions sont délimité par des hyperplans orthogonaux aux axes
  - Les représentants sont alignés parallèlement aux axes
- Gains potentiels de la QV
  - Gain géométrique : forme arbitraire des régions de décision et position arbitraire des représentants
  - Gain de corrélation : on exploite directement la corrélation du signal

# Propriétés de la VQ

La VQ permet des améliorations par rapport à la SQ :

- on peut exploiter la corrélation entre échantillons
- on supprime la contrainte de cellules hypercubiques
- on supprime la contrainte de débits entiers



# Formalisme

Définition de la quantification vectorielle :

$$Q : \underline{x} \in \mathbb{R}^N \rightarrow \underline{y} \in \mathcal{C} = \{\hat{\underline{x}}^1, \hat{\underline{x}}^2, \dots, \hat{\underline{x}}^L\} \subset \mathbb{R}^N$$

- Dimension de l'espace :  $N \in \mathbb{N}$
- Taille du dictionnaire :  $L \in \mathbb{N}$
- Résolution (ou débit) de la QV :  $b = \frac{\log_2 L}{N} = \frac{R}{N}$

# Formalisme

$$Q : \underline{x} \in \mathbb{R}^N \rightarrow \underline{y} \in \mathcal{C} = \{\hat{\underline{x}}^1, \hat{\underline{x}}^2, \dots, \hat{\underline{x}}^L\} \subset \mathbb{R}$$

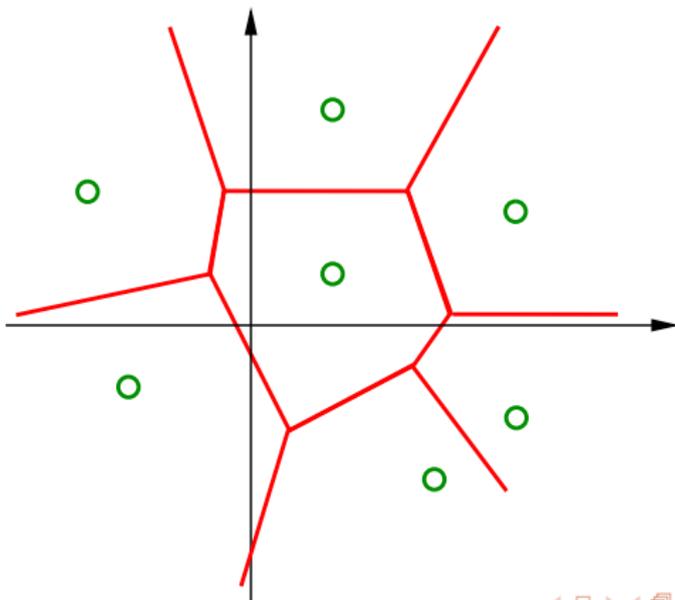
- Régions de décision (ou cellules)

$$\Theta^i = \{\underline{x} : Q(\underline{x}) = \hat{\underline{x}}^i\} \quad \Theta^i \cap \Theta^j = \emptyset \quad \cup_i \Theta^i = \mathbb{R}^N$$

- $\mathcal{C}$  est appelé dictionnaire
- Les éléments de  $\mathcal{C}$  sont appelés *représentants*, *vecteurs de sortie*, ou *vecteur de reproduction*
- On considère uniquement des quantificateurs *réguliers* :
  - Convexité :  $\underline{x}_1, \underline{x}_2 \in \Theta^i \Rightarrow [\lambda \underline{x}_1 + (1 - \lambda) \underline{x}_2 \in \Theta^i] \forall \lambda \in [0, 1]$
  - Représentant dans la région :  $\forall i \in \{1, \dots, L\}, \hat{\underline{x}}^i \in \Theta^i$

# Quantificateur régulier

- Régions convexes
- $\hat{\underline{x}}^i \in \Theta^i$

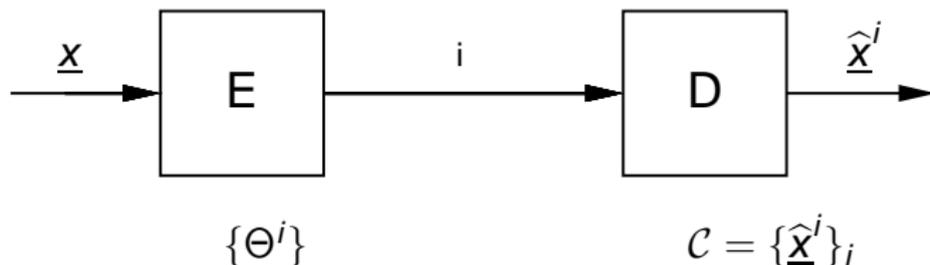


# Quantification vectorielle

## VQ comme codage et décodage

On peut la voir comme suite de deux opérations :

