Quantification vectorielle par emboîtage d'une hiérarchie de réseaux réguliers de points

Vincent Ricordel, Claude Labit

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Quantification vectorielle par emboîtage d'une hiérarchie de réseaux réguliers de points

VINCENT RICORDEL, CLAUDE LABIT

N° 2667

Octobre 1995

PROGRAMME 4

ISSN 0249-6399

Quantification vectorielle par emboîtage d'une hiérarchie de réseaux réguliers de points

Vincent Ricordel*, Claude Labit**

Programme 4 — Robotique, image et vision
Projet Temis

Rapport de recherche n° 2667 — Octobre 1995 — 71 pages

Résumé : Nous proposons de décrire la conception d'un nouveau schéma de quantification vectorielle destiné à prendre place au sein d'une chaîne de codage pour la compression de séquences d'images animées.
L'innovation de notre approche, qui doit permettre une construction rapide du dictionnaire, repose sur la coopération bénéfique de deux techniques déjà éprouvées séparément : la quantification vectorielle algébrique avec la mise en oeuvre de réseaux réguliers de points (treillis), l'édification par apprentissage et suivant un critère débit-distorsion d'un dictionnaire arborescent non-équilibré. Précisément, pour concevoir notre quantificateur, nous mettons en place une hiérarchie multigrille de treillis de même nature à résolution emboîtée. Nous décrivons donc le choix du treillis, puis la construction de cet ensemble hiérarchique de réseaux emboîtés, et enfin l'utilisation de cette hiérarchie dans un schéma simple de quantification. Deux algorithmes pour la construction du dictionnaire arborescent non-équilibré sont détaillés : l'un d'élagage et l'autre de découpage de l'arbre.
Un treillis tronqué et un arbre incomplet caractérisent le dictionnaire obtenu, la transmission de ce dernier ne requiert donc l'envoi d'aucun vecteur représentant.

(Abstract: pto)

*. IRISA e-mail ricordel@irisa.fr
**. IRISA e-mail labit@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Vector quantization by hierarchical packing of embedded truncated lattices

Abstract: The purpose of this study is to introduce a new vector quantizer (VQ) which takes place in a temporal-adaptative coding scheme for the compression of digital image sequences.
Our approach, which has to perform a fast codebook construction, unify both efficient coding methods : a fast lattice encoding and an unbalanced tree-structured codebook design according to a distortion vs. rate tradeoff. Moreover, this tree-structured lattice vector quantizer (TSLVQ) has a convenient property : because of its lattice structure, no reproduction vectors have to be transmitted.
Briefly speaking, the TSLVQ is based on the hierarchical packing of embedded truncated lattices. We investigate its design : by, first, explaining how to determine the support lattice and secondly how to obtain the hierarchical set of truncated lattice structures which can be optimaly embedded involving the hierarchical packing. We then give the simple quantization procedure and we describe the corresponding tree-structured codebook. Finally we present two unbalanced tree-structured codebook design algorithms based on the BFOS distorsion vs. rate criterion.

Table des matières

1 Introduction

Nous pourrions simplifier en désignant les quantificateurs comme les convertisseurs analogiques-numériques utilisés dans les appareils électroniques tels que les dispositifs de mesure, les systèmes d'enregistrement, ceux de communication. En effet, ces derniers ne manipulent que des données numériques ayant été mises en forme par un quantificateur.
Si nous nous plaçons plus précisément dans le domaine du codage de sources, en plus d'une représentation du signal, souple et adaptée aux traitements, la quantification permet une compression supplémentaire de l'information ceci au prix d'une perte que l'on peut contrôler. Cette technique se prête donc particulièrement au codage d'images qu'il faut comprimer pour les transmettre ou les archiver.

Ce rapport se divise en deux grandes parties.
Dans la première nous rappelons ce qu'est la quantification vectorielle et pourquoi elle demeure supérieure à celle scalaire. Ensuite nous passons en revue les principales méthodes déjà utilisées en quantification vectorielle et nous décrivons le contexte de l'étude : la compression de séquences d'images animées.
Dans la seconde partie nous présentons un nouveau quantificateur : le quantificateur vectoriel par emboîtage d'une hiérarchie de réseaux de points réguliers. Nous décrivons tout d'abord les outils mathématiques utilisés pour ensuite introduire les techniques mises en jeu. Enfin des exemples de quantification de sources synthétiques ainsi que d'une source réelle sont donnés, et ces premiers résultats sont analysés.

2 La quantification vectorielle (QV)

2.1 Définitions

La **quantification** consiste en l'approximation d'un signal d'amplitude continue par un signal d'amplitude discrète.

La **quantification vectorielle** [20] [18] (notée **QV**) consiste alors à représenter tout vecteur $\mathbf{x}$ de dimension k par un autre vecteur $\mathbf{y}_i$ de même dimension mais ce dernier appartenant à un ensemble fini D de L vecteurs. Les $\mathbf{y}_i$ sont appelés les **vecteurs représentants**, les **vecteurs de reproduction** ou les **code vecteurs**. D est appelé le **dictionnaire** ou le **catalogue des formes**.

Il n'y a rien de mystérieux à considérer des espaces de grandes dimensions, pour mieux appréhender les raisonnements il suffit de s'avoir que tout s'organise autour des coordonnées des vecteurs, qu'il n'y a pas lieu de s'imposer une représentation mentale géométrique. Pour l'illustrer nous précisons qu'un **vecteur x** de l'espace $\mathbb{R}^k$ est simplement une matrice colonne constituée de k nombres réels x_i : $\mathbf{x} = (x_1, x_2, ..., x_k)^T$, et que par exemple, une sphère entièrement caractérisée par son centre $\mathbf{u} = (u_1, u_2, ..., u_k)^T$ son rayon ρ est constituée des points dont les coordonnées satisfont : $\sum_{i=1}^{k}(x_i - u_i)^2 = \rho^2$.

Un **Quantificateur Vectoriel de dimension k et de taille L** peut-être défini mathématiquement comme une application Q de $\mathbb{R}^k$ vers D :

$$\begin{array}{ccccc} Q & : & \mathbb{R}^k & \longmapsto & D \\ & & \mathbf{x} & & Q(\mathbf{x}) = \mathbf{y}_i \end{array}$$

avec

$$D = \left\{ \mathbf{y}_i \in \mathbb{R}^k / i = 1, 2, ..., L \right\}$$

Cette application Q détermine implicitement une **partition** de l'espace source $\mathbb{R}^k$ en L régions C_i. Ces régions encore appelées **classes** ou **régions de Voronoï** sont déterminées par :

$$C_i = \left\{ \mathbf{x} \in \mathbb{R}^k / Q(\mathbf{x}) = \mathbf{y}_i \right\}$$

Les conditions suivantes sont satisfaites :

$$\bigcup_{i=1}^{L} C_i = \mathbb{R}^k$$

$$C_i \bigcap C_j = \emptyset, \quad i \neq j, \quad \begin{array}{l} \forall i = 1, 2, ..., L \\ \forall j = 1, 2, ..., L \end{array}$$

Dans le but d'alléger les notations, l'abréviation QV est utilisée pour désigner selon le contexte, soit un quantificateur vectoriel, soit la quantification vectorielle.

2.2 QV appliquée à la compression de données

2.2.1 La compression

La **compression** ou **codage** de données vise à diminuer la quantité des éléments binaires nécessaire à la représentation de l'information contenue dans le signal à transmettre ou à archiver. Cette diminution peut autoriser ou non la perte d'information [32].

Principe : Soit $\{x_n\}$ l'ensemble des échantillons du signal source :

- **Coder** c'est déterminer une loi de codage $\mathbf{C}$ telle que, $\mathbf{C} : \{x_n\} \rightarrow \{y_n\}$,
- **Décoder** c'est définir une loi de décodage $\mathbf{D}$ telle que, $\mathbf{D} : \{y_n\} \rightarrow \{\hat{x}_n\}$.

Si $\mathbf{D}^{-1} = \mathbf{C}$, il s'agit d'un code sans perte d'information dit **réversible** (codage statistique ou entropique).
Si $\mathbf{D}^{-1} \neq \mathbf{C}$, il s'agit d'un code avec perte d'information dit **irréversible**.

Une bonne compression est réalisée si on réunit :

- la loi $\mathbf{C}$ minimisant le nombre d'éléments binaires y_n à transmettre,
- La loi $\mathbf{D}$ assurant une bonne reconstruction du signal source $\{x_n\}$ par les $\{\hat{x}_n\}$.

2.2.2 QV appliquée au codage

La quantification Vectorielle offre la combinaison des opérations de codage et de décodage.
En effet, considérons I l'**ensemble des L index** des vecteurs de reproduction $\mathbf{y}_i$ du dictionnaire :
$I = \{1, 2, ..., L\}$.

Alors la loi de codage est déterminée par :

$$\begin{array}{rccc} \mathbf{C} & : & \mathbb{R}^k & \longmapsto & I \\ & & \mathbf{x} & & \mathbf{C}(\mathbf{x}) = i \end{array}$$

Le processus de décodage est lui défini par :

$$\begin{array}{rccc} \mathbf{D} & : & I & \longmapsto & D \\ & & i & & \mathbf{D}(i) = \mathbf{y}_i \end{array}$$

Finallement nous obtenons : $Q = \mathbf{D} \circ \mathbf{C}$.

Un QV se décompose donc en deux applications: un codeur et un décodeur.

2.2.2.1 Le codeur

Le rôle du codeur consiste, pour tout vecteur $\mathbf{x}$ du signal d'entrée, à rechercher dans le dictionnaire D le code-vecteur $\mathbf{y}_i$ le plus proche de $\mathbf{x}$.

Nous introduisons ici la définition générale de la **métrique** L_p, où la norme d'un vecteur $\mathbf{x}$ de dimension k est :

$$L_p(\mathbf{x}) = ||\mathbf{x}||^p = \sum_{j=1}^{k} |x_j|^p$$

La distance entre 2 vecteurs $\mathbf{x}_1$ et $\mathbf{x}_2$ est alors :

$$\begin{aligned} ||\mathbf{x}_1 - \mathbf{x}_2||^p &= d_p(\mathbf{x}_1, \mathbf{x}_2) \\ &= \sum_{j=1}^{k} |x_{1j} - x_{2j}|^p \end{aligned}$$

La notion de proximité que nous avons mise en place est la **distance euclidienne**:

$$d(\mathbf{x}, \mathbf{y}_i) = \sum_{j=1}^{k} (x_j - y_{ij})^2$$

C'est donc le cas particulier de L_p où $p = 2$. Cette distance mesure la **distorsion** entre les vecteurs $\mathbf{x}$ et $\mathbf{y}_i$.

Les régions de voronoï sont alors données par :

$$C_i = \left\{ \mathbf{x} \in \mathbb{R}^k / Q(\mathbf{x}) = \mathbf{y}_i, \; si \; d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \forall j \neq i \right\}$$

Chaque région contient un ensemble de vecteurs de $\mathbb{R}^k$ et tous les vecteurs $\mathbf{x}$ qui appartiennent à C_i sont représentés par le même vecteur $\mathbf{y}_i$ du dictionnaire.

La compression d'information est réalisée à ce niveau car c'est uniquement l'index du représentant $\mathbf{y}_i$ minimisant le critère de distorsion qui sera transmit ou stocké au lieu d'un vecteur. Cette opération, qui perd de l'information, fait que la QV est une **opération irréversible**, le signal original ne pourra plus être restitué tel quel.

La quantité d'information requise pour représenter le vecteur source est donnée par R le **débit binaire** en **bits par composante** du vecteur, ou **bits par dimension**, ou encore **bits par échantillon**:

$$R = \frac{1}{k}. \log_2 L$$

2.2.2.2 Le décodeur

Le **décodeur** est considéré comme un récepteur voué à la reconstruction du vecteur source, pour cela il dispose d'une réplique du dictionnaire qu'il consulte afin de restituer le code vecteur correspondant à l'index qu'il reçoit. Le décodeur réalise donc l'opération de décompression.

2.2.2.3 Schéma général d'un QV

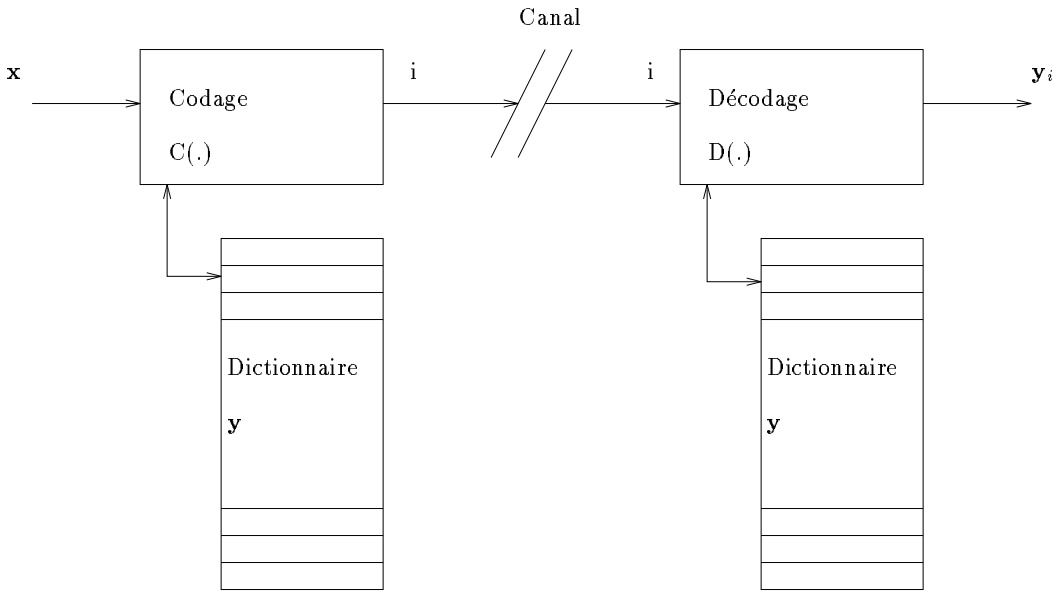

FIG. 1 - *Schéma d'un quantificateur vectoriel*

2.2.2.4 Principe de la QV

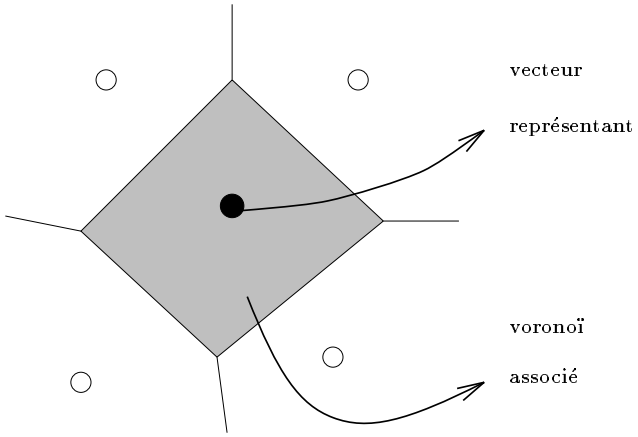

FIG. 2 - *Principe de la quantification vectorielle*

2.2.2.5 Evaluation des performances du système

Le but de tout système de codage est d'obtenir, avec un débit minimal, une distorsion moyenne caractérisant les performances globales du QV, elle aussi minimale. Il importe surtout que la mesure de distorsion mise en oeuvre traduise la dégradation subjective faite au signal. Seule une mesure traduisant les caractéristiques perceptuelles humaines peut apporter un tel résultat [32].

Précisément la mesure de distorsion doit réunir 3 conditions essentielles [21]. En effet cette fonction réelle positive doit-être :

- simple à calculer ;
- utilisable par un algorithme de minimisation;
- significative telle qu'une grande distorsion implique une mauvaise qualité de la restitution subjective du signal (et inversement).

Le choix d'une erreur quadratique moyenne (la mesure de distorsion moyenne la plus répandue qui s'interprète comme la puissance de l'erreur de quantification) permet de vérifier les 2 premières conditions, cependant elle ne caractérise que pauvrement la dégradation qualitative faite au signal. La troisième condition serait parfaitement obtenue au prix de l'introduction d'une pondération spychovisuelle adaptée au codage des séquences d'images. Cependant nous n'avons pas encore mis en place cette pondération.
Ainsi, à priori, notre travail présente la conception d'un QV non dédié au codage d'une source spécifique. Pour l'adapter au traitement de la parole ou de l'image, il suffit d'opter pour la mesure de distorsion la plus adéquate.

3 La quantification scalaire (QS)

3.1 La quantification scalaire uniforme

La **quantification scalaire** [18] (notée **QS**) est une forme particulière de la QV, celle où la dimension des vecteurs est égale à un.

Afin d'introduire différentes notions de bruits rencontrées en quantification, nous allons présenter le plus simple des quantificateurs : le quantificateur scalaire uniforme à débit fixe.

Ce type de quantificateur est entièrement déterminé par :

- les $L+1$ **niveaux de décisions** : $x_0, x_1, ..., x_L$, qui partitionnent en L intervalles égaux l'axe des réels $\mathbb{R}$ et déterminent le pas de quantification ;
- Les L **valeurs de reproduction** : $y_1, y_2, ..., y_L$, qui sont les centres de masse de chacun des intervalles de décision.

Le débit de ce quantificateur est donc donné par $R = \log_2 L$ [bits/échantillon].

Il apparait deux sortes d'erreurs ou bruits de quantification :

- le **bruit granulaire** qui se produit lorsque la valeur d'entrée x se situe dans l'une des cellules $[x_i, x_{i+1}]$, l'erreur résultante qui est la différence entre x et $Q(x)$ peut être majorée par un demi pas de quantification ;
- le **bruit de surcharge** ou de **dépassement** qui se produit lorsque la valeur d'entrée se situe hors de l'intervalle $[x_0, x_L]$. La valeur de reproduction est alors soit y_1 soit y_L, et l'erreur résultante supérieure à un demi pas de quantification.

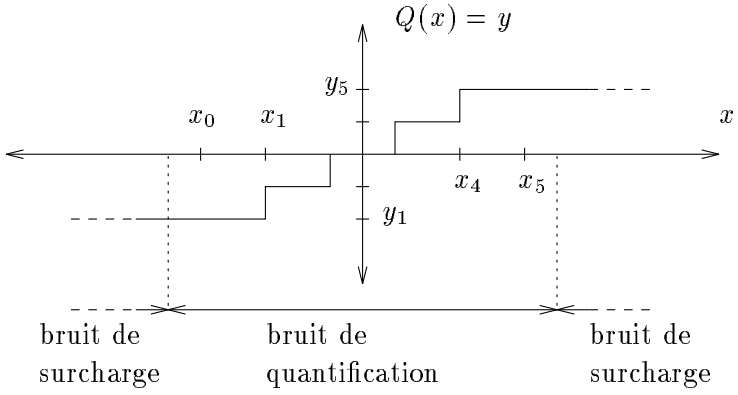

FIG. 3 - *Exemple d'un QS uniforme pour $L = 5$*

3.2 Le QS optimal

Le QS optimal est celui qui minimise, pour une source donnée et un débit fixé, l'erreur moyenne de reconstruction due aux bruits de quantification et de surcharge. Les niveaux de reconstruction sont donc répartis en tenant compte de la densité de probabilité de la variable à quantifier. Intuitivement nous comprenons que la concentration des niveaux de reconstruction est plus importante dans la zone de l'espace où la densité de probalité des vecteurs à quantifier est plus élevée. En fait ce problème d'optimisation à une solution unique et le QS obtenu est connu sous le nom de **quantificateur de Lloyd-Max** [28] [30].
En définissant le **gain en distorsion** comme étant le rapport, à débit fixe, des distorsions de deux quantificateurs, le quantificateur de Lloyd-Max offre, du fait de la répartition non uniforme de ses pas de quantification, un gain en distorsion par rapport aux autres quantificateurs scalaires. Cependant ce gain est loin de rejoindre le maximum annoncé par la théorie de l'information.

4 Eléments de la théorie de l'information

Les références bibliographiques relatives à cette partie du rapport sont [34] [25] [35] [6] [36] [7] [37] [23].

4.1 Entropie

Considérons une source à temps discret et ergodique $\{x(n)\}, n = 0, \pm 1, \pm 2, ...$
Le flux des **symboles** $x(n)$ forme une séquence aléatoire $\{X(n)\}, n = 0, \pm 1, \pm 2, ...$ dont les réalisations x_k appartiennent à l'**alphabet** de la source $A_k = \{x_k / k = 1, 2, ..., K\}$.
Si K est fini la source est dite à **amplitude discrète**, si K est infini la source est à **amplitude continue**.
La source est **sans mémoire** si les échantillons successifs sont statistiquement indépendants.

4.1.1 Source à amplitude discrète et sans mémoire

Soit P_k la probabilité d'occurence du symbole x_k : $P_k = Pr\{X(n) = x_k\}$.
Alors une appréciation numérique de l'information reçue est donnée par : $I(x_k) = -\log_2 P_k$ [en bits].
La quantité d'information moyenne reçue ou **entropie** est alors [en bits/échantillon] :

$$H(X) = E(I(X)) = -\sum_{k=1}^{K} P_k . \log_2 P_k$$

On a :

$$0 \leq H(X) \leq \log_2 K$$

- si $H(x) = 0$ la source est totalement **prédictible**,
- si $H(x) = \log_2 K$ la source est **non prédictible**, les symboles sont équiprobables.

$\log_2 K$ mesure la **capacité** de l'alphabet.
Les **redondances** de la source sont appréciées en calculant $R(X) = \log_2 K - H(X)$.

4.1.2 Source à amplitude discrète avec mémoire

Il existe alors des dépendances statistiques entre les échantillons successifs de la source.
Soit $\mathbf{x} = \mathbf{x}(n) = (x(n), x(n+1), ..., x(n+N-1))^T$, un vecteur constitué de N échantillons successifs de la source.
Ce vecteur est caractérisé par sa probabilité jointe $P(\mathbf{x})$ qui est indépendante du temps si nous considérons une source stationnaire.
Alors $\mathbf{x}$ est une réalisation du vecteur aléatoire $\mathbf{X} = (X(n), X(n+1), ..., X(n+N-1))^T$.

Soit l'**entropie des vecteurs** aléatoires [en bits/échantillon] :

$$H_N(\mathbf{X}) = \frac{1}{N} . E(-\log_2 P(\mathbf{X})) = -\frac{1}{N} . \sum \sum_{\mathbf{x}} ... \sum P(\mathbf{x}) . \log_2 P(\mathbf{x})$$

On a :

$$H(X) = \lim_{N \to \infty} H_N(\mathbf{X})$$

On considère aussi l'entropie conditionnelle d'un symbole à l'instant n étant donnés les $(N-1)$ symboles précédents : $H(X(n)/X(n-1), X(n-2), ..., X(n-N+1))$

Il est démontré que :

$$H(X(n)/X(n-1), X(n-2), ..., X(n-N+1)) \leq H_N(\mathbf{X})$$

et que :

$$\lim_{N \to \infty} H(X(n)/X(n-1), X(n-2), ..., X(n-N+1)) = H(X)$$

Les deux fonctions $H_N(\mathbf{X})$ et $H(X(n)/X(n-1), X(n-2), ..., X(n-N+1))$ sont décroissantes avec N.
De plus, pour deux sources ayant deux alphabets identiques et même probabilité pour les symboles, il est prouvé que :

$$(H(X) \text{ / source avec mémoire}) \leq (H(X) \text{ / source sans mémoire})$$

4.1.3 Codage sans perte d'une source à amplitude discrète

La théorie annonce qu'un codage sans perte d'information avec un débit binaire proche de l'entropie est réalisable pour une source à amplitude discrète. Ce codage est dit **entropique** ou à **longueur de code variable** (ex : codage de Huffman [22]). Pour présenter de façon intuitive ce code, nous pouvons expliquer que les statistiques du signal à coder sont exploitées en affectant les mots de code les plus courts aux représentants les plus fréquents. Ce code est efficace (c.a.d le débit proche de l'entropie) si les symboles de la source ont des probabilités qui puissent être approchées par des puissances négatives de deux. Pour avoir un tel résultat, il est souvent nécessaire de regrouper les échantillons et de considérer les probabilités des vecteurs obtenus, les procédures de codage deviennent alors plus sophistiquées.

4.1.4 Entropie d'un dictionnaire

Le dictionnaire, généré pour atteindre une distorsion moyenne donnée, est un exemple de source à amplitude discrète.

En reprenant les notations du paragraphe **2.2**, à chaque vecteur représentant de taille k est associé un index i parmi le L index de l'alphabet I. Chaque index i se présente comme un mot de code binaire c_i de longueur b_i bits.

L'**entropie de ce dictionnaire** est calculée par la formule [en bits/échantillon] :

$$H(D) = \frac{1}{k}.\sum_{i=1}^{L} P_i.(-\log_2 P_i)$$

avec :

- P_i , la probabilité de sélectionner le vecteur représentant $\mathbf{y}_i$ du dictionnaire D ;
- $(-\log_2 P_i)$, l'appréciation numérique sur l'incertitude de sélectionner $\mathbf{y}_i$, l'unité d'information étant le bit.

L'entropie du dictionnaire précise donc la longueur moyenne des mots du code binaire le plus économique pour transmettre l'information. C'est le débit binaire moyen minimum qui peut-être approché en construisant un code entropique. On remarque que l'entropie est toujours inférieure à la longueur moyenne des index :

$$H(D) \leq \sum_{i=1}^{L} P_i.b_i$$

Les résultats du paragraphe **4.1.2** nous annonce, dans le cas d'une source avec mémoire, un premier intérêt de la quantification vectorielle : la théorie affirme que pour rejoindre le débit entropique il faut regrouper les échantillons de la source, ce qui permet d'en exploiter la mémoire, les redondances.
Intuitivement on comprend que dès des tailles raisonnables de vecteurs, l'effet de mémoire s'amenuise, les blocs deviennent statistiquement indépendants. Alors le calcul du code entropique associé à ces vecteurs est donc simplifié, plus rapide. A priori, même dans le cas d'une source sans mémoire, le calcul d'un code entropique efficace nécessite un regroupement des échantillons.
De plus la propriété d'équirépartition asymtotique implique que, pour une taille très élevée des vecteurs, le code adéquat est à longueur fixe (l'équiprobabilité des vecteurs rend un codage entropique inutile).

4.1.5 Source à amplitude continue et sans mémoire

Soit $\{X(n)\}$ une source stationnaire et sans mémoire, nous considérons également qu'elle est centrée.
Soient $p_x(x)$ sa fonction de densité de probabilité, $\sigma_x^2 = E(X^2(n))$ sa variance et $R_{xx} = \sigma_x^2.\delta(k)$ sa fonction d'autocorrélation ($\delta(k)$ étant le symbole de Kronecker).
Ce signal a une **entropie absolue** infinie, en effet $P(x_k) = 0$ donc $H(X) = +\infty$.
C'est pourquoi on introduit l'**entropie différentielle** (qui peut-être positive, négative ou nulle en fonction de l'amplitude de la source) :

$$h(X) = E(-\log_2 p_x(X)) = -\int_{-\infty}^{+\infty} p_x(x).\log_2 p_x(x)\ dx$$

Il est démontré que l'entropie différentielle est maximale si la source suit une loi Gaussienne $\mathcal{N}(0, \sigma_x^2)$, dans ce cas :

$$p_x(x) = (2.\pi.\sigma_x^2)^{-1}.\exp(\frac{-x^2}{2.\sigma_x^2})$$

Alors :

$$h(X)_G = \log_2 \sqrt{2.\pi.e.\sigma_x^2}$$

Il est aussi pratique de définir la **puissance entropique** qui traduit la répartition de l'information par unité de variance du signal :

$$Q = \frac{1}{2.\pi.e}.2^{2.h(X)}$$

Pour une source gaussienne $Q = \sigma_x^2$, pour une non-gaussienne $Q < \sigma_x^2$.