- *Codage* : Association entre  $\underline{x}$  et l'index  $i$  (la région  $\Theta^i$ )
- *Décodage* : Association entre l'index  $i$  (la région  $\Theta^i$ ) et le mot de code  $\hat{\underline{x}}^i$



## Erreur de quantification

- On généralise directement la définition de la QS
- On utilise la densité de probabilité conjointe et la norme euclidienne

$$\begin{aligned}\sigma_Q^2 &= \frac{1}{N} \int_{\mathbb{R}^N} \|\underline{x} - Q(\underline{x})\|^2 p_{\underline{X}}(\underline{x}) d\underline{x} \\ &= \frac{1}{N} \sum_{i=1}^L \int_{\Theta^i} \|\underline{x} - Q(\underline{x})\|^2 p_{\underline{X}}(\underline{x}) d\underline{x} \\ &= \frac{1}{N} \sum_{i=1}^L \int_{\Theta^i} \|\underline{x} - \hat{\underline{x}}^i\|^2 p_{\underline{X}}(\underline{x}) d\underline{x}\end{aligned}$$

# Plan

- 1 Introduction
- 2 Construction du dictionnaire
- 3 Performances
- 4 Techniques de QV
  - QV gain-forme

# Construction du dictionnaire

Problème : Déterminer le dictionnaire  $\mathcal{C}$  minimisant l'erreur  $\sigma_Q^2$

Solution : Algorithme de Lloyd généralisé (GLA)

- 1 Initialisation :  $k = 0$ , on choisit un certain dictionnaire  
 $\mathcal{C}_{(k)} = \{\hat{\mathbf{X}}_{(k)}^0, \hat{\mathbf{X}}_{(k)}^1, \dots, \hat{\mathbf{X}}_{(k)}^L\}$ , on assigne  $\sigma_{Q,(k-1)}^2 = \infty$ , on choisit  $\epsilon$
- 2 Optimiser les cellules par rapport au dictionnaire : on obtient  $\{\Theta_{(k)}^i\}$
- 3 Optimiser le dictionnaire par rapport aux cellules : on obtient  $\mathcal{C}_{(k+1)}$
- 4 Calculer la distorsion  $\sigma_{Q,(k)}^2$  associée à  $\mathcal{C}_{(k)}$ ,  $\{\Theta_{(k)}^i\}_i$
- 5 Si  $\frac{\sigma_{Q,(k-1)}^2 - \sigma_{Q,(k)}^2}{\sigma_{Q,(k-1)}^2} < \epsilon$  on arrête, sinon  $k \leftarrow k + 1$  et on revient au pas 2

# Algorithme de Lloyd généralisé (GLA)

- Optimiser les cellules par rapport au dictionnaire :

$$\Theta_{(k)}^i = \left\{ \underline{\mathbf{x}} : \|\underline{\mathbf{x}} - \hat{\underline{\mathbf{x}}}_{(k)}^i\| \leq \|\underline{\mathbf{x}} - \hat{\underline{\mathbf{x}}}_{(k)}^j\| \forall j \in \{1, \dots, L\} \right\}$$

- Règle du plus proche voisin
- Complexité proportionnelle à  $L$

## Algorithme de Lloyd généralisé (GLA)

- Optimiser le dictionnaire par rapport aux cellules : règle du centroïd
- Pour simplicité de notation,  $D = \sigma_{Q,(k)}^2$

$$D = \frac{1}{N} \sum_{i=1}^L \int_{\Theta^i} \|\underline{x} - \hat{\underline{x}}^i\|^2 p_{\underline{X}}(\underline{x}) d\underline{x} = \frac{1}{N} \sum_{i=1}^L D_i$$

$$\frac{\partial D_i}{\partial \hat{x}^{i,j}} = \int_{\Theta^i} 2(x^j - \hat{x}^{i,j}) p_{\underline{X}}(\underline{x}) d\underline{x}$$

$$\hat{x}^{i,j} = \frac{\int_{\Theta^i} x^j p_{\underline{X}}(\underline{x}) d\underline{x}}{\int_{\Theta^i} p_{\underline{X}}(\underline{x}) d\underline{x}}$$

$$\hat{\underline{x}}^i = E[\underline{X} | \underline{X} \in \Theta^i]$$

# Algorithme de Lloyd généralisé (GLA)

## Bilan

- Nécessité d'estimer les distributions de probabilité
- Complexité très élevée : algorithmes suboptimaux
- L'algorithme ne converge qu'à un optimum locale : problème de l'initialisation
- Algorithmes de recuit simulé (très complexes, optimum globale avec une probabilité donnée, dans certaines hypothèses)

# Initialisation du GLA

## Algorithme Linde-Buzo-Gray (LBG)

- Algorithme itératif
- Pour  $L = 1$ , application du GLA
- Le résultat  $\hat{\underline{x}}_0$  est le centroïde de la population
- **Split** de  $\mathcal{C}_0 = \{\hat{\underline{x}}_0\}$ :  $\hat{\underline{x}}_1 = \hat{\underline{x}}_1 \quad \hat{\underline{x}}_0 + \epsilon$
- GLA sur le nouveau dictionnaire  $\mathcal{C}_1 = \{\hat{\underline{x}}_1, \hat{\underline{x}}_1\}$
- **Split** de  $\mathcal{C}_1$  : chaque mot de code  $\hat{\underline{x}}_1^i$  donne lieu à  $\hat{\underline{x}}_2^{2i} = \hat{\underline{x}}_1^i$   
et  $\hat{\underline{x}}_2^{2i+1} = \hat{\underline{x}}_1^i + \epsilon_i$
- Itérer jusqu'à obtenir le nombre  $L$  de vecteurs

## Création d'un dictionnaire empirique

- Souvent on a pas les loi des probabilités des données
- Par contre on peut construire une *base d'apprentissage*
  - Morceaux de son, blocs de pixels pris des images, ...
- La base d'apprentissage est utilisée pour créer le dictionnaire :
  - *Plus proche voisin* : les vecteurs de la base sont associés à la cellule dont le représentant est le plus proche voisin
  - *Centroid* : dans chaque cellule on recalcule le centroid comme la moyenne des vecteurs associés