4.1.6 Source à amplitude continue avec mémoire

Nous considérons cette fois un vecteur $\mathbf{x}$ constitué de N échantillons successifs de la source. $\mathbf{x}$ est décrit par sa fonction de densité de probabilité jointe $p_{\mathbf{X}}(\mathbf{x})$ et on définit l'entropie différentielle par :

$$h(X) = \lim_{N \to \infty} h_N(\mathbf{X})$$

avec :

$$\begin{aligned} h_N(\mathbf{X}) &= \frac{1}{N}.E(-\log_2 p_{\mathbf{X}}(\mathbf{X})) \\ &= -\frac{1}{N}.\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\ldots\int_{-\infty}^{+\infty} p_{\mathbf{X}}(\mathbf{x}).\log_2 p_{\mathbf{X}}(\mathbf{x})\ d\mathbf{x} \end{aligned}$$

De façon générale :

$$(h(X)\ /\ \text{source avec mémoire}) < (h(X)\ /\ \text{source sans mémoire}) \leq \frac{1}{2}.\log_2(2.\pi.e.\sigma_x^2)$$

La borne supérieure est atteinte dans le cas d'une source gaussienne, les entropies inférieures à cette valeur sont dues aux redondances de la source (sa densité de probabilité n'est pas gaussienne et/ou la source présente une mémoire alors son spectre de puissance n'est pas "plat" ou "blanc").

La puissance entropique d'une source gaussienne "colorée" (c.a.d avec mémoire) est :

$$Q = \gamma_x^2.\sigma_x^2$$

où γ_x^2 est la **mesure de platitude du spectre**. Nous avons :

$$0 \leq \gamma_x^2 \leq 1$$

Si $\gamma_x^2 = 1$ on retrouve le cas d'une source gaussienne sans mémoire.
Si $\gamma_x^2 < 1$ la source présente une mémoire, le signal est plus ou moins prédictible.

L'inégalité suivante est alors prouvée :

$$(Q\ /\ \text{source avec mémoire}) < (Q\ /\ \text{source sans mémoire}) \leq \sigma_x^2$$

4.2 Fonction débit-distorsion

4.2.1 Introduction

En pratique la source à coder est le plus souvent à amplitude continue.

On souhaite alors que le système codeur-décodeur assure la transmission de l'information avec un débit R [en bits/échantillon] adapté au canal pour une erreur moyenne de reconstruction D' minimale. En faisant varier R on obtient une courbe $D'(R)$.
La théorie de l'information annonce qu'il existe une **fonction débit-distorsion** $D(R)$ qui fournit une borne

aux performances du système de codage. Cette fonction $D(R)$ indique la distorsion minimale théorique pour le codeur avec le débit R :

$$D(R) \leq D'(R)$$

Dans le cas d'une source à amplitude discrète (pour laquelle une transmission sans erreur est possible), on préfère utiliser la courbe $R'(D)$. Là encore, la théorie fournit la **courbe distorsion-débit** $R(D)$ qui est l'inverse de la fonction $D(R)$ et telle que :

$$R'(D) \geq R(D)$$

4.2.2 Source à amplitude discrète

Un codage entropique est donc réalisable.

Le débit minimum pour transmettre, sans perte, l'information est :

$$\min\{R\} = R(0) = H(X)$$

$R'(0) = h(X)$ est atteint à l'aide d'un code entropique ou à longueur variable. Si la source est avec mémoire la calcul du code est plus aisé.
Nous rappelons qu'un exemple d'une telle source est la sortie d'un quantificateur.
La figure 4 donne un exemple d'une telle courbe, $D(R)$ est donc une fonction monotone décroissante avec R.
La source étant de variance finie, alors $D(0) = \sigma_x^2$ est finie. En effet on peut considérer que, pour $R = 0$, le codeur n'émet que des 0, les erreurs de reconstruction au décodeur sont alors égales aux échantillons des vecteurs source et la variance de l'erreur équivaut à celle de la source.

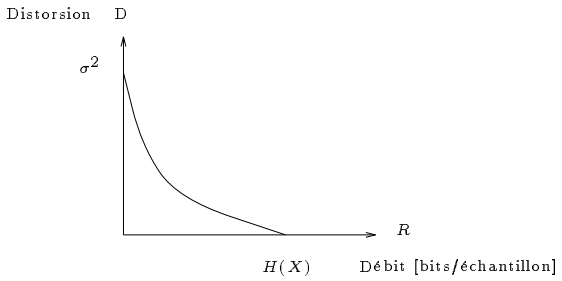

FIG. 4 - *Exemple de la courbe débit-distorsion d'une source à amplitude discrète.*

4.2.3 Source à amplitude continue

4.2.3.1 Distorsion

Nous considérons $\{X(n)\}$, $n = 0, \pm 1, \pm 2, ...$ une source stationnaire, à amplitude continue et sans mémoire. Soit $\mathbf{x}$ un vecteur de N échantillons de la source, $\mathbf{x}$ est une réalisation du vecteur aléatoire $\mathbf{X}$, ce dernier est caractérisé par sa fonction de densité de probabilité jointe $p_{\mathbf{X}}(\mathbf{x})$. Les $\mathbf{y}$ sont eux les vecteurs de reproduction qui arrivent au récepteur. $\mathbf{y}$ est donc une réalisation du vecteur aléatoire $\mathbf{Y}$. Nous parlons de modèle à amplitude continue car les échantillons de ces vecteurs appartiennent à la droite réelle.
En général le vecteur de reproduction $\mathbf{y}$, correspondant au $\mathbf{x}$ émis, en est différent :

$$\mathbf{y} \neq \mathbf{x}$$

Nous choisissons donc d'apprécier la **distorsion moyenne par symbole** à l'aide d'une mesure moyenne d'erreur quadratique :

$$E[d_N(\mathbf{X}, \mathbf{Y})] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} d_N(\mathbf{x}, \mathbf{y}).p(\mathbf{x}, \mathbf{y})\ d\mathbf{x}.d\mathbf{y}$$

avec

$$d_N(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{k=1}^{N} (x(k) - y(k))^2$$

Nous avons $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}/\mathbf{x}).p(\mathbf{x})$. La distorsion moyenne, sujette à une contrainte sur le débit, dépend donc :

- de la statistique de la source $p(\mathbf{x})$,

– de la probabilité des transitions $p(\mathbf{y}/\mathbf{x})$.

Pour la minimiser il faut choisir une modélisation appropriée entre la source et les vecteurs de reconstruction.

4.2.3.2 Information mutuelle

Le calcul de la courbe $D(R)$ repose sur le concept d'information mutuelle.
L'**information mutuelle** [par symbole] $I(X,Y)$ est la mesure capable de décrire le flux d'information entre le codeur et le décodeur, exactement elle apprécie la quantité moyenne d'information qu'implique la "réception" des vecteurs par rapport à ceux émis.

$$I(X,Y) = \lim_{N\to\infty} I_N(\mathbf{X},\mathbf{Y})$$

avec :

$$\begin{aligned} I_N(\mathbf{X},\mathbf{Y}) &= \frac{1}{N}.\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\ldots\int_{-\infty}^{+\infty} p(\mathbf{x},\mathbf{y}).\log_2\left(p(\mathbf{y}/\mathbf{x}).p(\mathbf{x})\right)\ d\mathbf{x}.d\mathbf{y} \\ &= \frac{1}{N}.\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\ldots\int_{-\infty}^{+\infty} p(\mathbf{x},\mathbf{y}).\log_2\left(p(\mathbf{x}/\mathbf{y}).p(\mathbf{y})\right)\ d\mathbf{x}.d\mathbf{y} \end{aligned}$$

$I(X,Y)$ dépend donc également de la statistique de la source et de celle des transitions.
L'adjectif "mutuel" vient de l'égalité : $I(X,Y) = I(Y,X)$.
L'information mutuelle calcule le débit minimal R nécessaire pour avoir une fidélité de reconstruction D. En effet, pour une densité $p(\mathbf{x})$ donnée, considérons :

$$S = \{p(\mathbf{y}/\mathbf{x}) : I_N(\mathbf{X},\mathbf{Y}) \leq R\}$$

S est l'ensemble de tous les schémas de codage ayant une fonction de densité de probabilité de transition pour laquelle l'information mutuelle par symbole est inférieure à un débit donné.
Chacun de ces schémas réalise une erreur moyenne $E[d_N(\mathbf{X},\mathbf{Y})]$. Nous recherchons alors celui assurant le minimum de distorsion :

$$D_N(R) = \min_{p(\mathbf{y}/\mathbf{x})\in S} E[d_N(\mathbf{X},\mathbf{Y})]$$

Le codeur réalisant $D_N(R)$ est optimal, il assure la distorsion moyenne minimale sous une contrainte de débit inférieur à R. On peut remarquer qu'à ce stade $D_N(R)$ est une fonction monotone décroissante avec R.
La théorie montre que ce schéma de codage optimal doit être tel que ses vecteurs source soient statistiquement indépendants. Alors la **fonction débit-distorsion** est définie par :

$$D(R) = \lim_{N\to\infty} D_N(R)$$

> La fonction $D(R)$ fournit, pour à un débit donné R, une borne minimale à la distorsion de tous codeurs. En pratique aucun schéma de codage ne peut atteindre une telle performance.
> Cependant ces équations annoncent que les meilleurs résultats seront obtenus en utilisant des quantificateurs vectoriels. Déjà on remarque que $D_N(R)$ est la distorsion moyenne minimale lorsque les entrées $\mathbf{X}$ et les sorties $\mathbf{Y}$ du système sont des vecteurs. Et il faut savoir que dès des tailles raisonnables, des vecteurs peuvent-être considérés quasiment indépendants.

La courbe inverse de $D(R)$ est $R(D)$. Elle correspond au débit minimal nécessaire pour que le signal reconstruit au récepteur ait une distorsion D.
Dans le cas d'une sourcqe à amplitude continue $R(0) = +\infty$ car, pour $D = 0$:

– $p(\mathbf{y}/\mathbf{x}) = (1 \text{ si } \mathbf{y} = \mathbf{x}\ ;\ 0 \text{ sinon})$,

– $H(X) = H(Y) = I(X,Y) = +\infty$.

Il n'y a donc pas codage. Le cas que l'on étudie est :

– $D > 0$

– $I(X,Y) < +\infty$

Le codage de la source est donc une **opération irréversible** causant une distorsion D pour un débit R.

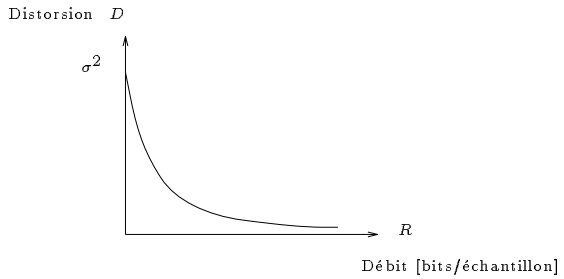

FIG. 5 - *Exemple de la courbe débit-distorsion d'une source à amplitude continue.*

4.2.3.3 Source sans mémoire

4.2.3.3.1 Cas d'une source gaussienne

Nous considérons la source obéissant à la loi normale $\mathcal{N}(0,\ \sigma_x^2)$.
Alors :

$$\begin{aligned} R(D)_G &= \max\left\{0\ ,\ \frac{1}{2}.\log_2\frac{\sigma_x^2}{D}\right\} \\ &= \begin{cases} \frac{1}{2}.\log_2\frac{\sigma_x^2}{D} & \text{si } 0 \leq D \leq \sigma_x^2 \\ 0 & \text{si } D \geq \sigma_x^2 \end{cases} \end{aligned}$$

et

$$D(R)_G = 2^{-2.R}.\sigma_x^2$$

Comme dans le cas de la source à amplitude discrète, il est évident que pour avoir $D = \sigma_x^2$ il n'y a pas d'information à transmettre (On peut considérer que tous les vecteurs de reproduction ont leurs composantes nulles). On peut remarquer que l'erreur quadratique moyenne est réduite d'un facteur quatre pour chaque bit ajouté à la transmission, ou encore que le rapport signal à bruit (SNR) [en DB] est 6.02 fois le débit.

Il est intéressant de modéliser les liens entrée/sortie du système global de codage afin d'en connaître la configuration optimale et d'analyser les effets de la quantification.

Ainsi le système réalisant $R(D)_G$ peut être décomposé en :

$$p(y/x) = \frac{1}{\sqrt{2.\pi.\beta.D}}.\exp\left(\frac{-(y-\beta.x)^2}{2.\beta.D}\right) \quad \text{avec} \quad \beta = 1 - \frac{D}{\sigma_x^2}$$

La sortie obéit à une loi normale $\mathcal{N}(\beta.x,\ \beta.D)$. Les erreurs de quantification suivent donc également une loi gaussienne et elles sont indépendantes vis à vis des variables à l'entrée.
Enfin :

$$R(D)_G = \frac{1}{2}.\left(\log_2\sigma_x^2 - \log_2 D\right)$$

Le débit nécessaire pour reproduire la source avec la distorsion D est la différence en entropie entre la source et le bruit de quantification qui sont deux variables aléatoires normales de variances σ_x^2 et D.

4.2.3.3.2 Cas d'une source non-gaussienne

Il n'existe pas de fonction débit-distorsion explicite mais des bornes de la forme :

$${}^L D(R) \leq D(R) \leq D(R)_G$$

La borne supérieure est la fonction $D(R)_G$ correspondant à la source gaussienne sans mémoire. La borne inférieure correspond à la **borne de shannon** :

$$\begin{aligned} {}^L D(R) &= \frac{1}{2.\pi.e}.2^{-2.(R-h(X))} \\ &= 2^{(-2.R.Q)} \end{aligned}$$

On a aussi :

$$^{L}R(D) = h(X) - \frac{1}{2}.\log_2(2.\pi.e.D)$$

La puissance entropique Q correspond pour une source non-gaussienne à la variance de la gaussienne qui aurait la même entropie différentielle.
Pour une gaussienne $Q = \sigma_x^2$ et on retouve $D(R) = {}^{L}D(R)$.
En pratique, pour une large classe de distributions et à haut débit, ${}^{L}D(R)$ tend vers la fonction débit-distorsion du système $D(R)$. Pour des débits inférieurs (de 1 à 3 bits/échantillon), ${}^{L}D(R)$ est une borne trop optimiste, $D(R)$ est alors calculée numériquement via l'algorithme de blahut [8].

4.2.3.4 Source avec mémoire

La théorie annonce qu'une plus grande compression est possible pour les sources non-gaussiennes avec mémoire. Cependant pour atteindre un tel résultat (traduit par la courbe $D(R)$), le système de codage nécessite plus d'information sur la source que celle uniquement fournit par son spectre de puissance ($S_{xx}(e^{j.w})$) et sa fonction d'autocorrélation.

La présentation théorique qui suit est introduite afin de mieux apréhender les problèmes liés à la quantification (la localisation des erreurs introduites par le codage).

La fonction $D(R)$ d'une source gaussienne colorée (c.a.d avec mémoire, par oposition à la source sans mémoire dont le spectre de puissance constant est dit blanc) est donnée sous une forme paramétrique (le paramètre est ϕ) :

$$\begin{aligned} D(\phi)_G &= \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \min\left\{\phi, S_{xx}(e^{j.w})\right\}\ dw \\ R(\phi)_G &= \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \max\left\{0, \frac{1}{2}.\log_2\frac{S_{xx}(e^{j.w})}{\phi}\right\}\ dw \end{aligned}$$

Ces équations peuvent êqtre interprétées de la façon suivante, l'axe des fréquence est divisé en 2 ensembles A et B :

- $w \in A$ si $S_{xx}(e^{j.w}) \geq \phi \Longleftrightarrow \frac{S_{xx}(e^{j.w})}{\phi} \geq 1$
  donc $R(\phi)_G > 0$, l'information est transmise, la zone est **passe bande** ;
- $w \in B$ si $S_{xx}(e^{j.w}) < \phi \Longleftrightarrow \frac{S_{xx}(e^{j.w})}{\phi} < 1$
  donc $R(\phi)_G = 0$, la contibution spectrale n'est pas transmise, c'est une zone **stoppe bande**.

Si nous considérons $S_{rr}(e^{j.w})$, la puissance spectrale de l'erreur de reconstruction $r(n) = x(n) - y(n)$:

$$\begin{aligned} S_{rr}(e^{j.w}) &= \min\left\{\phi, S_{xx}(e^{j.w})\right\} \\ &= \left\{\phi = \text{cste si } w \in A,\ S_{xx}(e^{j.w}) \text{ si } w \in B\right\} \end{aligned}$$

Alors :

$$\begin{aligned} D(\phi)_G &= \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} S_{rr}(e^{j.w})\ dw \\ R(\phi)_G &= \frac{1}{2.\pi}.\int_{+\pi}^{-\pi} \frac{1}{2}.\log_2\frac{S_{xx}(e^{j.w})}{S_{rr}(e^{j.w})}\ dw \end{aligned}$$

- dans les zones qB, la puissance spectrale de l'erreur est égale à celle de la source (l'information relative à ces zones n'est pas transmise) ;
- dans les zones A, la puissance spectrale de l'erreur est constante, l'information relative à ces zones est transmise et ce que l'on désire alors est un rapport signal sur bruit le plus grand possible.

En pratique, les sources continues ont un spectre de puissance strictement décroissant et on pourrait croire que la quantification est parfaite lorsque son action peut être modélisée par un filtre passe-bas idéal ($H(e^{j.w}) = 1$ pour $w \in A$) suivit de l'addition d'un bruit blanc ayant une puissance spectrale constante (égale à ϕ) dans

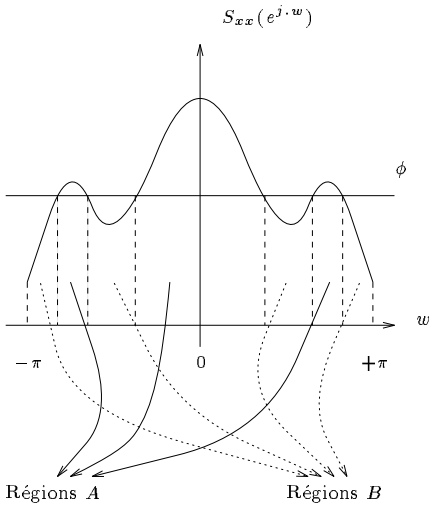

FIG. 6 - *Schéma de principe (on reconnait les zones A et B).*

cette même région.
En fait, le cas idéal où en effet le spectre de l'erreur des zones A est constante et égale à ϕ, est modélisé par la combinaison

- d'un filtre passe-bas non idéal :

$$H(e^{j.w},\ \phi) = \min\left\{0,\ 1 - \frac{\phi}{S_{xx}(e^{j.w})}\right\}$$

- d'un bruit $n(n)$ additif, non blanc, indépendant de la variable à l'entrée et tel que :

$$S_{nn}(e^{j.w}) = \min\left\{0,\ \phi.(1 - \frac{\phi}{S_{xx}(e^{j.w})})\right\}$$

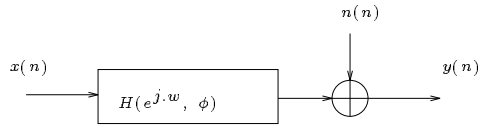

FIG. 7 - *Modélisation de la quantification idéale.*

Les erreurs $r(n) = x(n) - y(n)$ ont donc une puissance :

$$\begin{aligned} S_{rr}(e^{j.w},\ \phi) &= S_{xx}(e^{j.w}).|1 - H(e^{j.w},\ \phi)|^2 + S_{nn}(e^{j.w},\ \phi) \\ &= \frac{\phi^2}{S_{xx}(e^{j.w})} + \phi.(1 - \frac{\phi}{S_{xx}(e^{j.w})}) \quad / \ w \in A \\ &= S^1_{rr}(e^{j.w}) + S^2_{rr}(e^{j.w}) \quad / \ w \in A \end{aligned}$$

Il y a 2 termes :

- S^1_{rr} qui est du au filtre passe bande des zones **A**,
- S^2_{rr} du à la contribution du bruit additif non blanc dans les mêmes zones.

Le troisième terme S^3_{rr} qui apparaît dans la figure 8, est du à la partie stoppe bande des régions B.
Dans la littérature S^1_{rr} et S^2_{rr} sont les **bruits de codage**, S^3_{rr} correspond au **bruit de surcharge**.

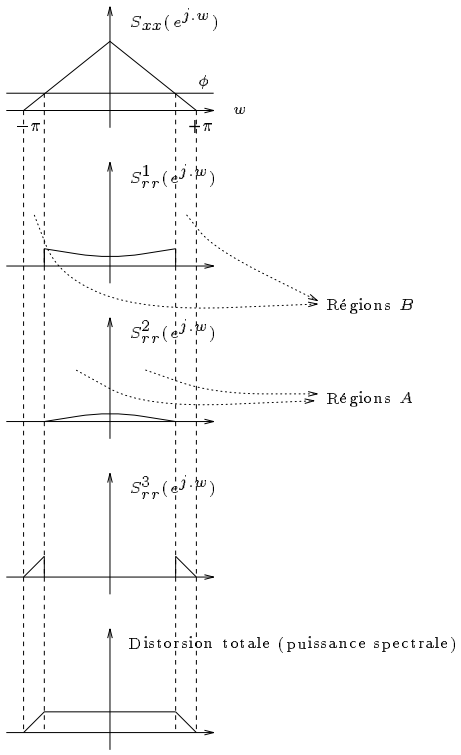

FIG. 8 - *Les bruits de quantification.*

4.2.3.4.1 Petites distorsions

On parle de "petites distorsions" lorsque :

$$\phi \leq \min_{w} \left\{ S_{xx}(e^{j.w}) \right\}$$

Alors une forme simple de la fonction débit-distorsion est obtenu avec :

$$D(R)_G = \gamma_x^2 . 2^{-2.R} . \sigma_x^2$$

avec γ_x^2 la mesure de platitude du spectre S_{xx} :

$$\gamma_x^2 = \frac{\exp\left(\frac{1}{2}. \int_{-\pi}^{+\pi} \log_e S_{xx}(e^{j.w})\ dw\right)}{\sigma_x^2}$$

De plus, pour un débit donné, on obtient :

$$(D(R)_G / \text{ source avec mémoire }) = \gamma_x^2 . (D(R)_G / \text{ source sans mémoire })$$

En exploitant cette mémoire la distorsion peut donc être réduite d'un facteur γ_x^2 dans la zone de "petites distorsions".

4.2.3.4.2 Fonction débit-distorsion en considérant des vecteurs de taille N

Nous avons déjà introduit $D_N(R)$, la fonction débit-distorsion correspondant à une source constituée de blocs de taille N, ces blocs étant indépendants statistiquement entre eux.
Dans le cas d'une telle source gaussienne on a toujours :

$$D(R) = \lim_{N \to +\infty} D_N(R)$$

et il est montré que :

$$D_N(R) \quad = \quad \frac{1}{N} \sum_{k=0}^{N-1} \min\left\{\phi,\ \lambda_k\right\}$$

$$R_N(\phi) \quad = \quad \frac{1}{N}\sum_{k=0}^{N-1} \max\left\{0,\ \frac{1}{2}.\log_2\frac{\lambda_k}{\phi}\right\}$$

λ_k étant la $k^{\text{ième}}$ valeur propre de la matrice d'autocorrélation d'ordre N du processus $\{X(n)\}$.
On remarque que :

- R_N apparaît comme la moyenne de N débits $R_k = \max\left\{0,\ \frac{1}{2}.\log_2\frac{\lambda_k}{\phi}\right\}$
  chaque R_k résulte du codage de sources gaussiennes sans mémoire de variance λ_k ;
- D_N apparaît aussi comme la moyenne de N distorsions optimales $D_k = \min\{\phi,\ \lambda_k\}$.

Toutes les variables aléatoires dont les variances sont supérieures au paramètre ϕ contribuent de la même façon à la distorsion globale du système. On n'a donc pas besoin de transmettre ces variables qui n'apportent aucune information, alors : $R_k = 0$ pour $\phi \geq \lambda_k$

Cas des petites distorsions

C'est la cas si :

$$\phi \leq \min_{k=0,\ 1,\ \ldots,\ N-1}\{\lambda_k\}$$

Alors $D_N = \phi$ et donc :

$$D_N(R) = 2^{-2.R}.(\prod_{k=0}^{N-1}\lambda_k)^{\frac{1}{N}}$$

Finallement on obtient :

$$D_N(R) \leq (D(R)_G/\text{ sans mémoire}) = 2^{-2.R}.\sigma_x^2$$

En effet :

$$\sigma_x^2 = \frac{1}{N}.\sum_{k=0}^{N-1}\lambda_k \geq (\prod_{k=0}^{N-1}\lambda_k)^{\frac{1}{N}}$$

on a égalité sqi et seulement si les λ_k sont tous égaux (la source est alors blanche).

Les sources avec mémoire peuvent donc être transmises avec des distorsions inférieures aux sources sans mémoire.

En utilisant le résultat connu (où $|^N Rxx|$ est le déterminant de la matrice d'autocorrélation) :

$$|^N Rxx| = \prod_{k=0}^{N-1}\lambda_k$$

on peut définir la puissance entropique par :

$$Q_N = |^N Rxx|^{\frac{1}{N}}$$

On obtient alors :

$$D_N(R) = 2^{-2.R}.Q_N = {}^L D(R)$$

La fonction débit-distorsion est donc égale à la borne de Shannon pour les petites distorsions.

Remarque :

Les théorèmes précédents sont aussi vrais si l'on considère des vecteurs successifs qui ne sont pas indépendants, alors il faut prendre N grand.

Source non gaussienne

On retrouve :

$${}^L D(R) \leq D(R) \leq D(R)_G$$

avec ${}^L D(R)$ la borne de Shannon. Là encore, en considérant un second moment fixé et la métrique euclidienne, une source gaussienne est moins compressible qu'une non gaussienne.

5 Supériorité de la quantification vectorielle sur celle scalaire

La théorie de l'information annonce donc d'une façon générale que de meilleures résultats en codage sont obtenus si l'on quantifie des vecteurs plutôt que des scalaires. En contre partie une complexification des systèmes ainsi que des délais de calcul plus importants sont nécessaires. Ce gain de la QV sur la QS est notamment du à l'exploitation de la mémoire de la source (c.a.d des corrélations existant entre les coordonnées des vecteurs). Cependant Zador a théoriquement prouvé la supériorité de la QV sur la QS même si les échantillons de la source sont statistiquement indépendants.

Résultats de Zador

Nous rappelons les notations du **2.2** :

- $\mathbf{x}$ est le vecteur dq'entrée de dimension k, soit $p(\mathbf{x})$ sa fonction de densité de probabilité ;
- le quantificateur choisit parmi L vecteurs de reproduction le $\mathbf{y}_i$ le plus "proche" de $\mathbf{x}$;
- C_i est le voronqoï associé à $\mathbf{y}_i$, l'ensemble des C_i recouvre $I\!R^k$ et ces cellules sont disjointes entre elles ;
- la notion de proximité (soit l'amplitude de l'erreur introduite par la quantification) est mesurée par la distance euclidienne $d(\mathbf{x}, \mathbf{y}_i)$;
- l'**erreur quadratique moyenne** [par dimension] est :

$$\begin{aligned} \mathbf{E} &= \frac{1}{k} \int_{I\!R^k} d(\mathbf{x}, \mathbf{y}_i).p(\mathbf{x})\ d\mathbf{x} \\ &= \frac{1}{k} \sum_{i=1}^{L} \int_{C_i} d(\mathbf{x}, \mathbf{y}_i).p(\mathbf{x})\ d\mathbf{x} \end{aligned}$$

Le problème est : pour k, L, et $p(\mathbf{x})$ donnés nous recherchons le **quantificateur optimal** (c.a.d les $\mathbf{y}_i$) qui minimisent $\mathbf{E}$

$$\mathbf{E}(k, L, p) = \min_{\mathbf{y}_i} \mathbf{E}$$

$\mathbf{E}(k, L, p)$ est l'erreur minimale que l'on peut atteindre.
P.L. Zador a montré qu'il est possible de réduire $\mathbf{E}$ en quantifiant des vecteurs de grandes dimensions [38]. Exactement il a prouvé que :

$$\lim_{L \to +\infty} L^{\frac{2}{k}}.\mathbf{E}(k, L, p) = G_k.(\int_{I\!R^k} p(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}}$$

que l'on peut réécrire (sachant $L = 2^{R.k}$, R étant le débit binaire) :

$$\lim_{L \to +\infty} \mathbf{E}(k, L, p) = 2^{-2.R}.G_k.(\int_{I\!R^k} p(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}}$$

La répartition optimale des vecteurs de reproduction est donc celle proportionnelle à $p(\mathbf{x})^{\frac{k}{k+2}}$
Le paramètre G_k ne dépend que de la dimension k.
Zador a montré que :

$$\frac{1}{(k+2).\pi}.\Gamma(\frac{k}{2}+1)^{\frac{2}{k}} \leq G_k \leq \frac{1}{k.\pi}.\Gamma(\frac{k}{2}+1)^{\frac{2}{k}}.\Gamma(1+\frac{2}{k})$$

donc, si $k \to +\infty$:

$$G_k \to \frac{1}{2.\pi.e} \approx 0.05855$$

Interprétation et calcul de G_k

$$G_k = \frac{\lim_{L \to +\infty} L^{\frac{2}{k}}.\mathbf{E}(k, L, p)}{(\int_{I\!R^k} p(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}}}$$

On considère le cas simple où $\mathbf{x}$ est uniformément distribué sur une grande région de $I\!R^k$ (sur une boule). Alors sur cet espace :

$$p(\mathbf{x}) = \frac{1}{\sum_{i=1}^{L} \int_{c_i} d\mathbf{x}}$$

$\mathbf{E}$ est minimale si chaque $\mathbf{y}_i$ est au centre de son voronoï. En considérant le cas asymptotique (c.a.d L très grand, il n'y a pas de problème de bord) :

$$\mathbf{E} = \frac{1}{k}.\frac{\sum_{i=1}^{L} \int_{c_i} d(\mathbf{x}, \mathbf{y_i}).d\mathbf{x}}{\sum_{i=1}^{L} \int_{c_i} d\mathbf{x}}$$

Nous calculons :

$$\begin{aligned} (\int_{I\!R^k} p(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}} &= (\sum_{i=1}^{L} \int_{C_i} p(\mathbf{x})^{\frac{k}{k+2}}\ d\mathbf{x})^{\frac{k+2}{k}} \\ &= (\sum_{i=1}^{L} \int_{c_i} d\mathbf{x})^{\frac{2}{k}} \end{aligned}$$

Alors :

$$G_k = \lim_{L \to +\infty} \frac{\mathbf{E}(k, L, p)}{(\frac{1}{L}.\sum_{i=1}^{L} \int_{C_i} d\mathbf{x})^{\frac{2}{k}}}$$

G_k s'interprète donc comme le rapport de l'erreur quadratique moyenne [par dimension] en considérant la quantification optimale d'une source uniforme et des conditions asymptotiques, sur un facteur qui rend G_k sans dimension.
G_k est alors tabulé pour une densité uniforme de la source [13] [14].

En notant $\int_{C_i} d\mathbf{x} = \mathrm{vol}(C_i)$:

$$G_k = \frac{1}{k}.\frac{\frac{1}{L}.\sum_{i=1}^{L} \int_{C_i} d(\mathbf{x}, \mathbf{y}_i)\ d\mathbf{x}}{(\frac{1}{L}.\sum_{i=1}^{L} \mathrm{vol}(C_i))^{1+\frac{2}{k}}}$$

La source considérée étant uniforme, on adopte le cas où les vecteurs de reproduction sont les points d'un réseau régulier. Les voronoï sont alors congrues au même polytope Π, alors :

$$\begin{aligned} G_k(\Pi) &= \frac{1}{k}.\frac{\int_{\Pi} d(\mathbf{x}, 0)\ d\mathbf{x}}{\mathrm{vol}(\Pi)^{1+\frac{2}{k}}} \\ &= \frac{1}{k}.\frac{\int_{\Pi} \mathbf{x}^T.\mathbf{x}\ d\mathbf{x}}{\mathrm{vol}(C_0)^{1+\frac{2}{k}}} \end{aligned}$$

On retrouve le moment d'ordre 2 de Π.

Par exemple si $k = 1$, les représentants sont uniformément distribués sur la droite réelle, on construit le réseau tel que les voronoï sont des intervalles de longueur 1, Π est l'intervalle $[-1/2, +1/2]$ (c'est le réseau $\mathbf{Z}$). On obtient :

$$G_1(\Pi) = \frac{\int_{-1/2}^{+1/2} x^2\ dx}{\int_{-1/2}^{+1/2} dx} = \frac{1}{12} \approx 0.08333$$

Ce résultat est classique en quantification : si une variable uniforme est quantifiée scalairement, l'erreur quadratique moyenne est $1/12$ ($> \frac{1}{2.\pi.e}$). Zador a donc montré que cette erreur peut être réduite en utilisant des quantificateurs de dimensions spatiales élevées. Déjà, pour $k = 2$, on trouve en utilisant le réseau régulier hexagonal $G_2(\Pi) = \frac{5}{36.\sqrt{3}} \approx 0.080175$

Propriété d'équirépartition asymptotique

La source envisagée est sans mémoire, ses échantillons $\{x_n\}$ obéissent à la loi marginale $p(x)$. Alors la densité de probabilité conjointe du vecteur $\mathbf{x}$ est donnée par :

$$p(\mathbf{x}) = \prod_{i=1}^{k} p(x_i)$$

Donc :

$$\begin{aligned} -\frac{1}{k}.\log_2 p(\mathbf{x}) &= -\frac{1}{k}.\log_2(\prod_{i=1}^{k} p(x_i)) \\ &= -\frac{1}{k}.\sum_{i=1}^{k}\log_2(p(x_i)) \end{aligned}$$

La loi des grands nombres entraîne la version suivante du théorème de Shannon-McMillan-Breiman [4] :

$$\begin{aligned} \text{si } k \to +\infty : -\frac{1}{k}.\sum_{i=1}^{k}\log_2 p(x_i) \longrightarrow -E(\log_2 p(\mathbf{x})) &= -\int_{-\infty}^{+\infty} p(\mathbf{x}).\log_2 p(\mathbf{x})\ d\mathbf{x} \\ &= h(X) \end{aligned}$$

$h(X)$ est l'entropie différentielle de la source et la convergence a lieu au sens des probabilités, c.a.d :

$$\forall \varepsilon > 0,\ \lim_{k \to +\infty} Pr(|-\frac{1}{k}.\sum_{i=1}^{k}\log_2 p(x_i) - h(X)| > \varepsilon) = 0$$

Ainsi :

$$\text{si } k \to +\infty : -\frac{1}{k}.\log_2 p(\mathbf{x}) \approx h(X) \Longleftrightarrow p(\mathbf{x}) \approx 2^{-k.h(X)}$$

On peut interpréter le résultat : si k est grand, la probabilité est concentrée sur les vecteurs pour lesquels $p(\mathbf{x}) \approx 2^{-k.h(X)}$. *Autrement dit, des vecteurs de grande dimension d'une source sans mémoire ont avec une forte probabilité une densité constante, et ils sont distribués approximativement uniformément dans la région compacte de l'espace où* $p(\mathbf{x})$ *est égale à* $2^{-k.h(X)}$.

6 QV optimale

6.1 Approche intuitive

Comme nous l'avons expliqué, quantifier consiste à répartir dans un espace de dimension fixée un nombre déterminé de représentants, ce nombre étant fonction du débit alloué au quantificateur. L'efficacité du quantificateur se mesure alors à la qualité de restitution du signal source qui doit être la plus fidèle possible (c.a.d avec une erreur de reconstruction minimale).
La QV se révèle supérieure à la QS car elle autorise :

- une utilisation meilleure de la mémoire de la source (exploitation des corrélations existant entre les coordonnées des vecteurs) ;
- une liberté pour la répartition dans l'espace des représentants.

Le quantificateur optimal est donc celui qui répartit les vecteurs de reproduction en tenant compte de la distribution des vecteurs à coder dans l'espace (en tenant compte de leur densité de probabilité multidimensionnelle). En pratique les dimensions d'espace sont limitées aussi un prétraitement de la source est réalisé afin de mieux "concentrer" l'information dans une région compacte de l'espace afin d'obtenir un signal le plus stationnaire possible : ainsi les représentants peuvent être répartis dans cette région (ils ne sont pas dispersés) et le dictionnaire obtenu reste valide au cours du temps. Des exemples de prétraitements sont :

- l'utilisation, classique en codage, de techniques de **transformation** du signal où l'information est décorrélée et donc concentrée (les redondances sont supprimées) ;
- la **classification** des données (on obtient des dictionnaires dédiés).

6.2 Quantificateur optimal - Définitions - Théorème

Pour une distribution statistique donnée de la source et un débit fixé :

- le **quantificateur globalement optimal** est celui qui minimise la distorsion moyenne ;
- un **quantificateur localement optimal** a un dictionnaire qui peut être légèrement perturbé sans que la distorsion moyenne n'augmente.

La théorie qui établit la supériorité de la QV sur la QS n'est pas constructive, elle ne décrit pas la façon de concevoir le dictionnaire globalement optimal. Seuls des propriétés suffisantes sont connues qui permettent de construire des dictionnaires localement optimaux.
Un quantificateur se décompose en 2 applications : un codeur et un décodeur. Le quantificateur (localement) optimal est alors celui réunissant [27] [18] :

- le **codeur optimal** (pour un dictionnaire fixé), celui-ci respecte la "règle du plus proche voisin" que nous avons décrite au **2.2.2.1** :
$$\forall j, \text{ si } d(\mathbf{x}, \mathbf{y}_i) < d(\mathbf{x}, \mathbf{y}_j) \Longrightarrow \mathbf{C}(\mathbf{x}) = \mathbf{y}_i$$
cette règle détermine la partition de l'espace $\mathbb{R}^k$ en voronoï ;
- le **décodeur optimal** (pour une partition de $\mathbb{R}^k$ donnée), le vecteur représentant $\mathbf{y}_i$ doit minimiser la distorsion associée au voronoï C_i, $\mathbf{y}_i$ est donc le centroïde de cette cellule : $\mathbf{y}_i = cent(C_i)$;
- une **troisième condition** est nécessaire : il faut que la probabilité d'avoir un vecteur à coder à la même distance de deux représentants soit nulle, sinon ce vecteur source est affecté à l'un des 2 représentants, la partition optimale de l'espace n'est plus et donc la condition de décodeur optimal est devenue impossible, si les vecteurs source sont à amplitude continue, cette troisième condition est toujours vérifiée.

6.3 Algorithme de Lloyd généralisé

6.3.1 Description

Les 3 conditions précédentes conduisent à la conception d'un algorithme qui réalise, à partir d'une séquence d'apprentissage représentative de la statistique de la source à coder, la construction d'un dictionnaire (localement) optimal.
Cet algorithme de classification, encore appelé **algorithme des K-moyens** ("K means" [29]) est l'extension

au cas vectoriel de l'algorithme de Lloyd-Max du cas scalaire.
Il s'agit d'un algorithme d'optimisation itératif opérant à partir d'un dictionnaire initial. A chaque itération (dite "itération de Lloyd"), 2 opérations distinctes sont appliquées :

- une **classification** suivant la règle de codage optimal,
- une **optimisation** suivant la règle de décodage optimal.

Chaque itération de Lloyd, en modifiant localement le dictionnaire, réduit ou laisse inchanger la distorsion moyenne. L'algorithme converge en un nombre fini d'itérations vers le minimum local le plus proche correspondant au dictionnaire initial. Le choix de ce dernier est donc capital.

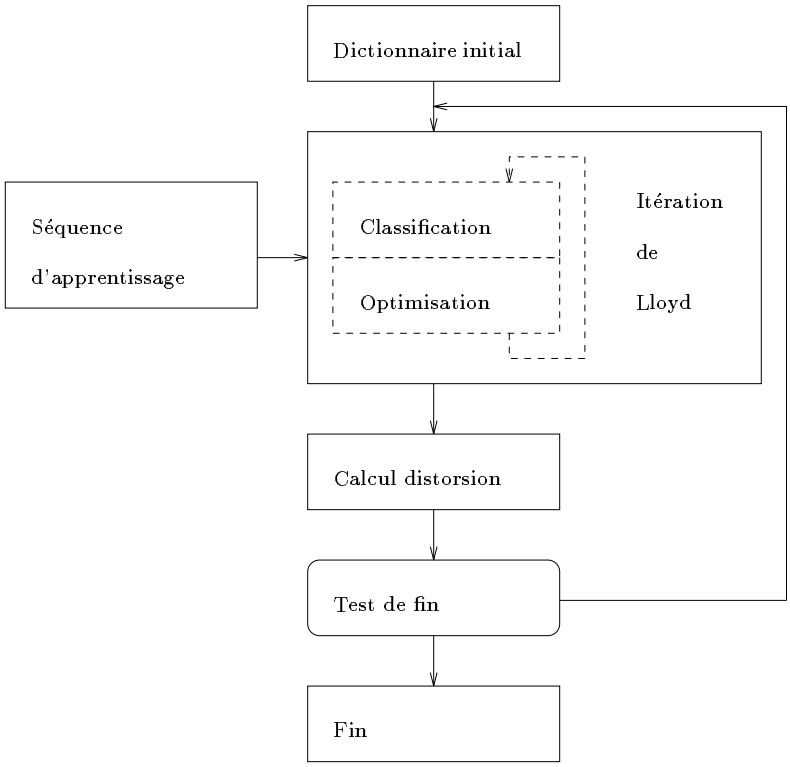

FIG. 9 - *Schéma de fonctionnement de l'algorithme de Lloyd.*

6.3.2 Choix du dictionnaire initial

Le choix de ce dictionnaire initial est essentiel car il conditionne les résultats finaux de l'algorithme. Plusieurs méthodes ont été proposées pour le déterminer.

6.3.2.1 Initialisation aléatoire

Le dictionnaire le plus simple est celui qui contient les L premiers vecteurs de la suite d'apprentissage ou L vecteurs extraits aléatoirement de cette suite. Ces vecteurs peuvent bien sûr ne pas être du tout représentatifs de la suite d'apprentissage et on aboutit à des résultats très médiocres.

6.3.2.2 L'algorithme à seuil

Au lieu de prendre L vecteurs aléatoirement, on fixe une distance minimale entre les éléments du dictionnaire initial. Cette méthode permet d'obtenir une meilleure représentativité que dans le cas précédent mais n'est toujours pas satisfaisante, le seuil étant souvent difficile à déterminer puisque dépendant de la complexité de la séquence d'apprentissage.

6.3.2.3 Méthode des "vecteurs produits"

Cette méthode nécessite de quantifier scalairement les k composantes des vecteurs de la séquence d'apprentissage sur P_k niveaux (avec $P_1.P_2 \dots P_K = L$) et d'effectuer un produit cartésien entre les dictionnaires de base pour obtenir les L représentants initiaux. Le traitement des composantes de manière indépendante ne permet pas d'obtenir à coup sûr un dictionnaire optimal. On peut même, si l'on n'y prend pas garde, obtenir des représentants initiaux ne représentant aucun vecteur de la séquence d'apprentissage.

6.3.2.4 Méthode par dichotomie vectorielle

Cette méthode référencée comme étant l'**algorithme de LBG** [27] combine à l'itération de Lloyd une technique de "splitting". Celle-ci consiste à découper chaque vecteur représentant $\mathbf{y}_i$ en 2 nouveaux vecteurs $\mathbf{y}_i + \varepsilon$ et $\mathbf{y}_i - \varepsilon$ (ε étant un vecteur de perturbation de faible énergie), avant d'appliquer au nouveau dictionnaire obtenu les itérations de Lloyd. Le dictionnaire initial est alors le centroïde de la séquence d'apprentissage, puis l'algorithme génère une succession de dictionnaires (à chaque boucle le nombre de vecteurs de reproduction est multiplié par 2).

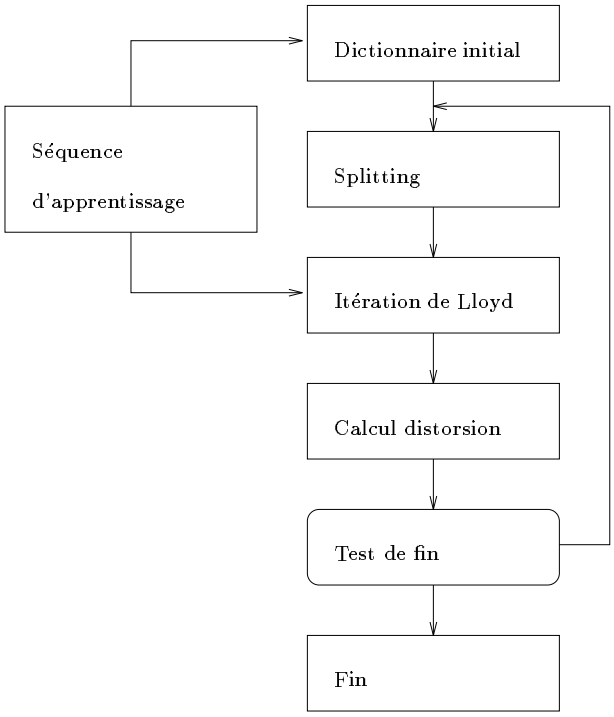

FIG. 10 - *Schéma de fonctionnement de l'algorithme de LBG.*

6.3.2.5 Complexité

6.3.2.5.1 Complexité de la construction du dictionnaire

La complexité de l'algorithme est fonction :

- des paramètres du dictionnaire (la dimension de l'espace k, le nombre de vecteurs représentants L) ;
- des paramètres de la séquence d'apprentissage (le nombre de vecteurs M).

Lors de la construction du dictionnaire, la phase de classification de l'itération de Lloyd est la plus coûteuse en terme calculatoire. En effet chaque vecteurs source doit être comparer à chacun des L représentants, il y a autant de calculs de distorsion dans $\mathbb{R}^k$ et cela à chaque itération. Ainsi si n itérations sont exécutées (en considérant la métrique euclidienne) :

- le nombre de multiplications effectuées est $N_\times = k.L.M.n$,

- le nombre d'additions effectuées est $N_+ = (2.k - 1).L.M.n$,
- le nombre de comparaisons effectuées est $N_c = (L - 1).M.n$.

La phase d'optimisation (calcul des nouveaux centroïdes des classes) nécessite aussi $k.(M - L).n$ additions et $k.L.n$ multiplications.
Enfin $k.(L + M) + M$ emplacements mémoires sont nécessaires pour stocker les vecteurs de la séquence d'apprentissage, leurs classes (auquel chacun appartient) et les vecteurs du dictionnaire.

6.3.2.5.2 Complexité de la recherche au sein du dictionnaire

Le dictionnaire obtenu ne possède aucune structure topologique particulière facilitant le codage. Ainsi pour trouver le vecteur représentant correspondant au vecteur de la source à coder, L calculs de distorsion sont nécessaires. Cette procédure de **recherche exhaustive** au sein du dictionnaire a donc une complexité d'ordre $L = 2^{R.k}$, (R étant le débit binaire). En détaillant, coder un vecteur nécessite :

- $L.k$ multiplications,
- $(2.k - 1).L$ additions,
- $(L - 1)$ comparaisons.

Or de bonnes performances ne sont atteintes qu'à débit élevé (pour de petites dimensions), ou inversement avec de grandes dimensions (pour des débits faibles).

Ce bilan rédhibitoire a conduit à envisager de nouvelles méthodes de quantification vectorielle permettant un codage avec des coûts calculatoires moindres. D'une façon générale ces méthodes consistent à imposer au dictionnaire des **contraintes structurelles** afin de simplifier le codage (rapidité de décision), cependant les résultats obtenus sont toujours sous optimaux (le représentant choisit n'est pas toujours le plus proche voisin du vecteur à coder, cependant il en est proche en moyenne) : il sagit alors de faire un compromis.

7 QV avec contraintes structurelles sur le dictionnaire

Le but de cette partie est de présenter brièvement deux QV classiques de ce type.

7.1 QV arborescent

7.1.1 Définitions

La figure 11 rappelle brièvement la vocabulaire relatif aux arbres. Elle présente l'exemple d'un **arbre B-aire** (ici $B = 2$, on parle d'arbre binaire) c.a.d que d'un noeud père partent B arêtes vers les noeuds fils, cet arbre est planté (il a une racine) et est **équilibré** (ses noeuds terminaux sont tous à la même profondeur).
Exactement, un **arbre** $\mathcal{T}$ est un ensemble de noeuds $\{n_0, n_1, n_2, \ldots\}$, où n_0 est la **racine**. $\tilde{\mathcal{T}}$ est l'ensemble des noeuds terminaux ou **feuilles** de l'arbre. Un **sous-arbre** $\mathcal{S}$ de $\mathcal{T}$ est un arbre dont la racine $n \in \mathcal{T}$, ses feuilles $\tilde{\mathcal{S}}$ n'appartiennent pas forcément à $\tilde{\mathcal{T}}$ (un noeud unique n est aussi considéré comme un sous-arbre). Si de plus $\tilde{\mathcal{S}} \subset \tilde{\mathcal{T}}$, alors $\mathcal{S}$ est une **branche** de $\mathcal{T}$ (notée $\mathcal{S} = \mathcal{T}_n$). Enfin un **sous-arbre élagué** $\mathcal{S}$ est un sous-arbre planté en n_0 (noté $\mathcal{S} \preceq \mathcal{T}$).

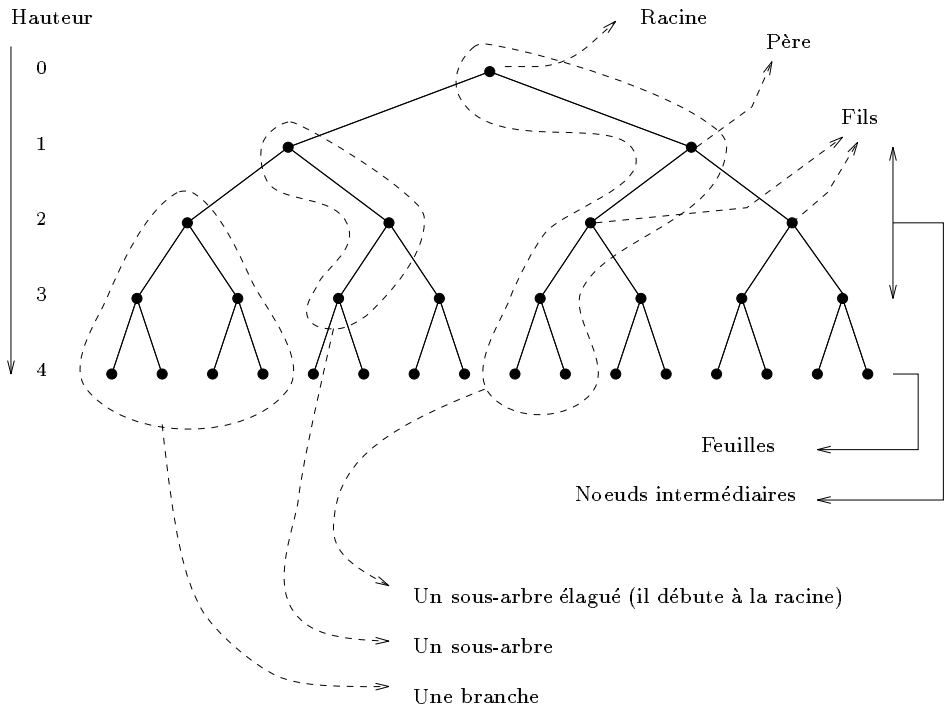

FIG. 11 - *Exemple d'un arbre binaire équilibré.*

Avec la QV arborescente, le codage est effectué à l'aide d'un arbre de décision et les vecteurs représentants sont les L feuilles de cet arbre. Alors, si l'arbre est B-aire et équilibré, sa **profondeur** (ou **hauteur**) est donnée par :

$$H = \log_B L$$

le débit binaire du QV s'exprime par [en bits/dimension] :

$$R = \frac{1}{k}.H.\log_2 B$$

le nombre de noeuds non terminaux est donné par :

$$\sum_{i=0}^{H-1} B^i = \frac{1 - B^H}{1 - B}$$

le nombre total de noeuds est donc :

$$\frac{1 - B^H}{1 - B} + B^H = \frac{1 - B^{H+1}}{1 - B}$$

On distingue 2 grandes méthodes de **construction de dictionnaires arborescents** :

- une **approche descendante** [10] [18] : elle consiste à intégrer la construction de l'arbre à celle du dictionnaire. C'est la méthode la plus utilisée. Par exemple on construit un arbre binaire en mémorisant les dictionnaires successifs obtenus avec l'algorithme LBG ;
- une **approche ascendante** [5] : l'arbre est construit une fois le dictionnaire obtenu. Ce dernier est créé à l'aide d'un algorithme qui apriori ne lui garantit aucune structure particulière (par exemple à l'aide de l'algorithme de Lloyd généralisé). Une classification hiérarchique ascendante est ensuite édifiée, elle consiste à former, à partir de petites classes homogènes, des classes de moins en moins homogènes jusqu'à l'obtention d'une classe unique. Pour cette classification 2 fonctions sont définies : une métrique mesurant la distance entre deux vecteurs et une fonction de discrimination évaluant la dissimilarité entre deux classes de vecteurs.

7.1.2 Principe du codage

Le codeur, pour effectuer sa recherche dispose de l'arborescence et démarre sa recherche à partir de la racine de l'arbre.
A chaque étape, on calcule la distance par rapport à chacun des fils du noeud courant et on sélectionne le noeud apportant une distorsion minimale, la recherche se poursuit ensuite dans le sous-arbre ayant ce noeud comme racine et ce processus est itéré jusqu'à ce que l'on atteigne un noeud terminal de l'arbre. Le code-vecteur associé à ce noeud terminal est alors considéré comme le représentant du vecteur source.
Si l'arbre est B-aire et équilibré, la **complexité de l'algorithme** de recherche est donnée par :

$$B.H = B.\log_B L$$

elle est donc devenue proportionnelle au logarithme de la taille du dictionnaire. La contrainte est qu'il faut stocker le dictionnaire et l'arbre.

7.1.3 Principe du décodage

Il convient de noter que le décodeur ne dispose généralement pas de l'ensemble des noeuds intermédiaires de l'arbre mais seulement des feuilles puisque le codeur lui transmet directement l'indice du code-vecteur qui représente le vecteur source.
Cependant dans le cas particulier de la **reconstruction progressive** [18], le décodeur utilise les noeuds intermédiaires. Au lieu d'attendre que le vecteur source ait été complètement spécifié par le codeur, la transmission se fait à chaque niveau de l'arbre. Au fur et à mesure que le codeur progresse dans sa recherche, le code-vecteur représentatif du vecteur source devient de plus en plus précis et est interprété par le décodeur.

7.1.4 Arbre non équilibré

Un **arbre** est **non équilibré** si les feuilles ne se situent pas toutes sur la même couche. La longueur du mot de code associé à chaque feuille étant proportionnelle à la hauteur de l'arbre, cette structure est donc particulièrement adaptée à la construction d'un **QV à débit variable** où le codeur affecte plus de bits aux régions plus riche en information.
On distingue 2 principales techniques de construction d'arborescences non équilibrées :

- **Algorithme de découpage** [33] : lors de la construction du dictionnaire par une approche descendante, une application non systématique du découpage des noeuds conduit à concevoir un arbre non équilibré ;
- **Algorithme d'élagage** [9] : un arbre équilibré est d'abord construit, cet arbre est ensuite élagué.

Pour chacun des 2 schémas, un citère est utilisé pour déterminer la branche à élaguer ou la feuille à découper. Ce critère, dans un contexte de codage, doit permettre un compromis débit-distorsion.

Nous verrons plus en détail ces techniques dans la suite du rapport.

7.2 QV à l'aide de réseaux réguliers de points (QV algébrique, QV en treillis)

7.2.1 Approche théorique

La propriété d'équirépartition asymptotique (cf le paragraphe **1.2**) établit que, pour une grande dimension d'espace, la probabilité de la source est concentrée sur l'ensembles des vecteurs $\mathbf{x}$ pour lesquels $p(\mathbf{x}) \approx 2^{-2.k.h(X)}$ Cette propriété décrit des conditions optimales pour la QV : les vecteurs à coder sont localisés dans une zone compacte de l'espace, et leur distribution est quasiment uniforme. Alors un dictionnaire, constitué des points d'un réseau régulier (ou **treillis**) ne recouvrant que la région où sont distribués les vecteurs à coder, est parfaitement adapté [17].