# Quantification scalaire et vectorielle



# Quantification scalaire

Débit 21 bpp PSNR 47.19 dB TC 1.143



# Quantification scalaire

Débit 18 bpp PSNR 42.38 dB TC 1.333



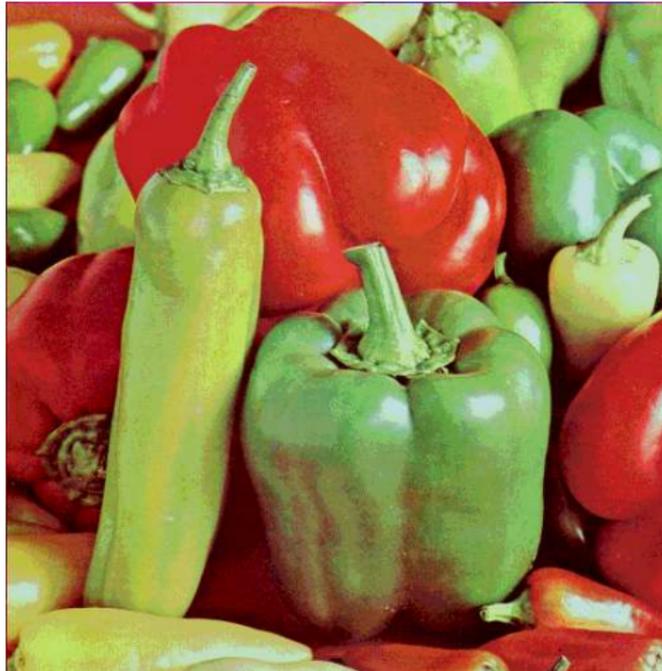
# Quantification scalaire

Débit 15 bpp PSNR 36.97 dB TC 1.600



# Quantification scalaire

Débit 12 bpp PSNR 31.40 dB TC 2.000



# Quantification scalaire

Débit 9 bpp PSNR 29.26 dB TC 2.667



# Quantification vectorielle

Débit 9.0 bpp PSNR 37.59 TC 2.667



# Quantification scalaire

Débit 6 bpp PSNR 27.83 dB TC 4.000



# Quantification vectorielle

Débit 6.0 bpp PSNR 33.00 TC 4.000



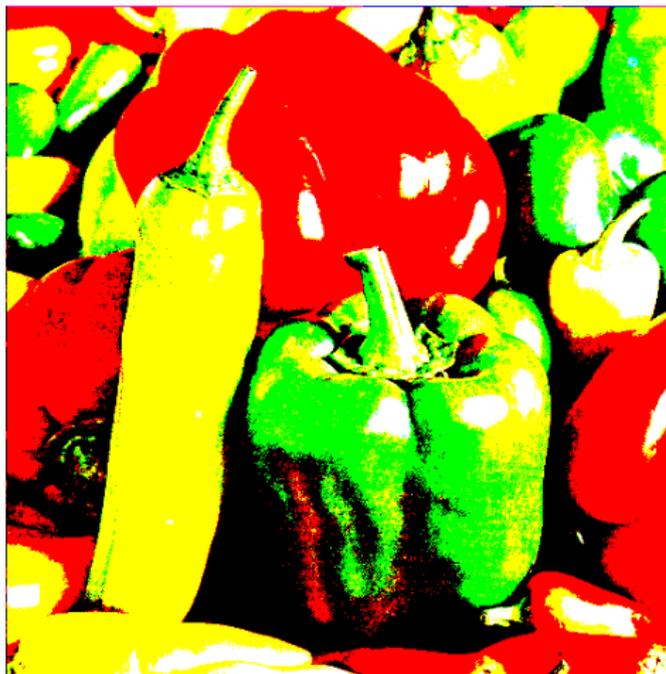
# Quantification vectorielle

Débit 4.5 bpp PSNR 30.78 TC 5.333



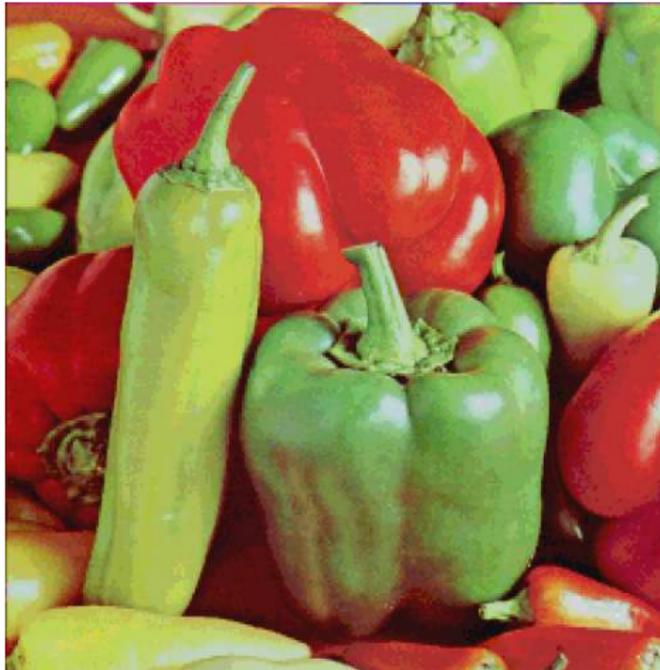
# Quantification scalaire

Débit 3 bpp PSNR 25.75 dB TC 8.000



# Quantification vectorielle

Débit 3.0 bpp PSNR 27.63 TC 8.000



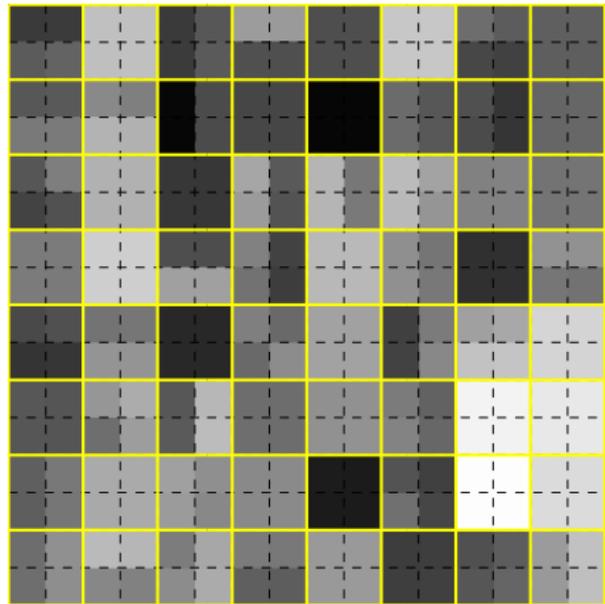
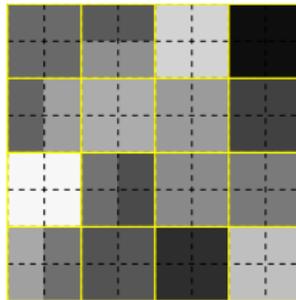
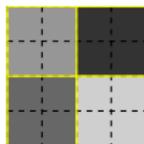
# Quantification vectorielle

Débit 1.5 bpp PSNR 21.41 TC 16.000



# Exemples de dictionnaires

$$N = 4, L \in \{4, 16, 64\}$$



# Plan

- 1 Introduction
- 2 Construction du dictionnaire
- 3 Performances**
- 4 Techniques de QV
  - QV gain-forme

## Performances du quantificateur optimal

Généralisation de la formule de Bennet (scalaire)

$\sigma_Q^2 = \frac{1}{12} \left[ \int_{\mathbb{R}} p_X^{1/3}(x) dx \right]^3 2^{-2b}$  au cas vectoriel en  $N$  dimensions :

$$\sigma_Q^2(N) = \alpha(N) \left[ \int_{\mathbb{R}^N} p_{\underline{X}}^{\frac{N}{N+2}}(\underline{x}) d\underline{x} \right]^{\frac{N+2}{N}} 2^{-2b}$$

Cas gaussien:

$$\sigma_Q^2(N) = \alpha(N) \left[ \int_{\mathbb{R}^N} \left( \frac{\exp(\underline{x}^T \mathbf{R} \underline{x})}{(2\pi)^{N/2} \sqrt{\det \mathbf{R}}} \right)^{\frac{N}{N+2}} d\underline{x} \right]^{\frac{N+2}{N}} 2^{-2b}$$

# Performances du quantificateur optimal

## Cas gaussien

$$\begin{aligned}\sigma_Q^2(N) &= \alpha(N) \left[ \int_{\mathbb{R}^N} \left( \frac{\exp(\underline{x}^T \mathbf{R} \underline{x})}{(2\pi)^{N/2} \sqrt{\det \mathbf{R}}} \right)^{\frac{N}{N+2}} d\underline{x} \right]^{\frac{N+2}{N}} 2^{-2b} \\ &= \alpha(N) 2\pi \left( \frac{N+2}{N} \right)^{\frac{N+2}{2}} (\det \mathbf{R})^{\frac{1}{N}} 2^{-2b} \\ &= c(N) (\det \mathbf{R})^{\frac{1}{N}} 2^{-2b}\end{aligned}$$

On montre que

$$c(1) = \frac{\sqrt{3}}{2} \pi > c(N) > c(\infty) = 1$$

# Performances du quantificateur optimal

## Cas gaussien

Gain de la QV = EQM de la QS / EQM de la QV

$$\sigma_Q^2(N) = c(N)(\det \mathbf{R})^{\frac{1}{N}} 2^{-2b}$$

$$G_V(N) = \frac{\sigma_Q^2(1)}{\sigma_Q^2(N)} = \frac{c(1)}{c(N)} \frac{\sigma_X^2}{(\det \mathbf{R})^{\frac{1}{N}}}$$

Gain “géométrique” :  $\frac{c(1)}{c(N)}$

Gain de corrélation :  $\frac{\sigma_X^2}{(\det \mathbf{R})^{\frac{1}{N}}}$

# Performances du quantificateur optimal

## Gain "géométrique"

- Le rapport  $c(1)/c(N)$  est toujours supérieur à 1
- Ce terme est décroissant avec  $N$
- Cela montre que la QV est préférable à la QS même pour une source sans mémoire
- Tout de même, le gain géométrique maximum est limité par :

$$\lim_{N \rightarrow +\infty} \frac{c(1)}{c(N)} = c(1) = \frac{\sqrt{3}}{2}\pi$$
$$10 \log_{10} \lim_{N \rightarrow +\infty} \frac{c(1)}{c(N)} = 4.35\text{dB}$$

# Performances du quantificateur optimal

## Interprétation du gain "géométrique"

- Le gain géométrique est lié à la meilleure capacité de la QV de couvrir  $\mathbb{R}^N$  par rapport à la QS
- Idéalement, il faudrait avoir des cellules délimitées par des hypersphères : avec la QV on s'y approche
- La QS est contrainte à utiliser des cellules hypercubiques
- Les  $2^N$  sommets d'un hypercube ont une distance de  $\sqrt{N}\frac{L}{2}$  du centre ; le centre d'une hyperface a distance  $\frac{L}{2}$
- Donc à l'augmenter de  $N$ , l'hypercube a de plus en plus des points très éloignée du centre : la cellule est de moins en moins une hypersphère

# Performances du quantificateur optimal

## Gain de corrélation

Le gain de corrélation a la même expression que dans le cas de quantification scalaire prédictive :

$$G_p = \frac{\sigma_X^2}{(\det \mathbf{R})^{1/N}}$$

Les 2 approches exploitent donc la même propriété :

La corrélation entre les échantillons du signal

Mais la QV a un marge de gain supérieur grâce au gain géométrique

# Performances du quantificateur optimal

## Limites théoriques de l'approche QV

- Pour un signal stationnaire, dès que  $N$  devient suffisamment grand,  $G_v > G_p$
- À l'augmenter de  $N$ , les performances de la QV se stabilisent
- La technique *ultime* de codage ?
- Toute technique de compression peut se voir comme une QV
- Toute technique peut être améliorée par le GLA ?