Par exemple si la source est sans mémoire et gaussienne alors :

$$p(\mathbf{x}) = \prod_{i=1}^{k} p(x_i) \quad \text{avec} \quad p(x_i) = \frac{1}{\sqrt{2.\pi.\sigma^2}}.\exp -\frac{x_i^2}{2.\sigma^2}$$

et l'entropie différentielle du signal est :

$$h(X) = \log_2 \sqrt{2.\pi.\sigma^2.e}$$

Soit $\mathbf{B}$ la région de l'espace telle que :

$$\mathbf{B} = \left\{ \mathbf{x} : p(\mathbf{x}) = 2^{-2.k.h(X)} \right\}$$

On calcule :

$$\begin{aligned} p(\mathbf{x}) = 2^{-2.k.h(X)} &\iff (\frac{1}{2.\pi.\sigma^2})^{\frac{k}{2}}.e^{-\frac{\sum_{i=1}^{k} x_i^2}{2.\sigma^2}} = 2^{-\frac{k}{2}.\log_2(2.\pi.\sigma^2.e)} \\ &\iff (\frac{1}{2.\pi.\sigma^2})^{\frac{k}{2}}.e^{-\frac{\sum_{i=1}^{k} x_i^2}{2.\sigma^2}} = (\frac{1}{2.\pi.\sigma^2})^{\frac{k}{2}}.e^{-\frac{k}{2}} \\ &\iff \frac{1}{k}.\sum_{i=1}^{k} x_i^2 = \sigma^2 \end{aligned}$$

$\mathbf{B}$ est donc une sphère de rayon σ sur laquelle sont distribués uniformément les vecteurs à coder. Alors le dictionnaire sera les points du treillis appartenant à cette surface (on peut aussi considérer les points du réseau à l'intérieur de $\mathbf{B}$) [16].

Au paragraphe **5.2.1**, nous avons expliqué que le paramètre G_k de l'équation de Zador s'interprète comme l'erreur quadratique moyenne minimale du quantificateur optimal (dans des conditions asymptotiques et pour une source uniforme). De plus il est montré que, si les vecteurs repésentants sont les points d'un treillis, alors G_k est le momment d'ordre 2 de ce réseau. Pour une dimension spatiale fixée, le **treillis optimal** pour quantifier est donc celui dont le moment d'ordre 2 est minimal. Cependant les seuls treillis optimaux connus ont une dimension $k \leq 24$ [14].

7.2.2 Construction d'un QV en treillis

Les étapes de la réalisation pratique d'un QV en treillis peuvent être :

- on utilise le réseau optimal correspondant à la dimension d'espace fixée (la plus grande possible) ;
- une étape de modélisation de la statistique de la source permet de déterminer $\mathbf{B}$ (par exemple si la source est gaussienne $\mathbf{B}$ est donc une sphère, si la source est laplacienne $\mathbf{B}$ est une pyramide [16]) ;
- on tronque alors le treillis en ne conservant que les points sur ou à l'intérieur de $\mathbf{B}$.

Il existe des algorithmes de quantification rapides et simples pour la quantification sur les treillis [12], ces techniques exploitent la haute régularité des réseaux : il n'y a pas de normes à calculer et la complexité est indépendante de la taille du dictionnaire, ce dernier est naturellement connu du codeur et du décodeur. La difficulté majeure réside dans le dénombrement et l'indexage des vecteurs de reproduction, il faut faire appel

à des méthodes complexes [14] [12] [26] [3]. Enfin, le codage entropique des représentants conduit souvent à utiliser une séquence d'apprentissage pour déterminer les index.

Nous présentons plus en détail dans la suite du rapport :

- les réseaux utilisés,
- les algorithmes de quantification à l'aide de ces réseaux.

8 Contexte de l'étude

8.1 Schéma de codage

Notre projet est de concevoir un QV qui doit prendre place au sein d'une chaîne de codage hybride pour la compression de séquence d'images animées. Un exemple d'un tel schéma de codage est donné avec la figure 12.

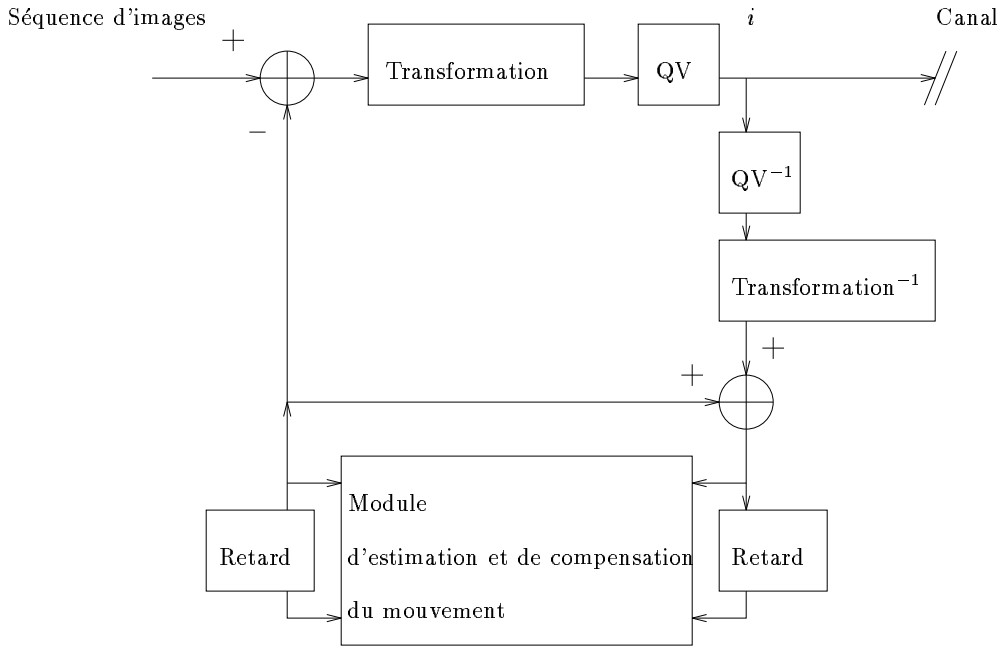

FIG. 12 - *Exemple d'un schéma de codage hybride.*

Les coordonnées des vecteurs à coder (c.a.d les coefficients de la source hybride) correspondent donc à des erreurs de prédiction de compensation de mouvement transformées (coefficients DCT, coefficients d'ondelette , ...). Une modélisation de la distribution statistique de ce type de source est généralement faite à l'aide d'une fonction de la famille des gaussiennes généralisées. Cependant, malgré le prétraitement de la source, le signal à coder n'est jamais tout à fait stationnaire [23] [1] [2].

Remarque :

Une fonction **gaussienne généralisée** est de la forme :

$$p(x) = a.\exp\left(-|b.x|^{\alpha}\right) \text{ avec } \begin{cases} a &= \frac{b.\alpha}{2.\Gamma(1/\alpha)} \\ b &= \frac{1}{\sigma}.\sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}} \end{cases}$$

où σ est l'écart type de la fonction de densité de probabilité à modéliser et Γ la fonction Gamma définie par :

$$\Gamma(n) = \int_0^{+\infty} e^{-x}.x^{n-1}\,dx$$

Dans le cas où $\alpha = 2$, on retrouve la loi normale.
Dans le cas où $\alpha = 1$, on retrouve la loi laplacienne.

8.2 Schéma du QV

Il n'existe pas d'algorithme de construction d'un dictionnaire global et optimal. C'est pourquoi nous préférons retenir une **technique d'apprentissage** pour concevoir notre QV qui doit aussi avoir un caractère **adaptatif** [19] [31] : ainsi une remise à jour des représentants pourra être facilement opéré au cours du temps à partir de séquences d'apprentissage représentatives de la statistique courante de la source. Un test de validité

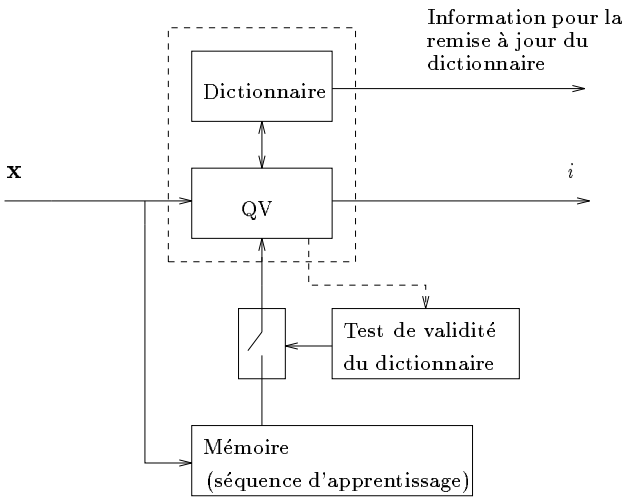

FIG. 13 - *Schéma du QV adaptatif.*

du dictionnaire sera mis en place dans la forme finale de notre quantificateur (ce test peut être la comparaison d'une distorsion moyenne à un seuil).

Ce rapport présente uniquement notre premier travail : la conception même du module de QV de la figure 13.

8.3 La QV par emboîtage d'une hiérarchie de réseaux réguliers de points (QVEHRRP)

Pour concevoir ce QV nous n'avons pas retenu une technique d'apprentissage du type LBG arborescent car l'encodage et surtout la construction du dictionnaire (et son éventuel élagage ou découpage), bien que accélérés, demeurent trop complexes. La QV en treillis est intéressante si la source est stationnaire et si sa statistique autorise une troncature aisée du réseau. Ce n'est pas le cas ici. De plus il ne faut pas oublier qu'en pratique les dimensions d'espace des treillis sont limitées.

La QV par emboîtage d'une hiérarchie de réseaux réguliers de points (QVEHRRP) vise à tirer profil des 2 techniques de codage déjà décrites :

- la quantification simple et rapide sur les treillis [11] ;
- la construction d'un dictionnaire arborescent qui autorise une quantification rapide mais également une partition de l'espace adaptée à la distribution de la source et adaptée à un critère débit-distorsion (arbre non équilibré).

Les **principes** de la QVEHRRP sont :

- la dimension d'espace étant fixée, nous considérons le réseau régulier optimal vis à vis de la quantification mais aussi vis à vis de la rapidité de cette opération : c'est le **réseau support** ;
- ce treillis est tronqué tel qu'il puisse être emboîté , après un changement d'échelle, dans un voronoï du réseau support. Ce **treillis tronqué** est l'élément **de base** de notre QV ;
- nous disposons alors d'un **ensemble hiérarchique** constitué du treillis tronqué de base à différentes échelles ;
- à l'aide de cette hiérarchie de treillis, l'espace est découpé, la partition de l'espace est adaptée à la statistique de la source ;
- le **dictionnaire** obtenu est **arborescent** : les points d'un treillis tronqué de résolution inférieure sont les fils du point associé au voronoï de résolution juste supérieure dans lequel ils sont emboîtés ;
- cet **arbre** peut être **découpé ou élagué** suivant un critère débit-distorsion ;

– la construction du dictionnaire et l'encodage des vecteurs sont **rapides** du fait des techniques utilisées (QV en treillis, dictionnaire arborescent), cette simplicité et cette rapidité du QV autorise le montage d'un schéma de quantification adaptatif.

La partie suivante du rapport décrit la construction d'un QVEHRRP.

9 Construction d'un QVEHRRP

9.1 Les réseaux réguliers de points (treillis)

Les références bibliographiques relatives à cette sous-partie sont [11] [14].

9.1.1 Définitions

Soit un **point** de $\mathbb{R}^k$: $\mathbf{x} = (x_1, x_2, \ldots, x_k)^T$
Une **sphère** de centre $\mathbf{u} = (u_1, u_2, \ldots, u_k)^T$ et de rayon ρ est l'ensemble des points $\mathbf{x}$ tel que :

$$\sum_{i=1}^{k} (x_i - u_i)^2 = \rho^2$$

Dans l'espace de dimension k, nous considérons un empilement régulier de sphères identiques de rayon ρ. Un **réseau régulier** (**treillis** ou "**lattice**") Λ est alors constitué de l'ensemble des centres de ces sphères. Par définition le point 0 est le **centre du réseau**.
On montre alors qu'il existe, en plus de 0, k sphères de centres $\mathbf{v}_i = (v_{i_1}, v_{i_2}, \ldots, v_{i_k})^T$ telles que tout point du treillis peut être déterminé par la somme :

$$\sum_{i=1}^{k} \varepsilon_i . \mathbf{v}_i \quad / \quad \varepsilon_i \in \mathbf{Z}$$

Les $\mathbf{v}_i$ forment la **base du réseau**, ces vecteurs sont linéairement indépendants.
On détermine le **parallélotope fondamental** du treillis Λ comme la région formée par :

$$\sum_{i=1}^{k} \theta_i . \mathbf{v}_i \quad / \quad 0 \leq \theta_i \leq 1$$

L'espace entier peut être recouvert en sommant ces régions (qui elles ne se recouvrent pas). On verra par la suite que l'on sait déterminer le volume du parallélotope, et donc celui d'un voronoï élémentaire du treillis.

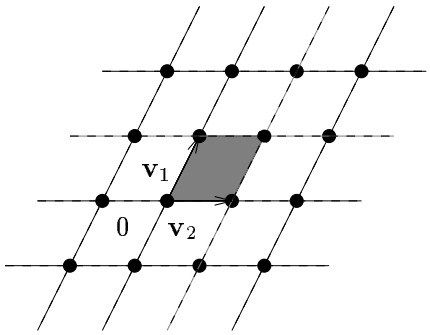

FIG. 14 - *Un treillis bidimensionnel et son parallélotope associé.*

La **matrice génératrice** du treillis est donnée par :

$$M = \begin{pmatrix} v_{1_1}, & v_{1_2}, & \ldots, & v_{1_m} \\ v_{2_1}, & v_{2_2}, & \ldots, & v_{2_m} \\ \vdots & & & \vdots \\ v_{k_1}, & v_{k_2}, & \ldots, & v_{k_m} \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1{}^T \\ \vdots \\ \mathbf{v}_k{}^T \end{pmatrix}$$

En effet, il est parfois plus facile de décrire un réseau k-dimensionnel à l'aide de vecteurs de m coordonnées ($m \geq k$). Cependant, pour la suite du rapport on considère avoir $m = k$.
Alors un vecteur $\mathbf{y}$ du réseau est déterminé par :

$$\mathbf{y} = M^T . \boldsymbol{\varepsilon} \text{ avec } \boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k)^T \quad / \quad \varepsilon_i \in \mathbf{Z}$$

On peut redéfinir un réseau régulier Λ comme l'ensemble des points $\mathbf{y}$ de $I\!R^k$ tel que :

$$\Lambda = \left\{ \mathbf{y} \in I\!R^k \ / \ \exists \boldsymbol{\varepsilon} \in \mathbf{Z}^k \ , \ \mathbf{y} = M^T.\boldsymbol{\varepsilon} = \sum_{i=1}^{k} \varepsilon_i.\mathbf{v}_i \right\}$$

On définit la **matrice de Gram** du réseau par : $A = M.M^T$
Son déterminant est noté (on considère une matrice M carrée) : $\det\Lambda = \det A = (\det M)^2$
Alors le **volume d'un parallélotope** (et donc celui d'un voronoï) est : $\det M = (\det\Lambda)^{1/2}$
Deux réseaux, Λ_1 et Λ_2, peuvent-être **équivalents** ou **similaires** après rotation et/ou réflexion et/ou changement d'échelle. Ceci se note : $\Lambda_1 \cong \Lambda_2$
Un treillis Λ_k admet un **dual** ou **réciproque** ${\Lambda_k}^*$ donné par :

$${\Lambda_k}^* = \left\{ \mathbf{y} \in I\!R^k \ / \ \mathbf{y}^T.\mathbf{u} \in \mathbf{Z} \ , \ \forall \mathbf{u} \in \Lambda_k \right\}$$

Les réseaux réguliers de points sont, à l'origine, des solutions trouvées aux différents problèmes :

- d'empilement de sphères dans un espace,
- de recouvrement de l'espace par des sphères,
- de recherche du "nombre de contacts".

Depuis ces objets mathématiques ont trouvés de nombreux champs d'application (théorie des nombres, télécommunications, chimie, mathématique appliquée ...).

9.1.1.1 Problème d'empilement de sphères dans un espace

On recherche, dans un espace de dimension fixée k, l'empilement le plus dense de sphères (toutes sont équivalentes et elles ne se recouvrent pas). La solution à ce problème n'est connue, pour $k \geq 3$, que si on considère les centres des sphères appartenant à un réseau régulier de points (la densité maximale peut-être obtenue avec des réseaux non réguliers). Dans ce cas on veut maximiser la **densité du réseau** définit par :

$$\begin{aligned} \Delta &= \text{proportion occupée par les sphères} \\ &= \frac{\text{volume d'une sphère}}{\text{volume d'une région fondamentale}} \\ &= \frac{V_k.\rho^k}{(\det\Lambda)^{1/2}} \end{aligned}$$

où :

- $V_k = \frac{\pi^{k/2}}{(k/2)!} = \frac{2^k.\pi^{k-1}}{k!}.(\frac{k-1}{2})!$ est le volume d'une sphère de rayon unité,
- $V_k.\rho^k$ est donc le volume de la sphère k-dimensionnelle de rayon ρ (sa surface est $k.V_k.\rho^{k-1}$).

Le rayon ρ de ces sphères empilées définit le **rayon d'empilement**.

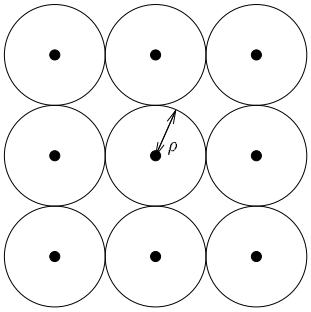

FIG. 15 - *Un empilement de sphères dans l'espace bidimensionnel.*

9.1.1.2 Problème de recouvrement d'espace par des sphères

Cette fois on désire obtenir le recouvrement le plus économique de l'espace euclidien de dimension k avec des sphères. Ces sphères toutes identiques qui recouvrent $\mathbb{R}^k$, se recouvrent également entre elles, et ce dernier recouvrement doit être minimal. Là encore la solution n'est connue, pour $k \geq 3$, que si on considère les centres de ces sphères appartenant à un réseau régulier.
Soit r le rayon de ces sphères. alors on veut minimiser Θ, la **densité du recouvrement** (son "épaisseur") définit par :

$$\begin{aligned} \Theta &= \text{nombre moyen de sphères contenues dans l'espace} \\ &= \frac{\text{volume d'une sphère}}{\text{volume d'une région fondamentale}} \\ &= \frac{V_k . r^k}{(\det\Lambda)^{1/2}} \end{aligned}$$

r définit le **rayon de recouvrement** du réseau, il correspond à la borne supérieure minimale de la distance entre un point $\mathbf{x}$ de l'espace $\mathbb{R}^k$ et le plus proche point du réseau. Si les $\mathbf{y}_i$ sont les points du réseau Λ et si $d(\mathbf{x}, \mathbf{y}_i)$ est la distance entre les 2 vecteurs :

$$r = \sup_{\mathbf{x} \in \mathbb{R}^k} \inf_{\mathbf{y}_i \in \Lambda} d(\mathbf{x}, \mathbf{y}_i)$$

Ces points isolés, à la distance r de leurs plus proches voisins dans le réseau, sont des **trous**.
Le problème de recouvrement apparait le dual de celui de l'empilement, cependant ils sont différents : pour le premier il s'agit de minimiser r, pour le second on veut au contraire maximiser ρ. Alors, pour une dimension donnée, les treillis solution de chacun de ces problèmes sont souvent différents.

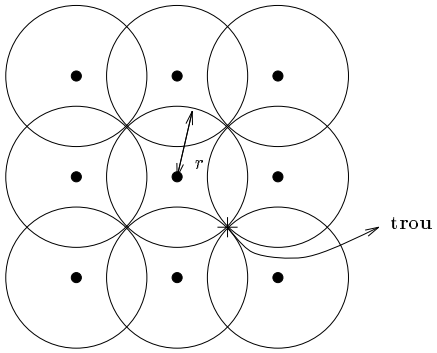

FIG. 16 - *Un recouvrement de l'espace bidimensionnel par des sphères.*

9.1.1.3 Problème du nombre de contacts ("kissing number")

Cette fois on recherche quel est le nombre maximal τ de sphères (toutes identiques et ne se recouvrant pas) qui peuvent être arrangées entre elles de façon à ce que chacune touche la même sphère centrale. Il s'agit donc d'un problème d'empilement mais uniquement sur la surface d'une même sphère, cet aspect local fait que les réseaux solutions sont en général différents de ceux solutions du problème d'empilement de sphères dans l'espace.
Alors que τ est variable si on considère des réseaux non réguliers, il ne l'est plus dans le cas de treillis. τ est donc en général connu que si on considère des points appartenant à un réseau régulier, exactement on ne connait la solution optimale que pour $k = 1,\ 2,\ 3,\ 8$ et 24
La détermination du nombre de contacts intervient naturellement dans la construction des meilleurs **codes sphériques**, un tel code est celui associé à un sous-ensemble de points appartenant à la surface d'une sphère, aussi la recherche de τ permet de mieux répartir ceux-ci.
Un nouvel outil intervient alors : les **séries Thêta** qui indiquent le nombre de points sur la surface d'une sphère.

Séries Thêta

Soit $\mathbf{y}$ un point du treillis Λ : $\mathbf{y} = M^T.\boldsymbol{\varepsilon}$
La norme euclidienne L_2 de $\mathbf{y}$ est :

$$||\mathbf{y}||^2 = \sum_{i=1}^{k} y_i^2 = \mathbf{y}^T.\mathbf{y} = \boldsymbol{\varepsilon}^T.M.M^T.\boldsymbol{\varepsilon} \ = \boldsymbol{\varepsilon}^T.A.\boldsymbol{\varepsilon} \ = f(\boldsymbol{\varepsilon})$$

Cette norme détermine la **forme quadratique** associée au treillis (il parfois plus simple d'utiliser les coordonnées $\boldsymbol{\varepsilon}$ que celles $\mathbf{x}$). Il existe donc une équivalence entre les lattices et leur forme quadratique.
En théorie des nombres un problème se pose : on désire connaitre le nombre de façons d'écrire un entier m comme la somme de k carrés (par exemple $m = \sum_{i=1}^{k} y_i^2$). Les treillis présentent donc, du fait de leur forme quadratique, une formulation à ce genre de problème : considérant le réseau régulier Λ, soit N_m, le **nombre de vecteurs y de Λ de norme** m (c.a.d $\mathbf{y}^T.\mathbf{y} = m$), N_m est encore le nombre de fois que la forme quadratique associée à Λ représente m.
Or le calcul de N_m est facilité par l'introduction des séries Thêta du lattice Λ :

$$\begin{aligned} \Theta_\Lambda(z) &= \sum_{\mathbf{y}\in\Lambda} q^{\mathbf{y}^T.\mathbf{y}} \\ &= \sum_{m=0}^{+\infty} N_m.q^m \text{ où } q = e^{i.\pi.z} \end{aligned}$$

Les séries Thêta produisent donc le nombre de points qu'il y a à chaque distance de l'origine (le nombre de points sur chaque sphère de rayon $\sqrt{m}$, ou encore, sur chaque surface d'énergie m). Les points étant répartis sur des sphères concentriques, pour connaitre le **nombre total N_T de points dans la sphère de rayon $\sqrt{m}$**, il suffit de faire la somme des points sur chacune des sphères de rayon inférieur :

$$N_T = \sum_{i=0}^{m} N_i$$

La plupart des treillis s'exprime, lors du développement de leur série Thêta, comme des fonctions des séries

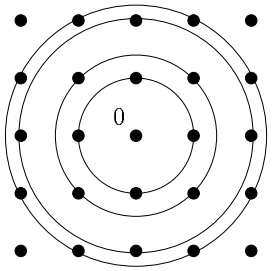

FIG. 17 - *Sphères concentriques d'un réseau bidimensionnel.*

thêta de Jacobi :

$$\begin{aligned} \theta_2(z) = &\ \textstyle\sum_{m=-\infty}^{+\infty} q^{(m+1/2)^2} &&= 2.q^{1/4} + 2.q^{9/4} + 2.q^{25/4} + \ldots \\ \theta_3(z) = &\ \textstyle\sum_{m=-\infty}^{+\infty} q^{m^2} &&= 1 + 2.q + 2.q^4 + 2.q^9 + \ldots \\ \theta_4(z) = &\ \textstyle\sum_{m=-\infty}^{+\infty} (-q)^{m^2} &&= 1 - 2.q + 2.q^4 - 2.q^9 + \ldots \end{aligned}$$

Elles s'expriment donc comme un développement en puissance de q donnant le nombre de vecteurs sur le sphères successives autour de l'origine. Nous rappelons par la suite les séries Thêta des principaux treillis ainsi que leurs valeurs tabulées par Conway et Sloane.

9.1.1.4 Les treillis meilleurs quantificateurs

La **cellule de voronoï** C_i associée au point $\mathbf{y}_i$ du réseau régulier Λ est la région de l'espace constituée des points $\mathbf{x}$ qui sont aussi proches de $\mathbf{y}_i$ que d'un autre point du réseau. On rappelle alors la définition déjà donnée au **2.2.2.1** :

$$C_i = \left\{ \mathbf{x} \in I\!R^k \quad / \quad d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j) \ , \ \forall i \neq j \right\}$$

Les $\mathbf{y}_i$ sont donc séparés par des hyperplans et les régions obtenues sont des polytopes convexes dont l'union forme $I\!R^k$, aux sommets des intersections entre ces hyperplans se situent les **trous** du réseau.
Pour information, nous indiquons que les **sommets d'un voronoï** appartiennent aux hyperplans médiateurs entre les $\mathbf{y}_i$, ce sont les points $\mathbf{x}$ pour lesquels les distances aux $\mathbf{y}_i$ sont des maxima locaux. Aussi il existe une partition de l'espace duale de celle avec les voronoï, il s'agit de la partition en régions convexes de Delaunay : pour chaque sommet, les points $\mathbf{x}$ qui en sont le plus proches forment un polytope, un **cellule de Delaunay**. Le réseau étant régulier, tous les voronoï sont congrus au même polytope Π et ont le même volume égal à celui de la région fondamentale :

$$\mathrm{vol}(\Pi) = \mathrm{vol}(C_i) = \mathrm{vol}(C_0) = \int_{C_0} d\mathbf{x} = (\det\Lambda)^{1/2}$$

Ils ont aussi le même **moment d'ordre 2** (ou moment d'inertie) normalisé :

$$\begin{aligned} G_k(\Pi) &= G_k(C_i) = G_k(C_0) \\ &= \frac{1}{k} . \frac{\int_{C_i} d(\mathbf{x}, \mathbf{y}_i) . d\mathbf{x}}{\mathrm{vol}(C_0)^{1+\frac{2}{k}}} \\ &= \frac{1}{k} . \frac{\int_{C_0} d(\mathbf{x}, 0) . d\mathbf{x}}{\mathrm{vol}(C_0)^{1+\frac{2}{k}}} = \frac{1}{k} . \frac{\int_{C_0} ||\mathbf{x}||^2 . d\mathbf{x}}{\mathrm{vol}(C_0)^{1+\frac{2}{k}}} \end{aligned}$$

On retrouve G_k le coefficient de "Zador" (cf le **5.1.1**).

Le treillis Λ meilleur quantificateur est donc celui pour lequel $G_k(\Pi)$ est minimal.

Là encore le problème est sans solution globale pour $k \geq 2$, et on ne considére que des réseaux réguliers.

Conway et Sloane ont développé pour les treillis $\mathbf{Z}^k$ $(k \geq 1)$, A_k $(k \geq 1)$, D_k $(k \geq 2)$, E_6, E_7, E_8, Λ_{16} et leurs réciproques, des algorithmes de quantification rapide. Pour notre application nous ne retenons, parmi les plus rapides, que $\mathbf{Z}^2$, D_4, E_8 et Λ_{16}.