# Plan

- 1 Introduction
- 2 Construction du dictionnaire
- 3 Performances
- 4 Techniques de QV
  - QV gain-forme

# Utilisation du quantificateur

## Conception du dictionnaire

- Choix des paramètres : débit, taille du dictionnaire, dimension

$$b = \frac{\log_2 L}{N} \qquad L = 2^{bN}$$

- Le débit  $b$  est lié à la qualité ciblée et aux contraintes de mémoire (stockage) et de largeur de bande (transmission)
- La dimension  $N$  influence les performances : pour atteindre les meilleures performances  $N$  doit être la valeur maximum telle que  $x(n)$  et  $x(n + N)$  soient encore corrélés
- La complexité croît exponentiellement avec  $N$
- La taille  $L$  du dictionnaire croît exponentiellement avec  $b$  et  $N$

# Utilisation du quantificateur

## Processus de quantification

- On dispose d'un dictionnaire de taille  $L$  et dimension  $N$
- On regroupe  $N$  échantillons du signal d'entrée dans un vecteur  $\underline{x}$  de dimension  $N$ 
  - $N$  échantillons consécutif d'un son
  - $N$  pixels d'une image (typiquement un bloc rectangulaire, pour exploiter la redondance horizontale et verticale)
- **Quantification** : on cherche dans le dictionnaire le meilleur représentant de  $\underline{x}$ 
  - Règle du plus proche voisin
  - $L$  calculs de distance, chacun avec complexité  $O(N)$
  - Au total  $O(N2^{bN})$

# Utilisation du quantificateur

## Décodage

- Le quantificateur envoie le numéro du vecteur dans le dictionnaire
- Le décodeur récupère le représentant dans le dictionnaire
- Processus très asymétrique : codeur complexe, décodeur simple

# Utilisation du quantificateur

Exemple : codage de la voix

- Contraintes
  - Bande passante 8 kbps
  - Fréquence d'échantillonnage : 8 kHz
  - Capacité de calcul :  $C = 1 \text{ Gflops} = 10^9$  opérations par seconde
- Trouvez  $N$  et  $L$

# Utilisation du quantificateur

Exemple : codage de la voix

- Résolution = Bits/s divisé par échantillons/s

$$b = \frac{8\text{kb/s}}{8\text{kS/s}} = 1\text{b/S}$$

- Opérations par vecteur :  $NL = N2^N$
- Opérations par échantillon :  $2^N$
- Opérations par seconde :  $2^N f_E$
- Contrainte :  $2^N f_E < C$

$$2^N [\text{Op/S}] \times 8000 [\text{S/s}] < 10^9 [\text{Op/s}]$$

$$N < \log_2 \frac{10^9}{8 \times 10^3} = 6 \log_2 10 - 3 = 16.93$$

$$N = 16 \quad L = 65536$$

# Utilisation du quantificateur

Exemple : codage de la voix

- $N = 16$  est insuffisant
- Le signal de voix est trop complexe pour être représenté sur  $2^{16}$  vecteurs
- La “fenêtre d’analyse” a une durée trop courte :

$$T = N/f_E = 16/8000 = 2\text{ms}$$

- La corrélation inter-fenêtre reste importante : on atteint pas le gain maximum de la QV

# Utilisation du quantificateur

Exemple : codage de la voix

- Il faut augmenter  $N$  (et  $L$ ) sans augmenter la complexité de l'encodeur
- Cela est possible au détriment des performances débit/distorsion en imposant une structure au dictionnaire
- La structure permet de effectuer l'encodage sans calcul des  $L$  distances, mais en exploitant la structure

## Quantification vectorielle arborescente

- Technique pour trouver le plus proche voisin
- La recherche est organisée en niveaux, à chaque niveau on élimine la moitié des candidats (arbre binaire)
- A chaque étape, on compare le vecteur en entrée avec 2 représentants des 2 moitiés du dictionnaire
- En suite considère seulement la moitié choisie du dictionnaire et on itère jusqu'à réduire le dictionnaire à un seul vecteur
- Complexité : de  $O(2^{bN})$  à  $O(bN)$
- Performances très proches du cas *full search*

# Quantification vectorielle par produit cartésien

- On décompose un vecteur de taille  $N$  en  $k$  sous-vecteurs
- L' $i$ -ème vecteur a taille  $N_i$ , en sorte que  $\sum_{i=1}^k N_i = N$
- Complexité :  $\sum 2^{bN_i}$
- Exemple :  $N = 10$ ,  $b = 1$ ,  $C = 2^{10} = 1024$
- $N_1 = 5$ ,  $N_2 = 5$ ,  $C = 2^5 + 2^5 = 64$
- On perd la possibilité d'exploiter les dépendances statistiques qui existent entre les parties des vecteurs
- Si on arrive à décomposer le vecteur en sorte d'éliminer ces dépendances on peut utiliser ce type de QV

# Quantification vectorielle multi-étage

- On procède par approximations successives
- On réalise une première quantification
- En suite on quantifie l'erreur de quantification
- On continue itérativement

# Quantification vectorielle par transformée

- La transformée permet de concentrer l'information d'un bloc de données en peu de coefficients
- Cela permettrait de réduire la dimension  $N$ , car on a réduit la corrélation des données