9.1.2 Les réseaux réguliers importants

9.1.2.1 Résultats généraux

Nous ne rappelons dans le tableau ci-dessous que les noms des treillis solutions des différents problèmes évoqués et en nous limitant à des dimensions $k \leq 24$.

k	1	2	3	4	5	6	7	8	12	16	24
Meilleur empilement	$\mathbf{Z}$	A_2	A_3	D_4	D_5	E_6	E_7	E_8	K_{12}	Λ_{16}	Λ_{24}
Plus grand nombre de contacts	$\mathbf{Z}$ 2	A_2 6	A_3 12	D_4 24	D_5 40	E_6 72	E_7 126	E_8 240	P_{12a} 840	Λ_{16} 4320	Λ_{24} 196560
Meilleur recouvrement	$\mathbf{Z}$	A_2	A_3^*	A_4^*	A_5^*	A_6^*	A_7^*	A_8^*	A_{12}^*	A_{16}^*	Λ_{24}
Meilleur quantificateur	$\mathbf{Z}$	A_2	A_3^*	D_4	D_5^*	E_6^*	E_7^*	E_8	K_{12}	Λ_{16}	Λ_{24}

Nous donnons une liste d'explications et de remarques :

- les treillis encadrés offrent la solution optimale globale (parmi les réseaux réguliers ou non) ;
- les treillis à gauche du trait vertical sont optimaux parmi les réseaux réguliers ;
- les $\mathbf{Z}^k$ ($k \geq 1$) sont les réseaux cubiques ;
- les A_k ($k \geq 1$) sont les réseaux "racine" (root lattices);
- la plupart de ces treillis appartiennent à la famille des réseaux par "strates" noté Λ_k et il existe de nombreuses équivalences ($\Lambda_1 \cong \mathbf{Z} \cong A_1 \cong A_1^* \cong D_1,\ \Lambda_2 \cong A_2 \cong A_2^*,\ \Lambda_3 \cong A_3 \cong D_3,\ \Lambda_4 \cong D_4 \cong D_4^*,\ \Lambda_5 \cong D_5,\ \Lambda_6 \cong E_6,\ \Lambda_7 \cong E_7,\ \Lambda_8 \cong E_8 \cong E_8^*,\ \ldots$) ;
- A_2 est le réseau de hexagonal ;
- A_3 est le réseau cubique à face centrée ;
- A_3^* est le réseau cubique à corps centré ;
- K_{12} est le réseau de Coxeter-Todd ;
- Λ_{16} est le réseau de Barnes-Wall ;
- Λ_{24} est le réseau de Leech.

9.1.2.2 Les treillis utilisés

Nous ne présentons donc en détail que les réseaux réguliers que nous avons retenus pour notre application.

9.1.2.2.1 Le réseau cubique $\mathbf{Z}^k$

$$\mathbf{Z}^k = \{\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ \ / \ \ y_i \in \mathbf{Z}\}$$

C'est l'ensemble des points de $I\!R^k$ dont les coordonnées sont des entiers. Nous avons :

- $M = \mathrm{Id}$ (où Id signifie la matrice Identité),
- $\det A = 1$,
- $\tau = 2.k$,
- $\rho = 1/2$,
- $r = \frac{\sqrt{k}}{2} = \rho.\sqrt{k}$,
- les voronoï sont des cubes,
- $\Theta_{\mathbf{Z}^k}(z) = (\theta_3(z))^k$ donc :

$$\begin{aligned}
\Theta_{\mathbf{Z}}(z) &= 1 + 2.q + 2.q^4 + 2.q^9 + \ldots \\
\Theta_{\mathbf{Z}^2}(z) &= 1 + 4.q + 4.q^2 + \ldots \\
\Theta_{\mathbf{Z}^3}(z) &= 1 + 6.q + 12.q^2 + \ldots \\
\Theta_{\mathbf{Z}^4}(z) &= 1 + 8.q + 24.q^2 + \ldots
\end{aligned}$$

Le tableau ci-dessous rappelle le nombre de points pour les premières sphères de $\mathbf{Z}$, $\mathbf{Z}^2$, $\mathbf{Z}^3$, $\mathbf{Z}^4$:

Energie m de la surface de la sphère	Nombre de points sur les sphères de $\mathbf{Z}$	Nombre de points sur les sphères de $\mathbf{Z}^2$	Nombre de points sur les sphères de $\mathbf{Z}^3$	Nombre de points sur les sphères de $\mathbf{Z}^4$
0	1	1	1	1
1	2	4	6	8
2	0	4	12	24
3	0	0	8	32
4	2	4	6	24
5	0	8	24	48
6	0	0	24	96
7	0	0	0	64
8	0	4	12	24
9	2	4	30	104
10	0	8	24	144
11	0	0	24	96
12	0	0	8	96
13	0	8	24	112
14	0	0	48	192
15	0	0	0	192
16	2	4	6	24
17	0	8	48	144
18	0	4	36	312
19	0	0	24	160
20	0	8	24	144

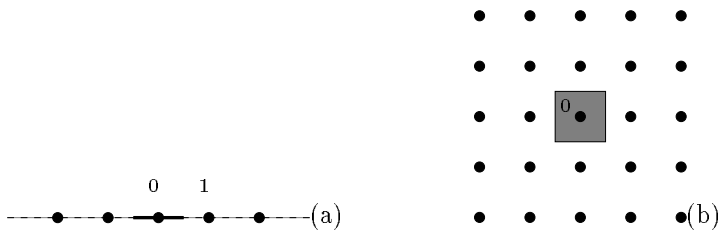

FIG. 18 - *Les réseaux* $\mathbf{Z}$ *(a) et* $\mathbf{Z}^2$ *(b).*

Quantification rapide dans un réseau cubique $\mathbf{Z}^k$

Soit $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_k)^T \in I\!R^k$ le vecteur à quantifier, on désire lui associer le vecteur de reproduction $\mathbf{y} \in \mathbf{Z}^k$ le plus proche.
Soit f la fonction qui, appliquée au réel x_i, nous rend l'entier le plus proche. Dans le cas où x_i est équidistant de 2 entiers, f nous rend l'entier ayant la valeur absolue la plus petite (c.a.d si $x_i = l_i + 0.5$ / $l_i \in \mathbf{Z}$, alors, $f(x_i) = l_i$), on choisit ainsi le vecteurs représentant ayant la plus petite énergie. On obtient :

$$\begin{aligned} \mathbf{y} &= f(\mathbf{x}) \\ &= (f(x_1),\ f(x_2),\ \ldots,\ f(x_k))^T \end{aligned}$$

On effectue donc une quantification scalaire uniforme de pas unité sur chaque coordonnée du vecteur source.

9.1.2.2.2 Le réseau D_k $(k \geq 2)$

$$D_k = \left\{ \mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_k)^T \ / \ \mathbf{y} \in \mathbf{Z}^k,\ \sum_{i=1}^{k} y_i = 0 \bmod(2) \right\}$$

D_k est donc l'ensemble des points de $\mathbf{Z}^k$ dont la somme des coordonnées est paire. Nous avons :

–

$$M = \begin{pmatrix} -1, & -1, & 0, & \ldots, & 0, & 0 \\ -1, & 1, & 0, & \ldots, & 0, & 0 \\ 0, & -1, & -1, & \ldots, & 0, & 0 \\ \vdots & & & & & \vdots \\ 0, & 0, & 0, & \ldots, & 1, & -1 \end{pmatrix}$$

– $\det A = 4$,

– $\tau = 2.k.(k-1)$,

– $\rho = 1/\sqrt{2}$,

– $r = \rho.\sqrt{2}$ $(k = 2,\ 3)$ ou $r = \rho.\sqrt{k/2}$ $(k \geq 4)$,

– $\Theta_{D_k}(z) = \frac{1}{2}.\left(\theta_3(z)^k + \theta_4(z)^k\right)$

Ce réseau est important car il sert aussi à construire les treillis E_8 et Λ_{16}.
Pour $k = 1,\ 2$: $D_k \cong \mathbf{Z}^k$ (ils sont équivalents à un facteur de dilatation et une rotation près).
Pour D_4 :

$$\Theta_{D_4}(z) = 1 + 24.q^2 + 24.q^4 + \ldots$$

Energie m de la surface de la sphère	Nombre de points sur les sphères de D_4
0	1
2	24
4	24
6	96
8	24
10	144
12	96
14	192
16	24
18	312
20	144

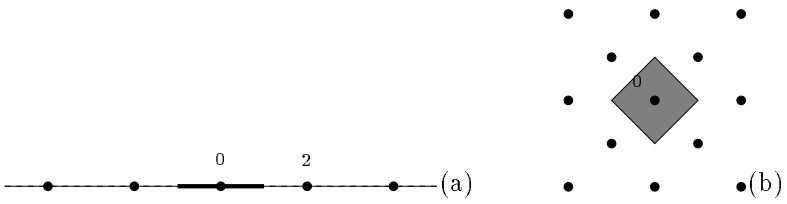

FIG. 19 - *Les réseaux D_1 (a) et D_2 (b).*

Quantification rapide dans un réseau D_k

Soit $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_k)^T \in I\!R^k$ le vecteur de la source à quantifier, on désire cette fois lui associer le vecteur de reproduction $\mathbf{y} \in D^k$ le plus proche.
Nous avons déjà déterminé la fonction f qui associe au réel x_i l'entier le plus proche $f(x_i)$.
Aussi $\delta(x_i) = x_i - f(x_i)$ correspond à l'erreur de quantification faite ($|\delta(x_i)| \leq 1/2$).
Soit w la fonction qui, appliquée à x_i, nous rend le second entier le plus proche ("wrong way") :

$$w(x_i) = f(x_i) + \text{sign}(\delta(x_i)) \quad \text{avec sign}(z) = \begin{cases} 1 & \text{si } z \geq 0 \\ -1 & \text{si } z < 0 \end{cases}$$

Alors, étant donné $\mathbf{x}$, soit l'entier n $(1 \leq n \leq k)$ tel que :

- $|\delta(x_n)| \geq |\delta(x_i)| \quad , \forall 1 \leq i \leq k$
- si $|\delta(x_n)| = |\delta(x_i)| \implies n \leq i$

x_n est donc la composante pour laquelle l'erreur de quantification est la plus grande.
Soit la fonction g définit par :

$$g(\mathbf{x}) = (f(x_1),\ f(x_2),\ \ldots,\ w(x_n),\ \ldots,\ f(x_k))^T$$

Par rapport à $f(\mathbf{x})$, on a remplacé $f(x_n)$ par $w(x_n)$. Les deux vecteurs $f(\mathbf{x})$ et $g(\mathbf{x})$ diffèrent donc par une seule composante, et la somme de leurs coordonnées diffère d'une unité. Le point appartenant à D_k est alors celui dont la somme des composantes est paire.
La procédure à suivre pour trouver le point $\mathbf{y} \in D_k$ le plus proche de $\mathbf{x}$ est alors :

- (1) calcul de $f(\mathbf{x})$, si la somme de ses coordonnées est paire alors $\mathbf{y} = f(\mathbf{x})$,
- (2) sinon, calcul de $g(\mathbf{x})$, alors $\mathbf{y} = g(\mathbf{x})$.

Dans le cas (1), il faut effectuer $2.k$ opérations : k arrondis (les $f(x_i)$), $k-1$ sommes $(\sum_{i=1}^{k} f(x_i))$, 1 test de parité.
Dans le cas $((1) + (2))$, $3.k + 2$ opérations sont effectuées en tout : il faut calculer en plus k différences (les $\delta(x_i)$), une recherche de maximum, un arrondi $(w(x_i))$.
La complexité de cet algorithme de quantification est donc de l'ordre de k. Contrairement aux algorithmes du type LBG, il n'y a pas de norme à calculer et le calcul est indépendant de la taille du dictionnaire.

Exemples (avec le réseau D_4) :

- (a)

$$\begin{aligned} \mathbf{x} &= (0.6,\ -1.1,\ 1.7,\ 0.1)^T \\ f(\mathbf{x}) &= (1,\ -1,\ 2,\ 0)^T \\ \delta(\mathbf{x}) &= (-0.4,\ -0.1,\ -0.3,\ 0.1)^T \\ g(\mathbf{x}) &= (0,\ -1,\ 2,\ 0)^T \end{aligned}$$

  La somme des coordonnées de $f(\mathbf{x})$ est paire, celle des coordonnées de $g(\mathbf{x})$ est impaire, donc $\mathbf{y} = f(\mathbf{x})$ est le point de D_4 le plus proche de $\mathbf{x}$.

- (b)

$$\begin{aligned} \mathbf{x} &= (0.5,\ 0.5,\ 0.5,\ 0.5)^T \\ f(\mathbf{x}) &= (0,\ 0,\ 0,\ 0)^T \\ g(\mathbf{x}) &= (1,\ 0,\ 0,\ 0)^T \end{aligned}$$

  donc $\mathbf{y} = f(\mathbf{x})$

9.1.2.2.3 Le réseau E_8

Le réseau E_8, encore appelé réseau en "diamant", est défini par la relation :

$$E_8 = D_8 \cup \left[[\frac{\mathbf{1}}{\mathbf{2}}] + D_8\right] \text{ où } [\frac{\mathbf{1}}{\mathbf{2}}] = \left(\frac{1}{2},\ \frac{1}{2},\ \ldots,\ \frac{1}{2}\right)$$

Il correspond donc à l'union du réseau D_8 avec le **réseau décalé** ('coset') $[\frac{\mathbf{1}}{\mathbf{2}}] + D_8$, soit encore :

$$\begin{aligned} E_8 = \Big\{ & \mathbf{y}' = (y_1',\ y_2',\ \ldots,\ y_k')^T,\ \ \mathbf{y}'' = (y_1'',\ y_2'',\ \ldots,\ y_k'')^T \ \ / \\ & y_i' \in \mathbf{Z} \text{ et } \textstyle\sum_{i=1}^{8} y_i' = 0 \bmod(2),\ y_i'' \in (\mathbf{Z} + \frac{1}{2}) \text{ et } \sum_{i=1}^{8} y_i'' = 0 \bmod(2) \Big\} \end{aligned}$$

Nous avons :

- $\det A = 1$,

- $\tau = 240$,
- $\rho = 1/\sqrt{2}$,
- $r = \rho.\sqrt{2} = 1$,
- $\Theta_{E_8}(z) = \frac{1}{2}.\left(\theta_2(z)^8 + \theta_3(z)^8 + \theta_4(z)^8\right) = 1 + 240.q^2 + 2160.q^4 + \ldots$

Energie m de la surface de la sphère	Nombre de points sur les sphères de E_8
0	1
2	240
4	2160
6	6720
8	17520
10	30240
12	60480
14	82560
16	140400
18	181680
20	272160

Quantification rapide dans un réseau décalé $\mathbf{r} + \Lambda$

Une procédure Φ permettant de trouver le point le plus proche d'un vecteur donné $\mathbf{x}$ danxs le réseau Λ peut être étendue afin de déterminer le point le plus proche dans le réseau décalé $\mathbf{r} + \Lambda$.
Ainsi si $\Phi(\mathbf{x})$ est le point le plus proche de $\mathbf{x}$ dans Λ, $\Phi(\mathbf{x} - \mathbf{r}) + \mathbf{r}$ est le point le plus proche dans $\Lambda + \mathbf{r}$.
On généralise en considérant une **union de réseaux décalés** $\mathcal{L} = \bigcup_{i=0}^{l-1} (\mathbf{r}_i + \Lambda)$.
$\mathbf{x}$ étant le vecteur à quantifier, la méthode devient :

- (1) calcul pour chaque $i = 0,\ 1,\ \ldots,\ l-1$ de $\mathbf{y}_i = \Phi(\mathbf{x} + \mathbf{r}_i)$;
- (2) comparaison de chacun des $\mathbf{y}_0,\ \mathbf{y}_1,\ \ldots,\ \mathbf{y}_{l-1}$ avec $\mathbf{x}$ et choix du plus proche au sens de la norme euclidienne L_2 (l calculs de distorsion sont donc nécessaires).

Quantification rapide dans un réseau E_8

Le réseau E_8 est l'union des réseaux D_8 et D_8 décalé de $[\frac{\mathbf{1}}{\mathbf{2}}]$, la procédure de quantification d'un vecteur donné $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_8)^T \in I\!R^8$ est alors :

- (1) calcul des vecteurs $f(\mathbf{x})$ et $g(\mathbf{x})$, puis sélection de celui dont la somme des coordonnées est paire, soit $\mathbf{y}_0$ ce vecteur ;
- (2) calcul des vecteurs $f(\mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}])$ et $g(\mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}])$, puis sélection de celui ayant une somme de coordonnées paire, ce vecteur auquel on ajoute $[\frac{\mathbf{1}}{\mathbf{2}}]$ est $\mathbf{y}_1$;
- (3) calcul des normes $d(\mathbf{y}_0, \mathbf{x})$ et $d(\mathbf{y}_1, \mathbf{x})$, le vecteur $\mathbf{y}_i$ le plus proche de $\mathbf{x}$ est retenu.

La complexité de quantification est donc la somme des complexités suivantes :

- complexité de la quantification dans D_8 ;
- complexité de la quantification dans D_8 décalé (c.a.d le décalage, la quantification dans D_8, le décalage inverse) ;
- recherche du minimum des 2 normes.

Cette complexité, indépendante de la taille du dictionnaire, demeure nettement moindres de celle des algorithmes du type LBG.

Exemples:

$$\begin{aligned} \mathbf{x} &= (0.1,\ 0.1,\ 0.8,\ 1.3,\ 2.2,\ -0.6,\ -0.7,\ 0.9)^T \\ f(\mathbf{x}) &= (0,\ 0,\ 1,\ 1,\ 2,\ -1,\ -1,\ 1)^T \\ g(\mathbf{x}) &= (0,\ 0,\ 1,\ 1,\ 2,\ 0,\ -1,\ 1)^T \end{aligned}$$

donc $\mathbf{y}_0 = g(\mathbf{x})$

$$\begin{aligned} \mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}] &= (-0.4,\ -0.4,\ 0.3,\ 0.8,\ 1.7,\ -1.1,\ -1.2,\ 0.4)^T \\ f(\mathbf{x}) - [\frac{\mathbf{1}}{\mathbf{2}}] &= (0,\ 0,\ 0,\ 1,\ 2,\ -1,\ -1,\ 0)^T \\ g(\mathbf{x}) - [\frac{\mathbf{1}}{\mathbf{2}}] &= (-1,\ 0,\ 0,\ 1,\ 2,\ -1,\ -1,\ 0)^T \end{aligned}$$

donc $\mathbf{y}_1 = g(\mathbf{x} - [\frac{\mathbf{1}}{\mathbf{2}}]) + [\frac{\mathbf{1}}{\mathbf{2}}] = (-0.5,\ 0.5,\ 0.5,\ 1.5,\ 2.5,\ -0.5,\ -0.5,\ 0.5)^T$
$d(\mathbf{y}_0, \mathbf{x}) = 0.65$ et $d(\mathbf{y}_1, \mathbf{x}) = 0.95$
Alors $\mathbf{y}_0$ est le point de D_8 le plus proche de $\mathbf{x}$.

9.1.2.2.4 Le réseau de Barnes-Wall Λ_{16}

Nous avons :

- $\det A = 256$,
- $\tau = 4320$,
- $\rho = 1$,
- $r = \rho.\sqrt{3}$,
-

$$\begin{aligned} \Theta_{\Lambda_{16}}(z) &= \frac{1}{2}.\left(\theta_2(2.z)^{16} + \theta_3(2.z)^{16} + \theta_4(2.z)^{16} + 30.\theta_2(2.z)^8.\theta_3(2.z)^{16}\right) \\ &= 1 + 4320.q^4 + 61440.q^6 + \ldots \end{aligned}$$

Energie m de la surface de la sphère	Nombre de points sur les sphères de E_8
0	1
2	0
4	4320
6	61440
8	522720
10	2211840
12	8960640
14	23224320
16	67154400
18	135168000
20	319809600

Ce réseau est défini par :

$$\Lambda_{16} = \bigcup_{i=0}^{31} (\mathbf{r}_i + 2.D_{16})$$

où les vecteurs de translation $\mathbf{r}_i$ correspondent aux lignes (ou colonnes) d'une matrice de Hadamard de type Sylvester $\tilde{H}_{16}$ dans laquelle on a effectué le changement $(-1,\ 1) \rightarrow (1,\ 0)$, et aux lignes (ou colonnes) de la matrice complémentée $\bar{\tilde{H}}_{16}$.
Nous rappelons que :

$$\tilde{H}_2 = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ et } \tilde{H}_{2N} = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} \tilde{H}_N & \tilde{H}_N \\ \tilde{H}_N & -\tilde{H}_N \end{pmatrix}$$

Quantification rapide dans un réseau Λ_{16}

Ce réseau se présente comme l'union de 32 réseaux D_{16} décalés, nous pouvons donc procéder ainsi :

- recherche du plus proche voisin dans chacun des réseaux décalés,
- choix parmi les 32 représentants possibles de celui le plus proche de $\mathbf{x}$ selon la norme L_2.

9.2 Emboîtage de réseaux réguliers de points

9.2.1 Troncature de réseaux

Nous savons déterminé le réseau régulier support correspondant à la dimension de l'espace vectoriel choisi [14]. La seconde étape de la construction du QV est alors la **troncature** de ce treillis. En effet, nous ne pouvons conserver qu'un sous-ensemble du nombre infini de points appartenant à la structure régulière.
De manière classique on ne conserve que les vecteurs représentants à l'intérieur ou sur la boule de rayon $\sqrt{\mathcal{E}_T}$, où $\mathcal{E}_T$ définit l'**énergie de troncature** du réseau [16]. Cette boule est une sphère si on utilise comme dans notre cas la norme L_2, c'est une pyramide si on utilise la norme L_1. Nous avons décrit au **7.2** ces méthodes classiques qui consistent à directement tronquer le treillis en fonction de la distribution modélisée des vecteurs source. Appliquer ces méthodes dans notre cas ne serait pas judicieux car la statistique de la source n'est ni stationnaire ni simplement modélisable à l'aide d'une loi normale ou laplacienne. Notre démarche est donc différente.

9.2.2 Quantification par projection dans un réseau tronqué

Tout d'abord nous retenons la technique de quantification qui consiste à projeter, une fois le treillis tronqué, les vecteurs à coder à l'intérieur de la sphère de rayon $\sqrt{\mathcal{E}_T}$.
Alors, si $\mathcal{S} = \left\{\mathbf{x}_j = (x_1, \ldots, x_k)^T \;\; / \;\; j = 0, 1, 2, \ldots\right\}$ est la source vectorielle,

- il faut premièrement apprécier l'**énergie maximale** possible d'un vecteur à coder :

$$\mathcal{E}_{max} = \max_{\mathcal{E}} \{\mathcal{E}(\mathbf{x}) \;\; / \;\; \mathbf{x} \in \mathcal{S}\}$$

  avec

$$\mathcal{E}(\mathbf{x}) = L_2(\mathbf{x}) = ||\mathbf{x}||^2 = \sum_{i=1}^{k} |x_i|^2$$

- on projette alors un vecteur $\mathbf{x}$, dans ou sur la sphère de rayon $\sqrt{\mathcal{E}_T}$, en multipliant chacune de ses coordonnées x_i par le **facteur de projection** :

$$F = \sqrt{\frac{\mathcal{E}_T}{\mathcal{E}_{max}}}$$

  Tous les vecteurs de la source seront à l'intérieur ou sur cette sphère car :

$$\sum_{i=1}^{k} (F.x_i)^2 = F^2.\sum_{i=1}^{k} x_i^2 = \frac{\mathcal{E}_T}{\mathcal{E}_{max}}.\mathcal{E} \leq \mathcal{E}_T$$

- le vecteur projeté est aussitôt quantifié en utilisant un algorithme de quantification rapide [11].

Il faut remarquer qu'avec cette méthode le sous-ensemble des points du treillis conservés est alors constitué de ceux dont le voronoï est entièrement ou partiellement à l'intérieur de la sphère d'énergie $\mathcal{E}_T$ (voir la figure 20).

Nous voulons adapter cette technique pour réaliser un emboîtage de réseaux réguliers.

9.2.3 Emboîtage de réseaux réguliers de points

Nous appelons **emboîtage** l'opération qui consiste à inclure dans un voronoï "récepteur" d'un treillis à une résolution donnée, un même treillis tronqué de résolution supérieure (échelle inférieure). On dira que ce dernier est emboîté dans le voronoï récepteur. L'**emboîtement** est alors l'état de ce qui est emboîté. Nous voulons évidemment que l'**emboîtement** soit **optimal**, c.a.d que les voronoï du treillis emboîté s'ajustent exactement avec le voronoï récepteur, soit encore que le volume de ce voronoï récepteur ne soit remplit qu'avec des voronoï entiers du treillis emboîté. Cette dernière condition est souvent impossible à obtenir. Notre propos est simplement de décrire une méthode générale garantissant un **emboîtement presque optimal** où le volume du voronoï récepteur est remplit d'un maximum de voronoï entiers du treillis emboîté.
La figure 21 illustre le principe du changement d'échelle d'un réseau, les figures 23 et 24 présentent des exemples d'un emboîtement optimal et d'un autre presque optimal.
Pour raisonner il est plus simple de considérer que l'échelle du réseau à emboîter est fixe (nous considérons le

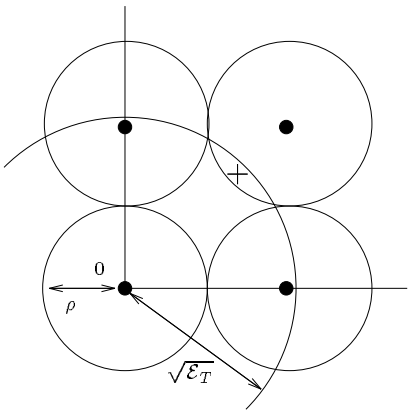

FIG. 20 - *Exemple bidimensionnel où le point à coder (la croix) est représenté par un point du réseau extérieur à la sphère d'énergie $\mathcal{E}_T$.*

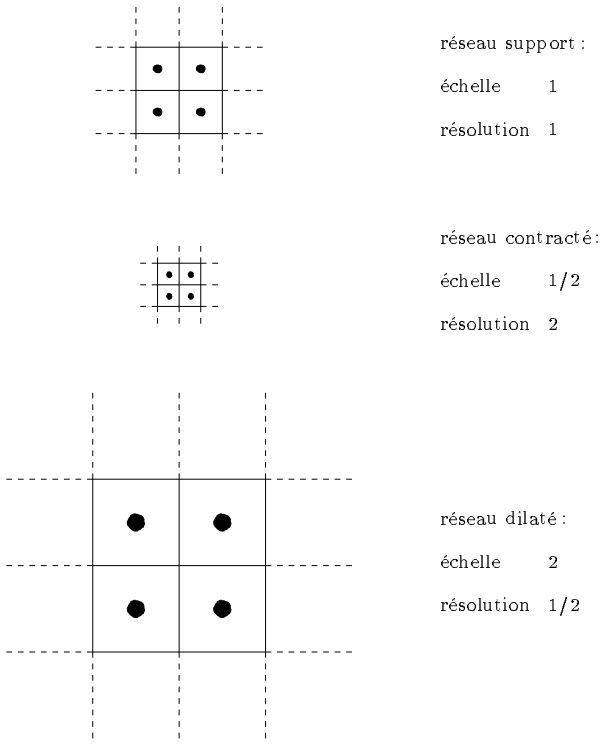

FIG. 21 - *Principe du changement d'échelle d'un réseau (exemple bidimensionnel).*

réseau support d'échelle 1), et que c'est l'échelle du voronoï récepteur que l'on fait varier : ce dernier, placé au centre du réseau support, est donc dilaté. Dans ce cas :

- si ρ et r sont respectivement les rayons d'empilement et de recouvrement caractéristiques du réseau support ;
- les rayons d'empilement et de recouvrement caractéristiques du voronoï récepteur [14] sont :

$$b.\rho \text{ et } b.r \text{ avec } b \in I\!R \ / \ b > 1$$

Ces 2 rayons sont aussi caractéristiques du réseau dilaté dont le voronoï récepteur est le centre.

Dans le réseau support les sphères de rayon ρ sont empilées régulièrement les unes contre les autres. Le point de contact entre 2 de ces sphères appartient nécessairement à l'hyperplan médiateur qui sépare les 2 points du treillis associés aux 2 sphères. Cet hyperplan est exactement tangent aux 2 sphères considérées. Enfin notons que l'empilement n'est pas modifié par un changement d'échelle du réseau.

Un emboîtage presque optimal est donc réalisé si on dilate le voronoï récepteur d'un facteur :

$$b = 2.n + 1 \ / \ n \in I\!N^*$$

Ainsi un maximum de points de contact entre sphères empilées du réseau correspondent à des points de contact entre sphères empilées du réseau dilaté. En ces points de contact communs, les hyperplans médiateurs entre points du réseau support sont inclus dans les hyperplans médiateurs entre points du réseau dilaté.

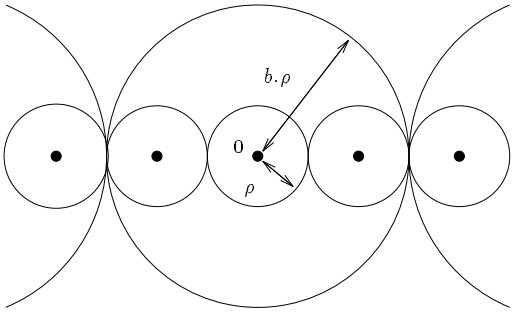

FIG. 22 - *Principe de l'emboîtage avec $n = 1$.*

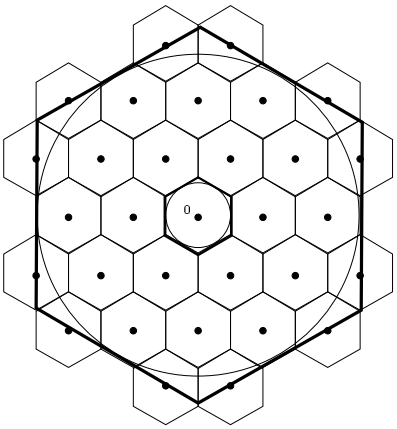

FIG. 23 - *Exemple d'un emboîtement presque optimal avec le réseau hexagonal ($n = 2$).*

9.2.4 Hiérarchie de réseaux réguliers emboîtés

La hiérarchie de treillis emboîtés est alors constituée de la suite des réseaux dont les échelles sont ajustées tels qu'ils puissent s'emboîtés de proche en proche. Le facteur d'échelle entre 2 réseaux consécutifs de la hiérarchie est donc b (voir la figure 25).

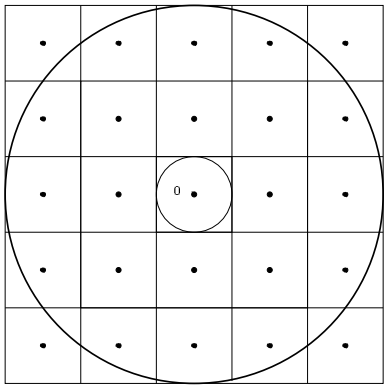

FIG. 24 - *Exemple d'un emboîtement optimal avec le réseau cubique ($n = 2$).*

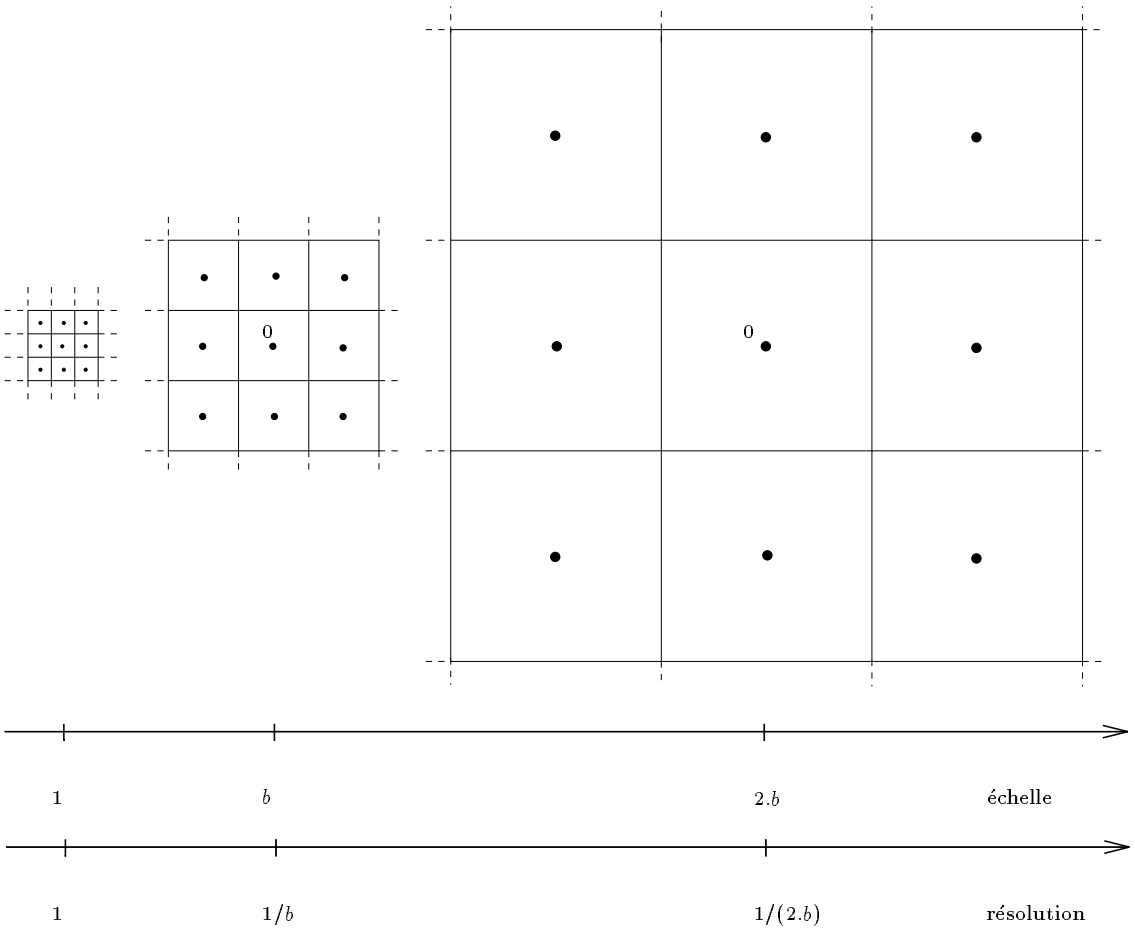

FIG. 25 - *Exemple de la hiérarchie de réseaux cubiques emboîtés ($n = 1$).*

9.2.5 Quantification à l'aide d'une hiérarchie de réseaux emboîtés

L'idée de la quantification vectorielle par emboîtage de réseaux réguliers de points est la suivante :

- le vecteur à coder est projeté un première fois à l'intérieur du réseau tronqué de résolution la plus grossière ;
- ensuite, en considérant le voronoï dans lequel ce point à coder se situe, ce vecteur peut-être quantifié plus finement en le projetant une nouvelle fois dans un nouveau réseau de résolution inférieure, l'opération peut-être décrite comme l'emboîtage d'un nouveau réseau tronqué dans ce voronoï ;
- cette opération peut-être répétée.

Il est évidemment plus pratique de modifier l'échelle du vecteur à coder plutôt que de manipuler plusieurs réseaux à différentes échelles. Ainsi on peut utiliser uniquement le réseau support tronqué et mettre en jeu le même algorithme de quantification rapide.
Nous voulons mettre en jeu à chaque étape de la quantification du vecteur le même réseau tronqué : il faut donc être certain que tous les vecteurs de la source seront inscrits, après la première étape de quantification, dans cet espace tronqué. C'est pourquoi le **premier facteur de projection** à appliquer a pour valeur :

$$F = \frac{b.\rho}{\sqrt{\mathcal{E}_{max}}}$$

où

- $\mathcal{E}_{max}$ est l'énergie maximale d'un vecteur à coder,
- ρ est le rayon d'empilement du réseau.

Dès lors un vecteur à coder est nécessairement inscrit dans un voronoï de cet espace tronqué. Si nous translatons ce vecteur à coder et son représentant au centre du réseau, on retrouve la cas de la figure 22. Le facteur de projection à appliquer au vecteur source décalé, pour le projeter dans le réseau emboîté suivant, est simplement b.

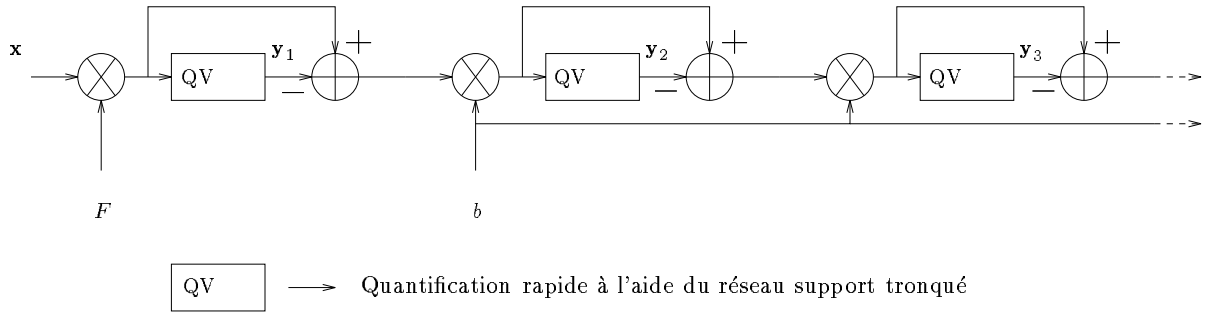

FIG. 26 - *Principe de la quantification d'un vecteur.*

Pour la figure 26, nous avons donc :

- $\mathbf{x}$ est le vecteur de la source à coder ;
- les $\mathbf{y}_i$ sont des vecteurs de reproduction du réseau tronqué (ce sont aussi les vecteurs de translation) ;
- $\mathcal{E}_{max}$ est l'énergie maximale possible d'un vecteur de la source ;
- $F = (b.\rho)/\sqrt{\mathcal{E}_{max}}$ est le facteur à appliquer pour projeter $\mathbf{x}$ dans le réseau tronqué ;
- b est le facteur pour projeter le vecteur translaté dans le réseau emboîté suivant.

Ainsi, une fois le réseau support et $b(n)$ choisis, seul $\mathcal{E}_{max}$ doit-être déterminée.

Nous rappelons qu'à chaque étage, la quantification (rapide) est réalisée à l'aide du même algorithme [11].

Cependant il ne faut pas confondre notre schéma avec un schéma classique de quantification par étages ("multistages VQ" [24] [18]) car dans notre cas :

- le nombre d'étages de quantification peut-être variable pour différents vecteurs (nous utiliserons par la suite cette propriété) ;
- les facteurs de projection sont déterminés automatiquement une fois $\mathcal{E}_{max}$ appréciée ;
- seul l'index qui correspond au vecteur de reproduction de l'étage final sera transmis.

Remarque :

La **valeur "réelle" du vecteur de reproduction** associé au vecteur $\mathbf{x}$ de la figure 26 est donc :

$$\frac{1}{F}.\mathbf{y}_1 + \frac{1}{F.b}.\mathbf{y}_2 + \frac{1}{F.b^2}.\mathbf{y}_3 + \ldots = \frac{1}{F.\sum_{i=1} b^{i-1}}.\mathbf{y}_i$$

L'**erreur "réelle" faite** est donc appréciée en calculant :

$$\mathbf{x} - \frac{1}{F.\sum_{i=1} b^{i-1}}.\mathbf{y}_i$$

9.2.6 Dénombrement des points du réseau emboîté

Il est important, pour des raisons pratiques d'allocation mémoire et afin d'indexer les vecteurs, de dénombrer les points du réseau emboîté et donc tronqué. Cette opération de dénombrement n'est pas évidente. Par exemple, considérons le cas où le vecteur à coder se situe dans le voronoï au centre du réseau support. Ce vecteur est projeté dans le réseau emboîté en lui appliquant le facteur de projection b. Cependant le point du réseau emboîté susceptible d'être mis en jeu n'a pas nécessairement son voronoï partiellement ou entièrement à l'intérieur de la sphère de rayon $b.\rho$ (voir la figure 27).
Nous nous proposons de majorer le nombre de points du réseau tronqué, en majorant l'énergie de la sphère les

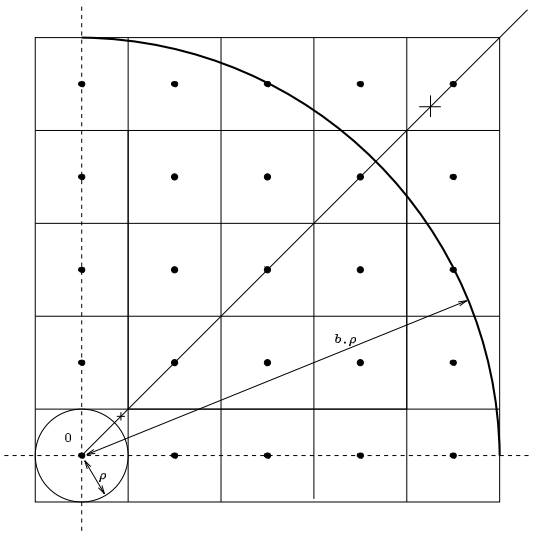

FIG. 27 - *Exemple, à l'aide de $\mathbf{Z}^2$, illustrant le cas où le vecteur à coder (symbolisé par la petite croix) une fois projeté (la grande croix) dans le réseau emboîté, est représenté par un point de reproduction dont le voronoï n'appartient à la sphère de rayon $b.\rho$.*

contenant, alors, en utilisant la série Thêta du réseau [14], nous pourrons dénombrer ces points.
Cette majoration est obtenue ainsi : le point projeté dans l'espace tronqué peut avoir comme plus haute énergie $(b.r)^2$, il correspond dans ce cas à un trou du réseau dilaté. Son représentant dans le réseau emboîté peut-être à l'extérieur de la sphère de rayon $b.r$, cependant ce représentant ne peut-être à une distance supérieure de r du vecteur qu'il représente.

Par conséquent le nombre de points du réseau emboîté (et donc tronqué) est majoré en considérant le nombre de points à l'intérieur et sur la sphère de rayon $(b+1).r$.

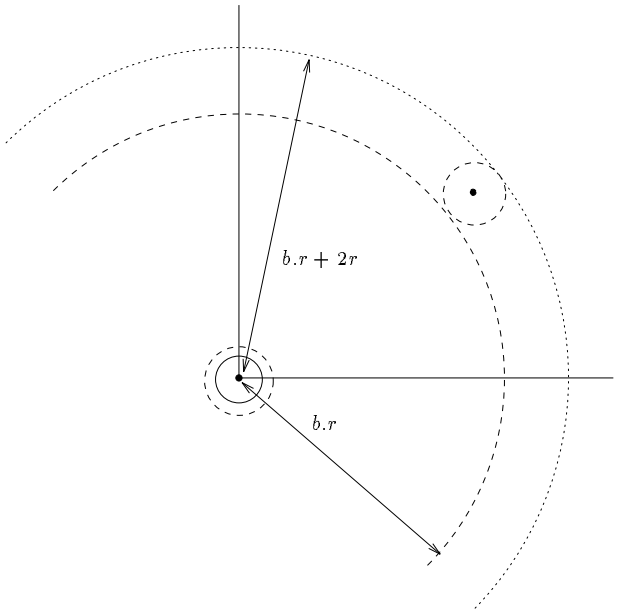

FIG. 28 - *Majoration de l'énergie de la sphère contenant les points du réseau tronqué : ces points sont nécessairement inscrits dans la sphère de rayon* $(b.r) + r$.

Cette majoration demeure grossière si nous considérons n petit. A présent nous nous intéressons au cas particulier où $n = 1$ (ce choix de n pour la quantification sera justifié par la suite).

Cas où $n = 1$

Nous pouvons alors affiner la majoration (c.a.d obtenir une borne supérieure plus proche du nombre exact de points du réseau emboîté).
Nous reprenons l'exemple précédent où le vecteur à coder, inclu dans le voronoï central du réseau support, est projeté dans le réseau emboîté par le facteur $b = 3$ ($b = 2.n + 1$ / $n = 1$).

Dans ce cas ($b = 3$) nous voulons montrer que le réseau emboîté (tronqué) est constitué de points à l'intérieur ou sur la sphère de rayon $3.r$.

1. Soit le point à coder, avant projection, est à l'intérieur de la sphère de rayon ρ. Une fois projeté, il est à l'intérieur de la sphère de rayon $3.\rho$ et le résultat est donc vérifié (voir la figure 29 [a]).
2. Le cas qu'il faut mieux étudier est lorsque le point à coder avant projection est un trou (il est donc sur la sphère de rayon r), projeté ce point est aussi un trou du réseau dilaté et il est à la distance maximale du centre du réseau emboîté :
   - un cas de figure se produit lorsque le rayon de recouvrement r caractéristique du réseau support est maximal, alors les sphères empilées de ce réseau forment entre elles un angle de $\pi/2$ et, suivant un axe, les sphères de rayon r ne se recouvrent pas (elles sont empilées). Dans ce cas, le point à coder projeté est encore un trou du réseau support : il est donc à la même distance de 2 points du réseau emboîté, et celui qui est choisi comme représentant est à l'intérieur de la sphère de rayon $3.r$ (pour $k = 2$, il est possible d'établir la relation : $r = \rho.\sqrt{2}$, voir la figure 29 [b]) ;
   - un autre cas de figure se produit lorsque r est minimal, les sphères empilées sur la sphère centrale de ce réseau forment entre elles un angle de $\pi/3$. Dans ce cas, le point à coder projeté se situe au centre d'un voronoï du réseau support et son représentant est sur la sphère de rayon $3.r$ (pour $k = 2$, il est possible d'établir la relation $r = \rho.2/\sqrt{3}$, voir la figure 29 [c]) ;
   - pour les autres cas de figure le point à coder projeté est donc dans un voronoï du réseau emboîté et tel que son représentant associé soit à l'intérieur de la sphère de rayon $3.r$. Plus r est grand, plus ce représentant est proche du rayon $3.r$.

3. Enfin, le point à coder peut se situer, avant projection, entre les sphères de rayon ρ et r, il demeure toujours inscrit dans le voronoï central. Ce dernier est un polytope convexe, ses frontières sont des hyperplans, aux sommets des intersections entre ces hyperplans sont les trous. Ce point à coder se situe donc dans un "coin" du voronoï proche d'un trou. Nous avons montré qu'un trou projeté à son représentant inscrit à l'intérieur de la sphère de rayon $3.r$. Ici, le facteur de projection ayant une valeur petite, ce point à coder projeté a alors une forte probabilité d'être représenté de la même façon que le trou projeté qui lui est proche.

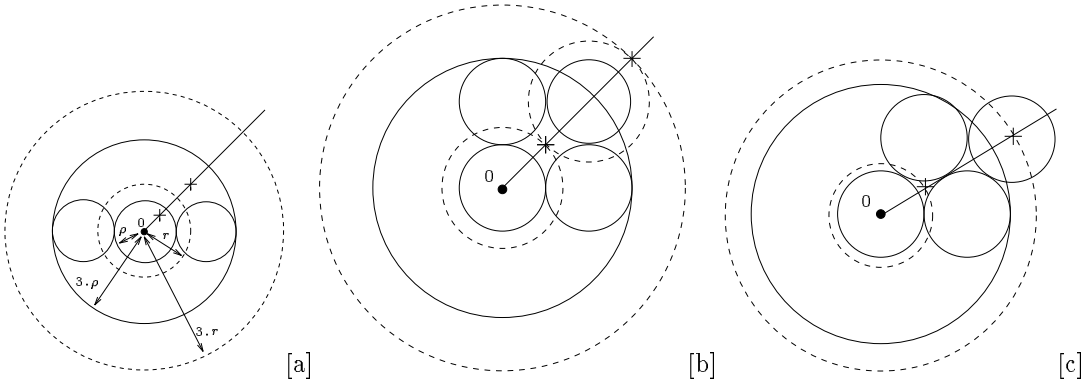

FIG. 29 - *Exemples (avec $k = 2$) illustrant que, pour $b = 3$, le réseau emboîté est constitué de points à l'intérieur ou sur la sphère de rayon $3.r$: le point à coder, avant et après projection, est symbolisé par une croix. Nous retrouvons respectivement les cas 1 ([a]) et 2 ([b] où r est maximal et [c] où r est minimal).*

9.3 Un dictionnaire arborescent

Nous rappelons que le vocabulaire relatif aux arbres est détaillé au **7.1.1**
Le dictionnaire obtenu par emboîtage d'une hiérarchie de réseaux régulier de points à une **structure arborescente** où chaque noeud de l'arbre est étiqueté par une point d'un réseau :

- la racine est étiquetée par le point 0 auquel est associé l'espace source en entier ;
- les fils d'un noeud sont les points du réseau emboîté dans le voronoï associé à ce noeud père : l'arbre est donc B-aire, B étant égal au nombre de points d'un réseau tronqué emboîté ;
- à chaque profondeur de l'arbre correspond une échelle de la hiérarchie : plus la couche est profonde, plus la résolution est fine.

Nous présentons aux figures 30, 31 et 32, des exemples d'étapes de construction de dictionnaires. La hiérarchie est constituée de réseaux $\mathbf{Z}^2$ et $b = 3$. Les vecteurs de la source sont les points blancs, leurs coordonnées indépendantes obéissent, respectivement pour chacune des figures, aux lois normale, laplacienne et gaussienne généralisée (pour laquelle $\alpha = 0.6$). Exactement ces figures détaillent le cas où les étages de quantification se succèdent. Un vecteur représentant et son voronoï (tous les deux en noir) ne sont représentés que s'ils sont mis en jeu lors de quantification. Les arbres obtenus sont équilibrés.
Ces exemples montrent comment, au fur et à mesure que l'arbre croît, l'échelle des treillis emboîtés décroît : les erreurs de quantification deviennent plus petite mais le nombre de représentants (et donc le débit) augmentent. Ces figures illustrent également comment la QVEHRRP adapte la découpe de l'espace à la répartition spatiale (la statistique) des vecteurs source.

Le dictionnaire est construit à l'aide d'une technique d'apprentissage, il est donc possible de caractériser numériquement (en termes de codage débit-distorsion) chacun des noeuds n_i de l'arbre $\mathcal{T} = \{n_0, n_1, n_2, \ldots\}$. Tout d'abord nous définissons :

- $\mathbf{y}_{n_i}$: le représentant associé à n_i ;
- C_{n_i} : le voronoï associé à n_i ;
- $SA = \{\mathbf{x}_i \ / \ i = 0, 1, 2, \ldots, N\}$: la séquence d'apprentissage ;
- $N = \text{card}(SA)$: la taille de la séquence d'apprentissage ;
- $\text{card}(C_{n_i})$: le nombre de vecteurs de la séquence d'apprentissage projetés dans C_{n_i} lors de la construction du dictionnaire.

Ainsi, lors de l'encodage de la séquence d'apprentissage, **à chaque noeud n_i de $\mathcal{T}$ est associé** :

- une probabilité d'occurence :
$$P(n_i) = \frac{\text{card}(C_{n_i})}{L}$$
- une distorsion moyenne :
$$d(n_i) = \frac{1}{\text{card}(C_{n_i})} \cdot \sum_{\mathbf{x} \in C_{n_i}, \ \mathbf{x} \in SA} \|\mathbf{x} - \mathbf{y}_{n_i}\|^2$$
- une longueur de mot du code binaire entropique [bits/vecteur] :
$$l(n_i) = -\log_2 P(n_i)$$

Nous considérons donc, qu'une fois l'arbre construit, **un codage entropique des feuilles est effectué**.

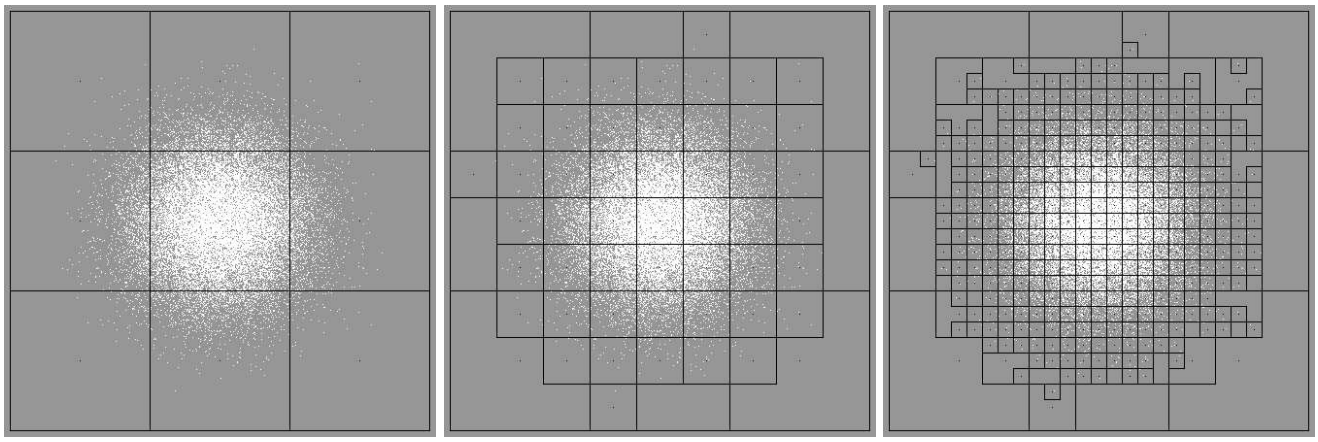

FIG. 30 - *[a][b][c] - Construction du dictionnaire par emboîtage d'une hiérarchie de réseaux* $\mathbf{Z}^2$ *(*$b = 3$*), mettant en jeu une étape [a], deux étapes [b] et trois étapes [c] de quantification. Les points blancs sont les vecteurs source dont les coordonnées i.i.d obéissent à une loi normale.*

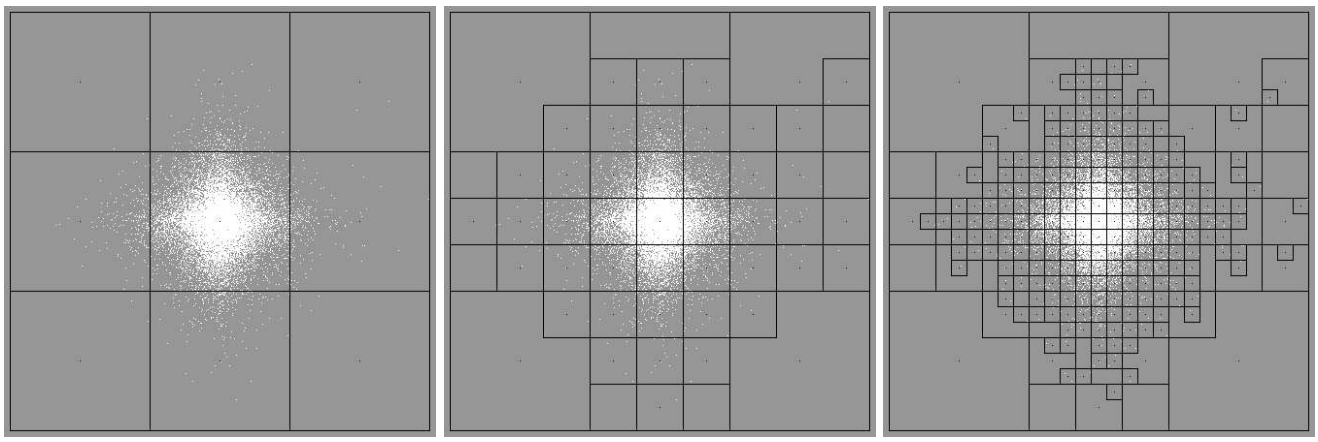

FIG. 31 - *[a][b][c] - Les points blancs sont les vecteurs source dont les coordonnées i.i.d obéissent à une loi laplacienne.*

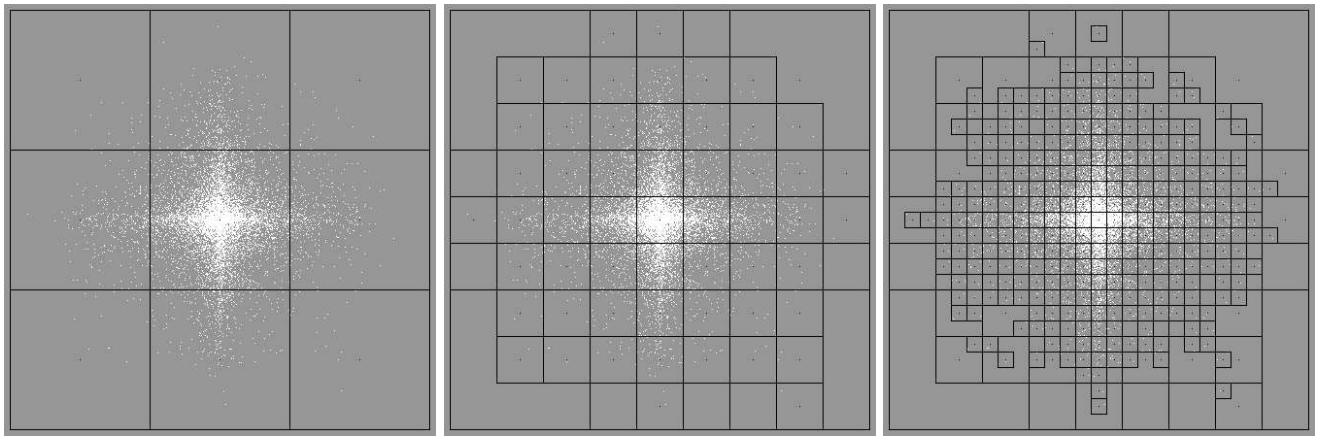

FIG. 32 - *[a][b][c] - Les points blancs sont les vecteurs source dont les coordonnées i.i.d obéissent à une loi gaussienne généralisée (*$\alpha = 0.6$*).*

Il devient aisé de **caractériser un sous-arbre** $\mathcal{S}$ de $\mathcal{T}$ par :

- la distorsion moyenne associée aux feuilles de $\mathcal{S}$:

$$d(\mathcal{S}) = \sum_{n_u \in \tilde{\mathcal{S}}} P(n_u).d(n_u)$$

- la longueur moyenne des mots du code binaire entropique associée aux feuilles de $\mathcal{S}$:

$$l(\mathcal{S}) = \sum_{n_u \in \tilde{\mathcal{S}}} P(n_u).l(n_u)$$

L'arbre entier est lui défini par $d(\mathcal{T}) = D$ et $l(\mathcal{T}) = R$.