# Quantification vectorielle algébrique

- Le dictionnaire n'est plus construit avec GLA
- Il est indépendant des statistiques de la source
- L'espace  $\mathbb{R}^N$  est reparti en manière régulière
- On parle de VQ sur réseaux
- Avantage : grande réduction de la complexité, pas besoin de mémoriser le dictionnaire (structure géométrique)
- Désavantage : perte de performance débit/distorsion

# Quantification vectorielle gain-forme

- Type de QV largement utilisé en codage audio
- Permet de gérer l'évolution de la puissance au cours du temps
- On peut la voir comme un cas particulier de QV par produit cartésien
- Le vecteur est décomposé en gain et forme, et les deux sont quantifié séparément

# Quantification vectorielle gain-forme

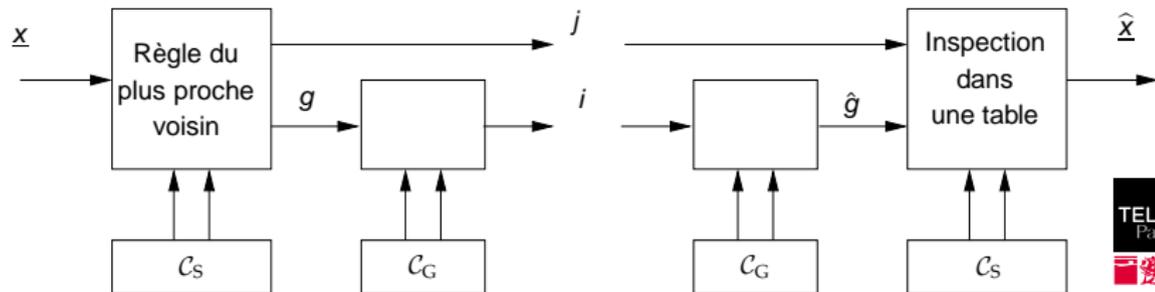
## Règle du plus proche voisin

- On utilise deux dictionnaires
  - Dictionnaire des formes (normés) :

$$\mathcal{C}_S = \{\hat{\mathbf{x}}^1, \hat{\mathbf{x}}^2, \dots, \hat{\mathbf{x}}^{L_1}\}$$

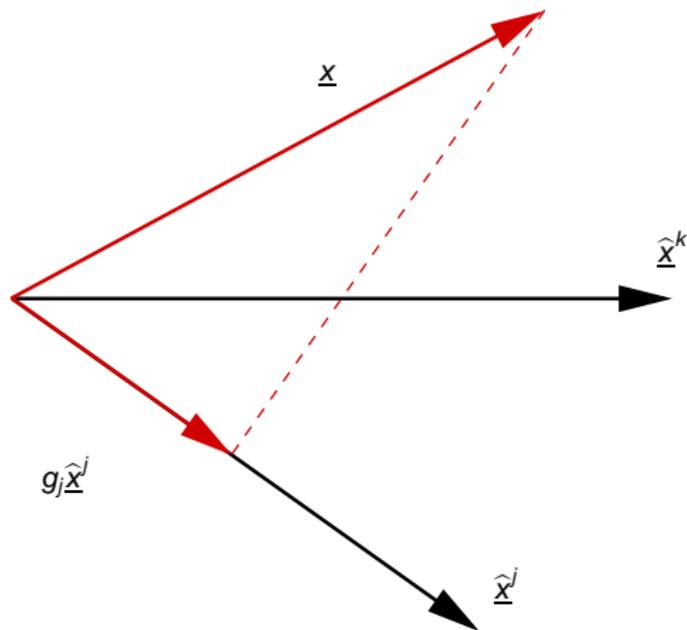
- Dictionnaire des gains :

$$\mathcal{C}_G = \{\hat{g}^1, \hat{g}^2, \dots, \hat{g}^{L_2}\}$$



# Quantification vectorielle gain-forme

## Règle du plus proche voisin



# Quantification vectorielle gain-forme

## Règle du plus proche voisin

- On recherche d'abord la forme la plus proche de  $\underline{x}$
- Il est possible de faire cela sans calculer le gain

$$\langle \underline{x} - g_j \hat{\underline{x}}^j, \hat{\underline{x}}^j \rangle = 0 \quad g_j = \frac{\langle \underline{x}, \hat{\underline{x}}^j \rangle}{\|\hat{\underline{x}}^j\|^2}$$

$$\|\underline{x} - g_j \hat{\underline{x}}^j\|^2 = \|\underline{x}\|^2 + g_j^2 \|\hat{\underline{x}}^j\|^2 - 2g_j \langle \underline{x}, \hat{\underline{x}}^j \rangle = \|\underline{x}\|^2 - \frac{\langle \underline{x}, \hat{\underline{x}}^j \rangle^2}{\|\hat{\underline{x}}^j\|^2}$$

$$\begin{aligned} j^* &= \arg \max_j \frac{\langle \underline{x}, \hat{\underline{x}}^j \rangle^2}{\|\hat{\underline{x}}^j\|^2} = \arg \max_j \left\langle \underline{x}, \frac{\hat{\underline{x}}^j}{\|\hat{\underline{x}}^j\|} \right\rangle^2 \\ &= \arg \max_j |\cos \phi_j| \end{aligned}$$