Par simplification nous utiliserons souvent les termes "distorsion" et "débit" au lieu de "distorsion moyenne" et "longueur moyenne des mots du code binaire entropique".

Alors que les étapes de la construction du dictionnaire (c.a.d les emboîtages) s'enchaînent, R croît (ou est laissé inchangé) et D décroît (ou est laissé inchangée), les **fonctionnelles** $d(\mathcal{S})$ et $l(\mathcal{S})$ sont dites **monotones** (respectivement croissante et décroissante). Ces fonctionnelles sont aussi dites **linéaires** car leur résultat est la somme de leurs valeurs aux feuilles [9] [18].
L'**arrêt de cette construction du dictionnaire** est décidé si un débit limite est atteint (arrêt si $R \geq R_{seuil}$) ou si la qualité de la reconstruction de la source est jugée suffisante (arrêt si $D \leq D_{seuil}$).
L'interprétation, relative à la **courbe débit-distorsion** du codeur, est simple car la paire (R, D) d'un arbre donné correspond à un point de cette courbe : à l'origine l'arbre est la racine ($R = 0$ et D est égale à la variance de la séquence d'apprentissage), le point correspondant de la courbe $D(R)$ est en haut à gauche. Puis, à fur et à mesure que l'arbre croît par des emboîtage successifs, le point de la courbe $D(R)$ se positionne de plus en plus bas et sur la droite jusqu'à franchir D_{seuil} ou R_{seuil}.

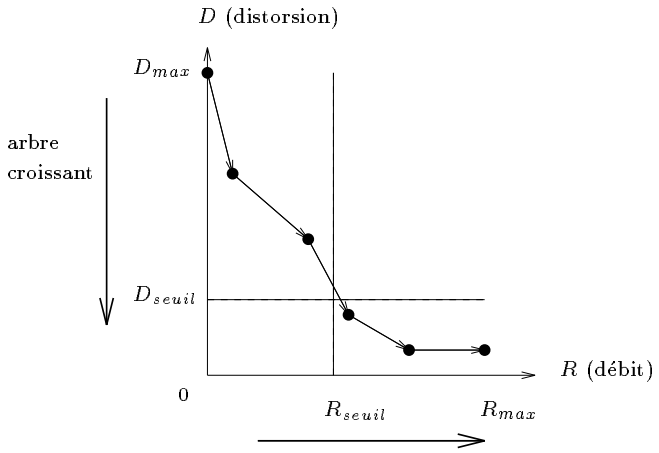

FIG. 33 - *Schéma de la courbe débit-distorsion expérimentale relative à la construction d'un dictionnaire par emboîtage successif de réseaux.*

9.4 Construction d'un dictionnaire arborescent non-équilibré

Les dictionnaires des figures 30, 31 et 32 sont des arbres équilibrés (tous les noeuds terminaux sont à la même profondeur), les vecteurs source sont finalement quantifiés à l'aide d'un ensemble de réseaux de même échelle. Mais il est plus intéressant de répartir le débit alloué au quantificateur afin que des zones de l'espace soient plus finement quantifiées que d'autres, l'arbre alors construit est non-équilibré. Ainsi, une région de l'espace correspondant à des feuilles profondes est finement quantifiée à l'aide de réseaux de petites échelles et, une région de l'espace correspondant à des feuilles peu profondes est grossièrement quantifiée à l'aide de réseaux de grande échelle.
Notons que pour contrôler une telle découpe de l'espace, il est nécessaire qu'elle se fasse progressivement : **le nombre de points représentants injectés à chaque nouvel emboîtage doit-être limité** [15]. De cette

façon les zones spatiales à découper sont localisées de façon précise et les bits alloués sont répartis avec justesse. N'oublions pas qu'un codage entropique des feuilles est efficace que si, celles ayant une probabilité faible, sont peu nombreuses. Ceci conduit à **fixer le facteur d'échelle** entre réseaux consécutifs de la hiérarchie avec une valeur minimale :

$$\boxed{\text{b(n =1)} = 3}$$

Afin de construire un dictionnaire arborescent non-équilibré nous adoptons, naturellement dans ce contexte de codage, **un critère débit-distorsion** pour la détermination des noeuds à découper. Pour simplifier, il s'agit de faire un compromis : lors de la construction du dictionnaire, les régions de l'espace où la distorsion est élevée sont découpées tout en évitant que le coût en terme de débit soit trop important. Pour construire un tel dictionnaire adapté au critère débit-distorsion, nous avons exploré 2 grandes stratégies classiques :

- une d'élagage de l'arbre,
- une autre de découpage de l'arbre.

9.4.1 Elagage de l'arbre

Il s'agit de l'**algorithme de BFOS généralisé** (BFOS pour Breiman, Friedman, Olshen et Stone [9] [18]) qui permet de réaliser un **élagage optimal** de l'arbre car l'**approche** est **globale**.
Avec cet algorithme, un arbre équilibré de grande taille (profond) doit d'abord être construit. Le stockage en mémoire des informations nécessaires pour la construction de ce dictionnaire (cette quantité d'information, qui sera détaillée, est directement proportionnelle au nombre de représentants) implique que cette méthode convient uniquement aux cas de faibles dimensions spatiales (k **petit**) lorsque le nombre B d'aires de l'arbre est réduit.
Une fois cet arbre complet construit, **un processus itératif d'élagage** est mis en place : à chaque boucle, suivant un critère débit-distorsion, un branche de l'arbre est supprimée.
Exactement si $\mathcal{S}_{n_i}$ est une branche de l'arbre complet $\mathcal{T}$, nous déterminons :

- la distorsion associée aux feuilles de $\mathcal{S}_{n_i}$:

$$d(\mathcal{S}_{n_i}) = \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).d(n_u)$$

- le débit associé aux feuilles de $\mathcal{S}_{n_i}$:

$$l(\mathcal{S}_{n_i}) = \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).l(n_u)$$

- la hausse de la distorsion si $\mathcal{S}_{n_i}$ est supprimée (hausse définie positive) :

$$\Delta d(\mathcal{S}_{n_i}) = P(n_i).d(n_i) - d(\mathcal{S}_{n_i})$$

- la baisse du débit si $\mathcal{S}_{n_i}$ est supprimée (baisse définie positive) :

$$\Delta l(\mathcal{S}_{n_i}) = l(\mathcal{S}_{n_i}) - P(n_i).l(n_i)$$

Par définition :

$$\lambda(n_i) = \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})}$$

est le **retour marginal** associé à la racine n_i de $\mathcal{S}_{n_i}$.

Nous allons donc, à chaque boucle de l'algorithme d'élagage, **supprimer la branche dont la racine indique un retour marginal minimal**, c.a.d celle dont la suppression entraîne une hausse de la distorsion minimale et une baisse du débit maximale.
L'interprétation à l'aide de la **courbe débit-distorsion** est simple. A l'arbre complet correspond le point de $D(R)$ le plus bas à droite. A chaque itération un choix doit être fait parmi les branches à élaguer. A chacun de ces choix possible correspond un sous-arbre élagué et un point débit-distorsion. Le point de la courbe $D(R)$ désigné par le critère du retour marginal minimal est celui offrant le meilleur compromis débit-distorsion : la courbe tracée circule suivant l'enveloppe convexe de tous les points possibles de $D(R)$. Chaque $\lambda(n_i)$ peut donc s'interpréter comme une portion possible de la pente de la courbe débit-distorsion.

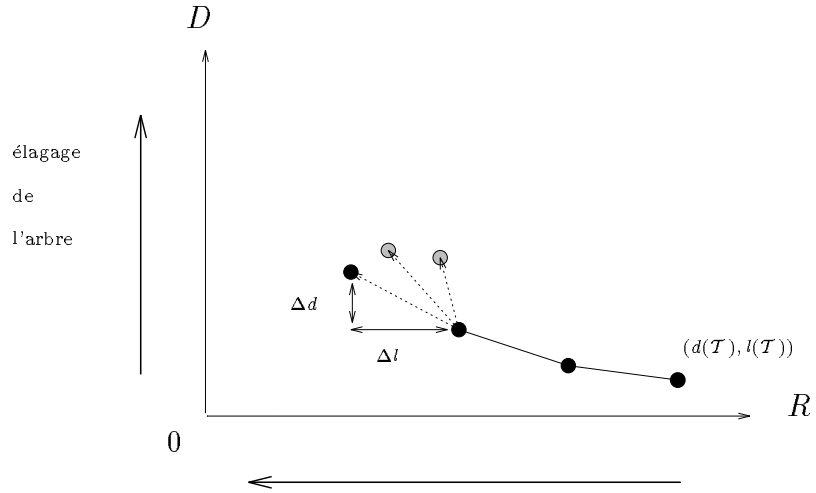

FIG. 34 - *Elagage de l'arbre: à une itération donnée de l'algorithme de BFOS, le point choisi est celui dont le retour marginal est minimal.*

Cette approche s'interprête aussi comme **une technique d'optimisation lagrangienne** où la fonctionnelle $J(\mathcal{S}) = d(\mathcal{S}) + \lambda.l(\mathcal{S})$ doit être minimisée, λ est donc le multiplicateur de Lagrange. En faisant varier λ, tous les points de la courbe débit-distorsion correspondant aux différents sous-arbres élagués possibles $\mathcal{S}$, peuvent-être testés et le plus intéressant choisi.

Après avoir construit l'arbre complet $\mathcal{T}$ et avoir déterminer pour ses noeuds n_i : $P(n_i)$, $d(n_i)$ et $l(n_i)$, il est nécessaire, avant de lancer le processus d'élagage, de calculer les retours marginaux associés à chacun de ces noeuds n_i. Heureusement une **formule de récurence** se met en place, elle permet, en progressant de proche en proche (des fils vers le père) à partir des feuilles jusqu'à la racine, de calculer les valeurs de $\Delta d(\mathcal{S}_{n_i})$ et $\Delta l(\mathcal{S}_{n_i})$. En effet :

- pour les feuilles :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= \Delta l(\mathcal{S}_{n_i}) = 0 \\ \lambda(n_i) &= +\infty \end{aligned}$$

- pour les autres noeuds, si n_i a B fils n_j (la démonstration suit) :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= P(n_i).d(n_i) + \sum_B \Delta d(\mathcal{S}_{n_j}) - \sum_B P(n_j).d(n_j) \\ \Delta l(\mathcal{S}_{n_i}) &= \sum_B \Delta l(\mathcal{S}_{n_j}) + \sum_B P(n_j).l(n_j) - P(n_i).l(n_i) \\ \lambda(n_i) &= \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})} \end{aligned}$$

Démonstration :

Nous exploitons la propriété de linéarité des fonctionnelles $d(\mathcal{S})$ et $l(\mathcal{S})$. Nous avons :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= P(n_i).d(n_i) - d(\mathcal{S}_{n_i}) \\ &= P(n_i).d(n_i) - \sum_{n_u \in \tilde{\mathcal{S}}_{n_i}} P(n_u).d(n_u) \\ &= P(n_i).d(n_i) - \sum_B \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u) \end{aligned}$$

or :

$$\Delta d(\mathcal{S}_{n_j}) = P(n_j).d(n_j) - \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u) \Longleftrightarrow \sum_{n_u \in \tilde{\mathcal{S}}_{n_j}} P(n_u).d(n_u) = P(n_j).d(n_j) - \Delta d(\mathcal{S}_{n_j})$$

donc :

$$\Delta d(\mathcal{S}_{n_i}) = P(n_i).d(n_i) + \sum_B \Delta d(\mathcal{S}_{n_j}) - \sum_B P(n_j).d(n_j) \qquad \text{(C.Q.F.D)}$$

En procédant de la même façon :

$$\begin{aligned}
\Delta l(\mathcal{S}_{n_i}) &= l(\mathcal{S}_{n_i}) - P(n_i).l(n_i) \\
&= \sum_{n_u \in \bar{\mathcal{S}}_{n_i}} P(n_u).l(n_u) - P(n_i).l(n_i) \\
&= \sum_B \sum_{n_u \in \bar{\mathcal{S}}_{n_j}} P(n_u).l(n_u) - P(n_i).l(n_i)
\end{aligned}$$

et :

$$\Delta l(\mathcal{S}_{n_j}) = \sum_{n_u \in \bar{\mathcal{S}}_{n_j}} P(n_u).l(n_u) - P(n_j).l(n_j) \Longleftrightarrow \sum_{n_u \in \bar{\mathcal{S}}_{n_j}} P(n_u).l(n_u) = \Delta l(\mathcal{S}_{n_j}) + P(n_j).l(n_j)$$

donc :

$$\Delta l(\mathcal{S}_{n_i}) = \sum_B \Delta l(\mathcal{S}_{n_j}) + \sum_B P(n_j).l(n_j) - P(n_i).l(n_i) \qquad \text{(C.Q.F.D)}$$

□

Nous remarquons que la **distorsion et le débit associés à l'arbre** $\mathcal{T}$ sont respectivement :

$$\begin{aligned}
d(\mathcal{T}) &= \sum_{n_u \in \bar{\mathcal{T}}} P(n_u).d(n_u) \\
&= P(n_0).d(n_0) - \Delta d(\mathcal{S}_{n_0}) \\
&= d(n_0) - \Delta d(\mathcal{S}_{n_0})
\end{aligned}$$

$$\begin{aligned}
l(\mathcal{T}) &= \sum_{n_u \in \bar{\mathcal{T}}} P(n_u).l(n_u) \\
&= P(n_0).l(n_0) + \Delta l(\mathcal{S}_{n_0}) \\
&= 0 + \Delta l(\mathcal{S}_{n_0}) \\
&= \Delta l(\mathcal{S}_{n_0})
\end{aligned}$$

Ensuite, après chaque élagage, il est nécessaire de remettre à jour les retours marginaux des ascendants (les pères) de la racine de la branche supprimée. Si nous définissons :

- le sous-arbre élagué produit à l'issue de la boucle j de l'algorithme de BFOS :

$$\mathcal{S}^j \quad / \quad j = 1,\ 2,\ \ldots \quad (\mathcal{S}^0 = \mathcal{T})$$

- la branche, plantée en n_i, élaguée lors de cette boucle j :

$$\mathcal{S}^j_{n_i} \quad / \quad j = 1,\ 2,\ \ldots$$

- les ascendants (jusqu'à la racine n_0) de n_i : les n_k

Alors, **une fois $\mathcal{S}^j_{n_i}$ élaguée, nous remettons à jour** :

- les retours marginaux des n_k en appliquant (la démonstration suit) :

$$\begin{aligned}
\Delta d(\mathcal{S}^{j+1}_{n_k}) &= \Delta d(\mathcal{S}^j_{n_k}) - \Delta d(\mathcal{S}^j_{n_i}) \\
\Delta l(\mathcal{S}^{j+1}_{n_k}) &= \Delta l(\mathcal{S}^j_{n_k}) - \Delta l(\mathcal{S}^j_{n_i}) \\
\lambda(n_k) &= \frac{\Delta d(\mathcal{S}^{j+1}_{n_k})}{\Delta l(\mathcal{S}^{j+1}_{n_k})}
\end{aligned}$$

– le retour marginal de la nouvelle feuille n_i :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}^{j+1}) &= \Delta l(\mathcal{S}_{n_i}^{j+1}) = 0 \\ \lambda(n_i) &= +\infty \end{aligned}$$

Démonstration :

Là encore, nous exploitons la propriété de linéarité des 2 fonctionnelles.
$\mathcal{S}_{n_k}^{j+1}$ est un sous-arbre élagué du sous-arbre $\mathcal{S}_{n_k}^{j}$ ($\mathcal{S}_{n_i}^{j}$ étant la branche supprimée). Alors :

$$\begin{aligned} d(\mathcal{S}_{n_k}^{j}) &= \sum_{n_u \in \bar{\mathcal{S}}_{n_k}^{j}} P(n_u).d(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}_{n_i}^{j}} P(n_u).d(n_u) + \sum_{n_u \notin \bar{\mathcal{S}}_{n_i}^{j},\ n_u \in \bar{\mathcal{S}}_{n_k}^{j}} P(n_u).d(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}_{n_i}^{j}} P(n_u).d(n_u) + \sum_{n_u \in \bar{\mathcal{S}}_{n_k}^{j+1}} P(n_u).d(n_u) - P(n_i).d(n_i) \end{aligned}$$

donc :

$$\begin{aligned} d(\mathcal{S}_{n_k}^{j}) &= d(\mathcal{S}_{n_k}^{j+1}) + d(\mathcal{S}_{n_i}^{j}) - P(n_i).d(n_i) \\ &= d(\mathcal{S}_{n_k}^{j+1}) - \Delta d(\mathcal{S}_{n_i}^{j}) \end{aligned}$$

soit encore :

$$P(n_k).d(n_k) - \Delta d(\mathcal{S}_{n_k}^{j+1}) = P(n_k).d(n_k) - \Delta d(\mathcal{S}_{n_k}^{j}) + \Delta d(\mathcal{S}_{n_i}^{j}) \Longleftrightarrow \Delta d(\mathcal{S}_{n_k}^{j+1}) = \Delta d(\mathcal{S}_{n_k}^{j}) - \Delta d(\mathcal{S}_{n_i}^{j})$$

(C.Q.F.D)

En procédant de la même façon :

$$\begin{aligned} l(\mathcal{S}_{n_k}^{j}) &= \sum_{n_u \in \bar{\mathcal{S}}_{n_k}^{j}} P(n_u).l(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}_{n_i}^{j}} P(n_u).l(n_u) + \sum_{n_u \notin \bar{\mathcal{S}}_{n_i}^{j},\ n_u \in \bar{\mathcal{S}}_{n_k}^{j}} P(n_u).l(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}_{n_i}^{j}} P(n_u).l(n_u) + \sum_{n_u \in \bar{\mathcal{S}}_{n_k}^{j+1}} P(n_u).l(n_u) - P(n_i).l(n_i) \\ &= l(\mathcal{S}_{n_k}^{j+1}) + \sum_{n_u \in \bar{\mathcal{S}}_{n_i}^{j}} P(n_u).l(n_u) - P(n_i).l(n_i) \\ &= l(\mathcal{S}_{n_k}^{j+1}) + l(\mathcal{S}_{n_i}^{j}) - P(n_i).l(n_i) \\ &= l(\mathcal{S}_{n_k}^{j+1}) + \Delta l(\mathcal{S}_{n_i}^{j}) \end{aligned}$$

soit encore :

$$P(n_k).l(n_k) + \Delta d(\mathcal{S}_{n_k}^{j+1}) = P(n_k).l(n_k) + \Delta d(\mathcal{S}_{n_k}^{j}) - \Delta d(\mathcal{S}_{n_i}^{j}) \Longleftrightarrow \Delta l(\mathcal{S}_{n_k}^{j+1}) = \Delta l(\mathcal{S}_{n_k}^{j}) - \Delta l(\mathcal{S}_{n_i}^{j})$$

(C.Q.F.D)
□

La distorsion et le débit du nouveau sous-arbre élagué sont donc :

$$\begin{aligned} d(\mathcal{S}^{j+1}) &= d(\mathcal{S}^{j}) + \Delta d(\mathcal{S}_{n_i}^{j}) \\ l(\mathcal{S}^{j+1}) &= l(\mathcal{S}^{j}) - \Delta d(\mathcal{S}_{n_i}^{j}) \end{aligned}$$

Ce résultat, traduisant la hausse de la distorsion et la baisse du débit à chaque élagage, implique d'adapter le **critère de fin de la construction du dictionnaire**: il faut continuer à élaguer tant que la distorsion est jugée acceptable (continuer à élaguer tant que $d(\mathcal{S}^{j+1}) \leq D_{seuil}$), ou tant que le débit demeure supérieur au débit désiré (continuer à élaguer tant que $l(\mathcal{S}^{j+1}) \geq R_{seuil}$). Les vecteurs, associés aux feuilles du sous-arbre élagué obtenu, sont les représentants du dictionnaire final.
La figure 36 présente un **synoptique** simplifié de cet algorithme d'élagage (BFOS).

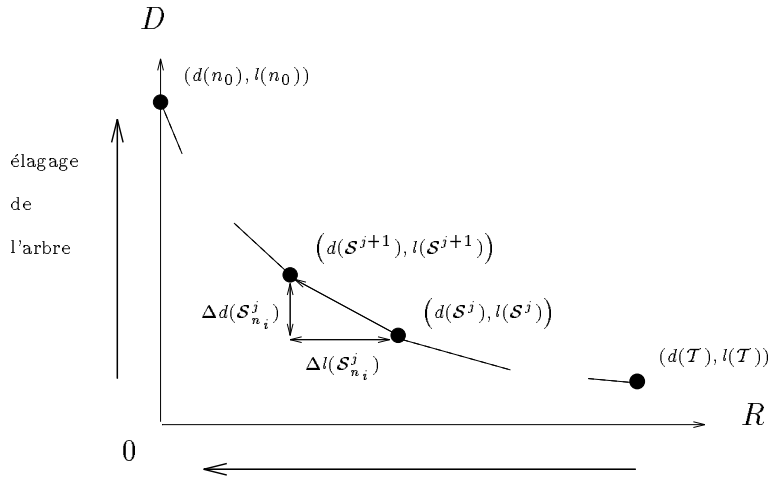

FIG. 35 - *Elagage de l'arbre : cas où, à l'itération j de l'algorithme de BFOS, la branche $\mathcal{S}^j_{n_i}$ est élaguée.*

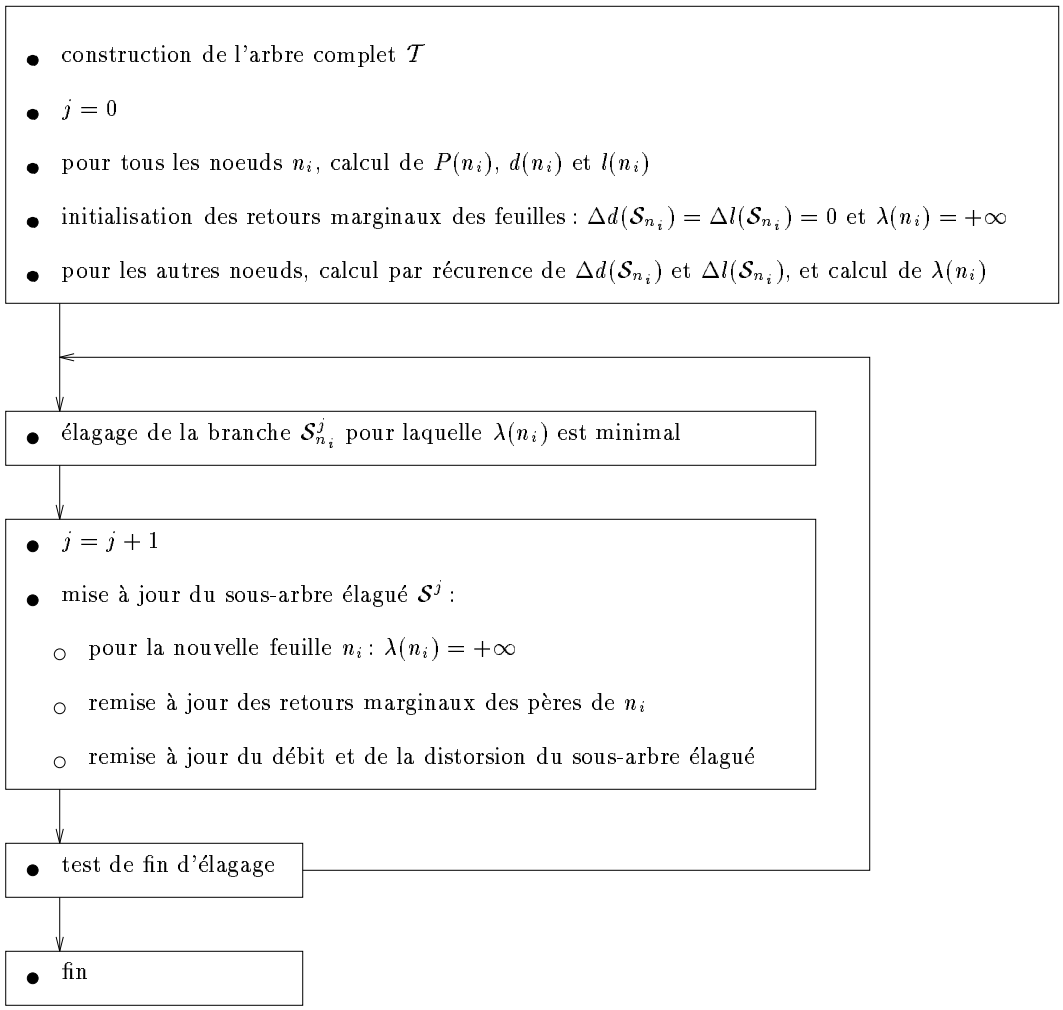

FIG. 36 - *Synoptique simplifié de l'algorithme d'élagage (BFOS).*

9.4.2 Découpage de l'arbre

Avec cet algorithme [33] [18] le dictionnaire arborescent est construit de façon progressive par **un processus itératif où, à l'issue de chaque boucle, une seule feuille de l'arbre demeure découpée** (un seul nouvel emboîtement a été réalisé). A l'origine, l'arbre n'est donc constitué que de la racine n_0. Précisément, à chaque boucle de l'algorithme de découpage :

- toutes les feuilles de l'arbre sont découpées (un nouveau réseau est emboîté dans chacun de leur voronoï associé) ;
- seule la branche, ainsi créée, présentant le meilleur compromis débit-distorsion est conservée (toutes les autres sont élaguées).