# Quantification vectorielle gain-forme

## Application du GLA au cas GS-VQ

- Dictionnaire des formes :  $\mathcal{C}_S = \{\hat{\underline{x}}^1, \hat{\underline{x}}^2, \dots, \hat{\underline{x}}^{L_1}\}$
- Vecteurs normés
- Base d'apprentissage :  $M$  vecteurs  $\underline{x}(m)$  avec  $i \in \{1, \dots, M\}$
- GLA : Plus proche voisin et centroïd
- La règle du plus proche voisin, comme l'on vient de voir, permet de répartir les vecteurs de la base en cellules
- Sans perte de généralité, la  $j$ -ème cellule est

$$\Theta^j = \{\underline{x}(m)\}_{m \in \{1, \dots, M(j)\}}$$

# Quantification vectorielle gain-forme

## Application du GLA au cas GS-VQ

- Pour optimiser le  $j$ -ème représentant, nous cherchons le vecteur  $\hat{\underline{x}}^j$  qui minimise l'erreur par rapport à la cellule  $\Theta^j$ , sous contrainte de norme unitaire

$$\arg \min_{\hat{\underline{x}}^j} \sigma_Q^2(j) \quad \text{s.c.} \|\hat{\underline{x}}^j\|^2 = 1$$

$$\sigma_Q^2(j) = \frac{1}{M(j)} \sum_{m=1}^{M(j)} \|\underline{x}(m) - g_j \hat{\underline{x}}^j\|^2$$

$$\begin{aligned} \|\underline{x}(m) - g_j \hat{\underline{x}}^j\|^2 &= \|\underline{x}(m)\|^2 - \frac{\langle \underline{x}(m), \hat{\underline{x}}^j \rangle^2}{\|\hat{\underline{x}}^j\|^2} \\ &= \|\underline{x}(m)\|^2 - \langle \underline{x}(m), \hat{\underline{x}}^j \rangle^2 \end{aligned}$$

# Quantification vectorielle gain-forme

## Application du GLA au cas GS-VQ

- Le problème devient

$$\arg \max_{\hat{\underline{x}}^j} Q = \sum_{m=1}^{M(j)} \langle \underline{x}(m), \hat{\underline{x}}^j \rangle^2 \quad \text{s.c.} \|\hat{\underline{x}}^j\|^2 = 1$$

- On appelle  $\Gamma$  la matrice de covariance empirique

$$\Gamma = \sum_{m=1}^{M(j)} \underline{x}(m)^T \underline{x}(m)$$

- On trouve :

$$(\hat{\underline{x}}^j)^T \Gamma \hat{\underline{x}}^j = \sum_{m=1}^{M(j)} (\hat{\underline{x}}^j)^T \underline{x}(m)^T \underline{x}(m) \hat{\underline{x}}^j = Q$$

# Quantification vectorielle gain-forme

Application du GLA au cas GS-VQ

- Le problème devient

$$\arg \max_{\hat{\underline{x}}^j} (\hat{\underline{x}}^j)^T \Gamma \hat{\underline{x}}^j \quad \text{s.c.} \|\hat{\underline{x}}^j\|^2 = 1$$

- On utilise la méthode des multiplicateurs de Lagrange
- On doit minimiser :

$$J = (\hat{\underline{x}}^j)^T \Gamma \hat{\underline{x}}^j - \lambda [(\hat{\underline{x}}^j)^T \hat{\underline{x}}^j - 1]$$

- On a

$$\frac{\partial J}{\partial \hat{\underline{x}}^j} = 0 \quad \Gamma \hat{\underline{x}}^j = \lambda (\hat{\underline{x}}^j)$$

# Quantification vectorielle gain-forme

## Application du GLA au cas GS-VQ

- Donc  $\hat{\underline{x}}^j$  est un vecteur propre de  $\Gamma$ , et  $\lambda$  est la valeur propre correspondante
- Si on multiplie  $\Gamma \hat{\underline{x}}^j = \lambda(\hat{\underline{x}}^j)$  par  $(\hat{\underline{x}}^j)^T$  on trouve:

$$(\hat{\underline{x}}^j)^T \Gamma \hat{\underline{x}}^j = \lambda (\hat{\underline{x}}^j)^T \hat{\underline{x}}^j$$
$$Q = \lambda$$

- Donc le problème d'optimisation est résolu en prenant le vecteur propre qui correspond à la plus grande valeur propre de la matrice de autocovariance empirique