Cette technique est adaptée à la construction du dictionnaire lorsque **la dimension spatiale k est grande**, et par conséquent le nombre B d'aires de l'arbre élevé. En effet, avec cet algorithme, la quantité d'information à stocker est limitée car exactement ajustée à la taille de l'arbre qui croît progressivement. Cependant **cette approche est locale**, les résultats obtenus (en termes de débit-distorsion) sont donc sous-optimaux si nous les comparons à ceux obtenus à l'aide de l'approche globale d'élagage de l'arbre.
Pour détailler les calculs nous reprenons les notations déjà introduites. En particulier nous considérons obtenir, à l'issue de la boucle j de l'algorithme, un sous-arbre élagué $\mathcal{S}^j$ de l'arbre complet final $\mathcal{S}^l = \mathcal{T}$ qui lui, est obtenu au bout de l itérations ($l \geq j$) et tel que le critère de fin de construction du dictionnaire soit rempli.
A l'initialisation nous avons donc $\mathcal{S}^0 = n_0$. Pour la boucle $j (j \geq 1)$ du processus de découpage nous considérons :

- les feuilles du sous-arbre élagué $\mathcal{S}^{j-1}$: les n_i ;
- les nouvelles branches obtenues par le découpage des n_i (c.a.d l'emboîtage d'un nouveau réseau dans leur voronoï associé), sachant qu'une seule de ces branches sera conservée pour $\mathcal{S}^j$: les $\mathcal{S}^j_{n_i}$;
- les retours marginaux aux n_i déterminés en calculant :

$$\begin{aligned} \Delta d(\mathcal{S}_{n_i}) &= P(n_i).d(n_i) - d(\mathcal{S}_{n_i}), \\ \Delta l(\mathcal{S}_{n_i}) &= l(\mathcal{S}_{n_i}) - P(n_i).l(n_i), \\ \lambda(n_i) &= \frac{\Delta d(\mathcal{S}_{n_i})}{\Delta l(\mathcal{S}_{n_i})}. \end{aligned}$$

Nous allons **conserver comme nouvelle branche pour $\mathcal{S}^j$ celle** offrant le meilleur compromis débit-distorsion, c'est à dire celle dont l'ajout offre une baisse de la distorsion maximale pour une hausse du débit minimale, il s'agit donc de la branche $\mathcal{S}^j_{n_i}$ **dont le retour marginal $\lambda(n_i)$ est maximal**.
L'interprétation à l'aide de la **courbe débit-distorsion** nous est devenue familière : à l'initialisation, le sous-arbre élagué $\mathcal{S}^0$ correspond au point de $D(R)$ en haut à gauche. A chaque nouvelle itération j de l'algorithme, un choix doit être fait parmi les $\mathcal{S}^j_{n_i}$ de la branche à ne pas élaguer, à chacun de ces choix possibles correspond un nouveau point de $D(R)$. Le critère du retour marginal maximal permet de désigner le point offrant **localement** le meilleur compromis baisse de la distorsion vs. hausse du débit.

La distorsion et le débit associés au nouveau sous-arbre élagué S^j sont donnés par (la démonstration suit) :

$$d(\mathcal{S}^j) = d(\mathcal{S}^{j-1}) - \Delta d(\mathcal{S}^j_{n_i})$$

$$l(\mathcal{S}^j) = l(\mathcal{S}^{j-1}) + \Delta l(\mathcal{S}^j_{n_i})$$

Démonstration :

$$\begin{aligned} d(\mathcal{S}^j) &= \sum_{n_u \in \bar{\mathcal{S}}^j} P(n_u).d(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}^{j-1}} P(n_u).d(n_u) + \sum_{n_u \in \bar{\mathcal{S}}^j_{n_i}} P(n_u).d(n_u) - P(n_i).d(n_i) \\ &= d(\mathcal{S}^{j-1}) - \Delta d(\mathcal{S}^j_{n_i}) \end{aligned}$$

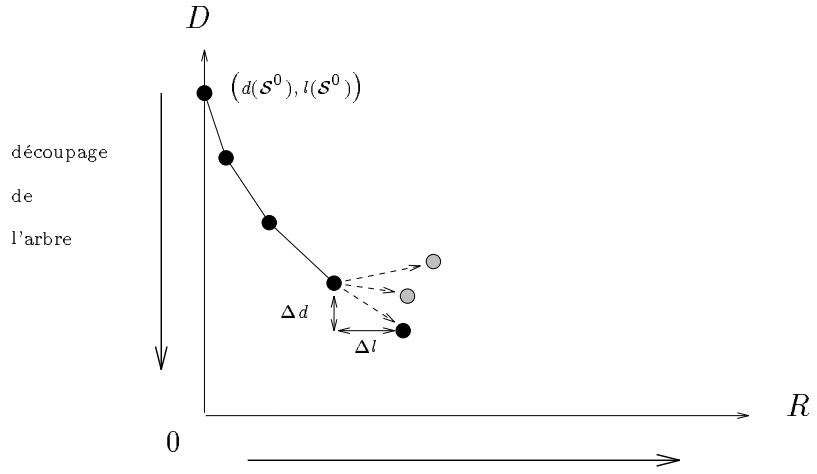

FIG. 37 - *Découpage de l'arbre: à une itération donnée de l'algorithme, le point choisi est celui dont le retour marginal est maximal.*

$$\begin{aligned} l(\mathcal{S}^j) &= \sum_{n_u \in \bar{\mathcal{S}}^j} P(n_u).l(n_u) \\ &= \sum_{n_u \in \bar{\mathcal{S}}^{j-1}} P(n_u).l(n_u) + \sum_{n_u \in \bar{\mathcal{S}}^j_{n_i}} P(n_u).l(n_u) - P(n_i).l(n_i) \\ &= l(\mathcal{S}^{j-1}) + \Delta l(\mathcal{S}^j_{n_i}) \end{aligned}$$

□

Ces paramètres sont les seuls à devoir être remis à jour à chaque itération du processus de découpage. Ce résultat traduit évidemment la baisse de la distorsion et la hausse du débit réalisées par ce nouvel emboîtage. L'arbre croissant au fur et à mesure des découpages, nous retrouvons le **critère d'arrêt de la construction du dictionnaire** du **9.3** : cette construction est stoppée si un débit limite est atteint (arrêt si $l(\mathcal{S}^j) \geq R_{seuil}$), ou si la qualité de la reconstruction de la source est jugée suffisante (arrêt si $d(\mathcal{S}^j) \leq D_{seuil}$).
La figure 39 présente un **synoptique** simplifié de cet algorithme de découpage de l'arbre.

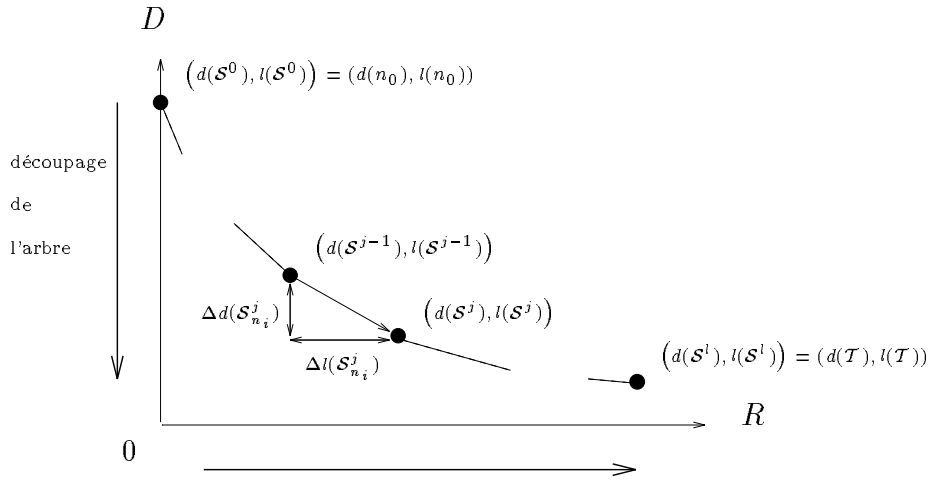

FIG. 38 - *Découpage de l'arbre: cas où, à l'itération j de l'algorithme, la branche $\mathcal{S}^j_{n_i}$ est ajoutée.*

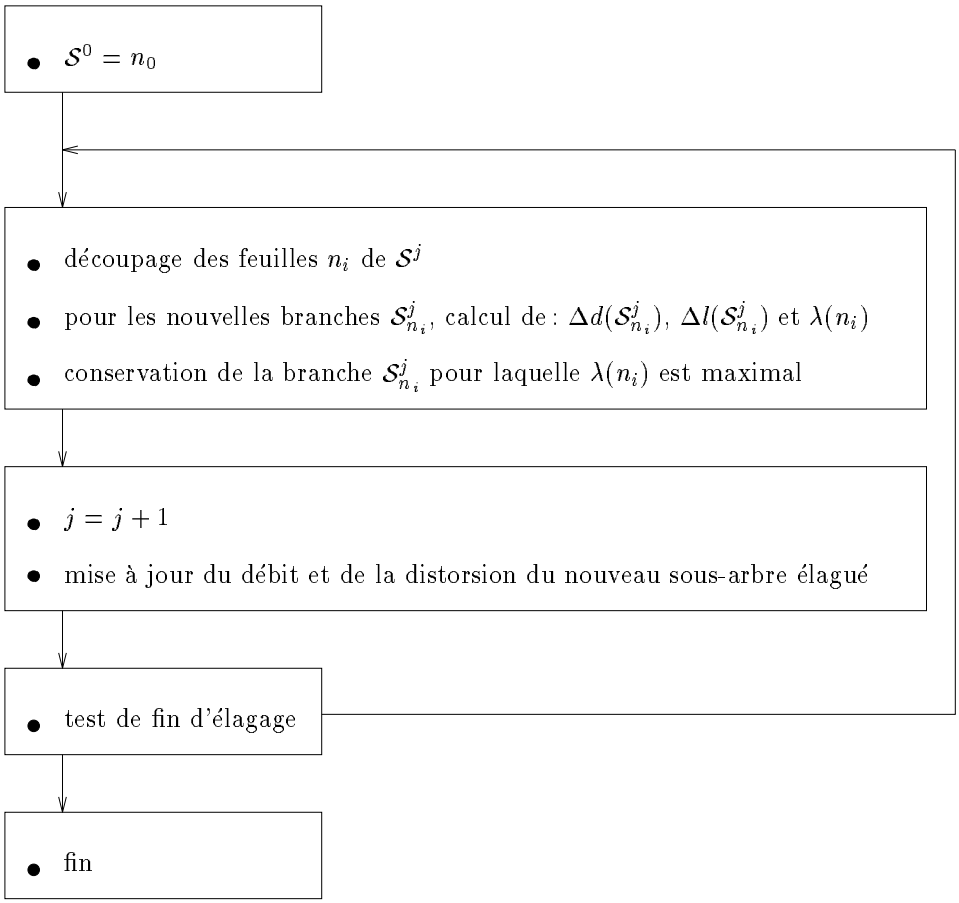

FIG. 39 - *Synoptique simplifié de l'algorithme de découpage de l'arbre.*

10 Résultats expérimentaux

Des premiers résultats de quantification de sources synthétiques et d'une source réelle sont présentés. Le facteur d'apprentissage [18], qui est défini comme le rapport du nombre de vecteurs d'apprentissage sur celui des vecteurs représentant (les feuilles de l'arbre), demeure supérieur à 150 pour les sources synthétiques et 100 pour la source réelle. Le débit considéré étant entropique, ce facteur est introduit afin de limiter le nombre des vecteurs de reproduction. Cependant il implique que la taille de la séquence d'apprentissage soit adaptée à la dimension vectorielle : pour un débit donné, plus la dimension de l'espace est grande, plus le nombre de représentants mis en jeu peut-être important, la taille de la séquence d'apprentissage doit donc augmenter afin de disposer d'un champ de vecteurs à coder suffisamment dense et représentatif de la statistique de la source.

La figure 40 montre des exemples d'arbres élagués issus de la quantification de sources synthétiques sans mémoire dont les échantillons obéissent respectivement à la loi normale, à la loi laplacienne et à celle gaussienne généralisée avec $\alpha = 0.6$. Nous rappelons que ces fonctions de la famille des gaussiennes généralisées permettent de modéliser la distribution statistique de source d'images différentielles ou hybrides. Ces 3 exemples illustrent comment notre approche adapte la découpe de l'espace à la distribution des vecteurs source tout en suivant le critère débit-distorsion : pour un débit donné, le quantificateur découpe grossièrement la région de l'espace où se concentrent les vecteurs source peu énergétiques, alors la région spatiale moins dense où se situent les vecteurs source énergétiques peut-être découper plus finement.
Notre schéma est approprié à la quantification de telles images différentielles ou hybrides car ces vecteurs peu énergétiques et ayant la probabilité la plus grande sont ceux dont les composantes correspondent aux pixels des régions homogènes des images différentielles (c.a.d les régions de ces images où les erreurs de prédiction de compensation du mouvement ont peu d'amplitude, typiquement ce sont des régions non affectées par les objets en mouvement ou, l'intérieur de zones uniformes en mouvement). Ces vecteurs, pauvres en information pertinente, peuvent donc être fortement quantifiés.
Les vecteurs source énergétiques et de probabilité faible sont ceux dont les coordonnées correspondent aux pixels des régions inhomogènes des images différentielles (typiquement ce sont les zones de ces images, recouvertes ou découvertes par les objets en mouvement). Pour une bonne restitution des images, ces vecteurs doivent donc être finement quantifiés.

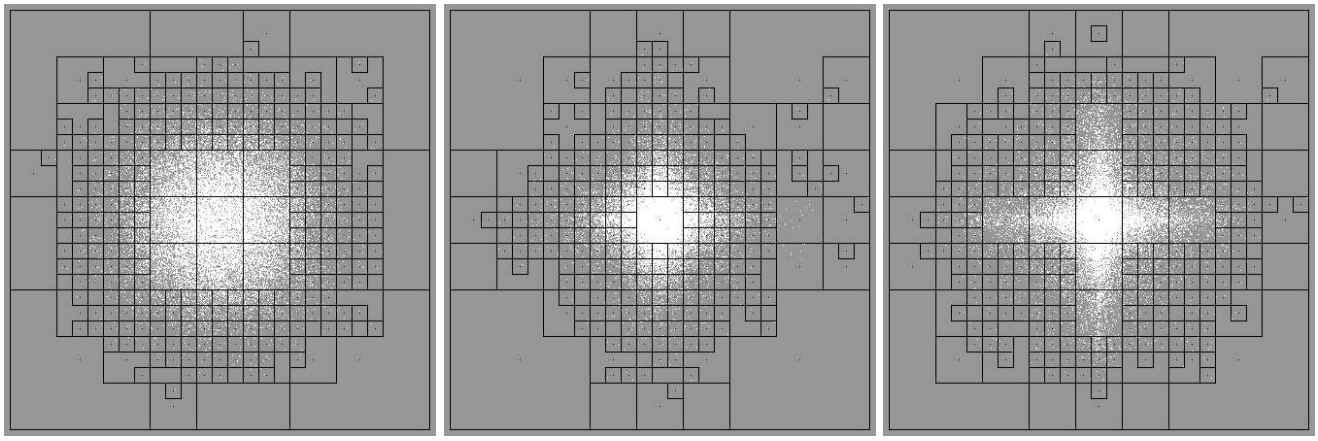

FIG. 40 - *[a][b][c] - Emboîtage de réseaux $\mathbf{Z}^2$, élagage de l'arbre, sources synthétiques: les points blancs sont les vecteurs source, leurs coordonnées i.i.d obéissent respectivement à une loi normale (a), à une loi laplacienne (b) et à une loi gaussienne généralisée (c).*

L'approche par découpage de l'arbre et quantification d'une source synthétique est illustrée par les figures 41 et 42. Celles-ci présentent les courbes expérimentales débit vs. distorsion (exactement entropie du dictionnaire vs. distorsion) obtenues par le codage d'une source sans mémoire dont les échantillons obéissent à la loi normale $\mathcal{N}(0,1)$. Pour comparer la borne de Shannon est tracée. L'intérêt de mettre en jeu des dimensions vectorielles supérieures est montré car la QVEHRRP donne de meilleurs résultats avec le réseau D_4 plutôt qu'avec $\mathbf{Z}^2$ (à débit identique, la distorsion est inférieure) et de plus bas débit sont atteints. Ces résultats sont mieux mis en évidence par la suite.

La figure 43 présente des courbes débit vs. PSNR (entropie du dictionnaire vs. Peak Signal-to-Noise Ratio) obtenues par la quantification de la source réelle. Excatement il s'agit d'une séquence d'images d'erreurs de prédiction de compensation du mouvement (une image extraite de cette séquence est montrée figure 44). Pour

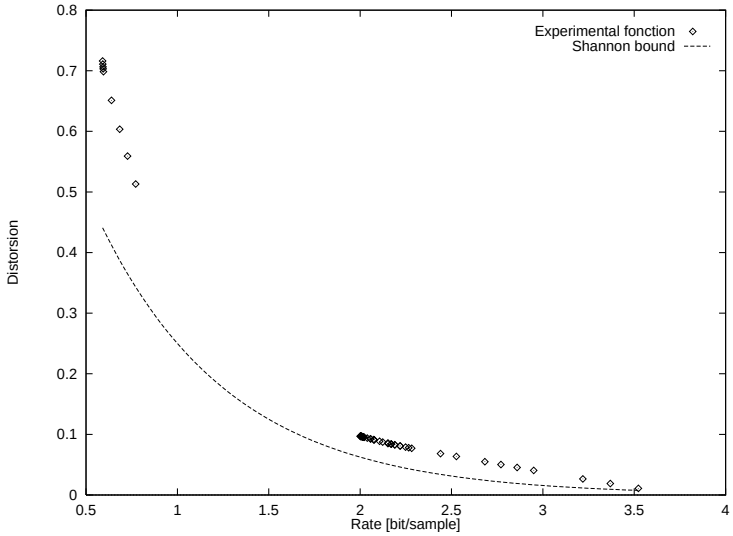

FIG. 41 - *Emboîtage de réseaux $\mathbf{Z}^2$, découpage de l'arbre, source synthétique : courbe débit vs. distorsion obtenue par quantification d'une source vectorielle dont les échantillons i.i.d obéissent à la loi normale $\mathcal{N}(0,1)$.*

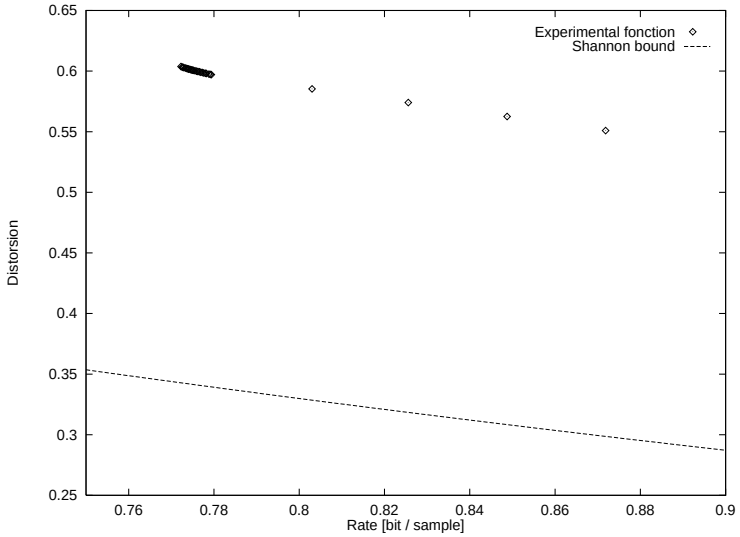

FIG. 42 - *Emboîtage de réseaux D_4, découpage de l'arbre, source synthétique : courbe débit vs. distorsion obtenue par quantification d'une source vectorielle dont les échantillons i.i.d obéissent à la loi normale $\mathcal{N}(0,1)$.*

comparer, nous avons inscrit la courbe obtenue avec l'algorithme de LBG [27] pour lequel la dimension vectorielle est fixée égale à 4 et la recherche au sein du dictionnaire est exhaustive. Les quantificateurs par emboîtage de réseaux mettant en jeu les plus hautes dimensions vectorielles offrent de meilleurs résultats (un PSNR plus élevé à bas débit) et permettent d'obtenir de bas débits. Pour une dimension vectorielle donnée, la quantification vectorielle par emboîtage de réseaux optimaux vis à vis de cette dimension [14], offre la meilleure performance, ainsi la QVEHRRP avec D_4 apparaît supérieure à celle avec $\mathbf{Z}^4$.

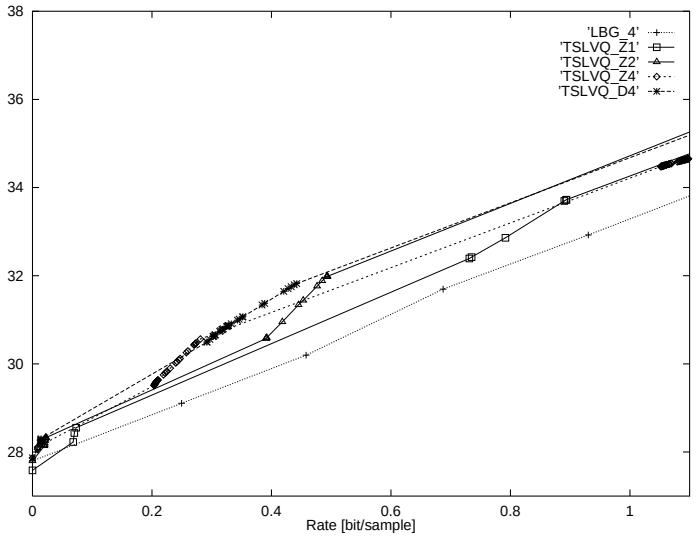

FIG. 43 - *Emboîtage de réseaux, découpage de l'arbre, source réelle: courbes débit vs. PSNR obtenues par quantification de la source réelle.*

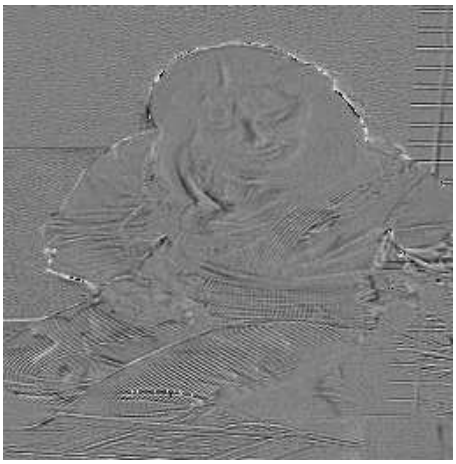

FIG. 44 - *Image extraite de la séquence d'images différentielles à quantifier: une image d'erreurs de prédiction de compensation du mouvement .*

11 Conclusion

Dans ce rapport nous avons décrit la quantification vectorielle, rappelé la supériorité théorique de cette approche et étudié les principales méthodes déjà utilisées. L'analyse du contexte de l'étude (la quantification de séquences d'images différentielles ou hybrides) nous a conduit à envisager un nouveau schéma de quantification : la quantification vectorielle par emboîtage d'une hiérarchie de réseaux réguliers de points. Nous avons donc décrit la conception complète d'un tel quantificateur où coopèrent des techniques de quantification vectorielle algébrique et d'édification par apprentissage, suivant un critère débit-distorsion, d'un dictionnaire arborescent non-équilibré. Des premiers résultats expérimentaux ont été analysés, ils illustrent le bon comportement de notre algorithme (qualité et rapidité de la quantification).

Références

[1] M. Antonini. – *Transformée en ondelettes et compression numérique des images.* – PhD thesis, Université de Nice-Sophia Antipolis, September 1991.

[2] M. Antononi, M. Barlaud, P. Mathieu, and I. Daubechies. – Image coding using wavelet transform. – *IEEE Transactions on Image Processing*, 2, April 1992.

[3] M. Barlaud, P. Solé, T. Gaidon, M. Antonini, and P. Mathieu. – Pyramidal lattice vector quantization for multiscale image coding. – *IEEE Transactions on Image Processing*, 3(4):367–381, July 1994.

[4] A.R. Barron. – The strong ergodic theorem for densities : generalized shannon-mcmillan-breiman theorem. – *Ann. Probab.*, 13:1292–1303, 1985.

[5] A. Benazza. – *Quantification Vectorielle en Codage d'Images.* – PhD thesis, Université de Paris XI Orsay, 1992.

[6] T. Berger. – *Rate Distortion Theory.* – PRENTICE-HALL, INC., Englewood Cliffs, New Jersey, 1971.

[7] E.R. Berlekamp (ed.). – *Key Papers in the Development of Coding Theory.* – IEEE Press, New York, 1974.

[8] R.E. Blahut. – Computation of channel capacity and rate distortion functions. – *IEEE Transactions on Information Theory*, pages 460–473, July 1972.

[9] L. Breiman, J.H. Friedman, R.A Olshen, and C.J. Stone. – *Classification and regression Trees.* – The Wadsworth Statistics/Probality Series. Wadsworth, Belmont, California, 1984.

[10] A. Buzo, A.H. Gray Jr., and J.D. Markel. – Speech coding based upon vector quantization. – In *IEEE Transactions on Acoust. Speech Signal Processing*, volume ASSP-28, pages 562–574, October 1980.

[11] J.H. Conway and Sloane N.J.A. – Fast quantizing and decoding algorithms for lattice quantizers and codes. – *IEEE Transactions on Information Theory*, IT-28(2):227–232, March 1982.

[12] J.H. Conway and Sloane N.J.A. – A fast encoding method for lattice codes and quantizers. – *IEEE Transactions on Information Theory*, IT-29(6):820–824, November 1983.

[13] J.H. Conway and Sloane N.J.A. – A lower bound on the average error of vectors quantizers. – *IEEE Transactions on Information Theory*, IT-31(1):106–109, January 1985.

[14] J.H. Conway and Sloane N.J.A. – *Sphere Packings, Lattices and Groups, 2nd edition.* – A series of Comprehensive Studies in Mathematics. Springer-Verlag, New York, 1993.

[15] T. Eriksson. – Mutistage vector quantization with dynamic bit allocation. – In *Proc. of European Signal Processing Conference*, volume 1. Edinburgh, Scotland, 1994.

[16] T.R. Fisher. – A pyramid vector quantizer. – *IEEE Transactions on Information Theory*, IT-32(4):568–583, July 1986.

[17] A. Gersho. – Asymptotically optimal block quantization. – *IEEE Transactions on Information Theory*, IT-25(4):373–380, July 1979.

[18] A. Gersho and R.M. Gray. – *Vector Quantization and Signal Compression.* – Kluwer Academic Publishers, Boston, 1992.

[19] M. Goldberg, P.R. Boucher, and S. Schlien. – Image compression using adaptative vector quantization. – *IEEE Transactions on Communications*, 34(2):180–187, February 1986.

[20] R.M. Gray. – Vector quantization. – *IEEE ASSP Magazine*, pages 4–29, April 1984.

[21] A.H. Gray Jr. and J.D. Markel. – Distance measures for speech processing. – *IEEE Transactions on Acoust. Speech Signal Processing*, 24(5), October 1976.

[22] D.A. Huffman. – A method for the construction of minimum-redundancy codes. – In *Proc. of the IRE*, pages 1098–1101, 1952.

[23] N.S. Jayant and Noll P. – *DIGITAL CODING OF WAVEFORMS - Principles and Applications to Speech and Video.* – Prentice-hall signal processing series. PRENTICE-HALL, INC., Englewood Cliffs, New Jersey, 1984.

[24] A.H. Juang, B.-H. ang Gray Jr. – Multiple stage vector quantization for speech coding. – In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 597–600. Paris, April 1982.

[25] A.N. Kolmogorov. – On the shannon theory of information transmission in the case of continous signals. – *IRE Transactions on Information Theory*, pages 102–103, 1956.

[26] C. Lamblin. – *Quantification Vectorielle Algibrique Sphirique par le riseau de Barnes-Wall: Application au Codage de la Parole.* – PhD thesis, University of Sherbrooke, Quebec, Canada, March 1988.

[27] Y. Linde, A. Buzo, and R.M. Gray. – An algorithm for vector quantizer design. – *IEEE Transactions on Communications*, 28:84–95, 1980.

[28] S.P. Lloyd. – Least squares quantization in pcm. – *IEEE Transactions on Information Theory*, IT-28(2):129–137, March 1982.

[29] J. MacQueen. – Some methods for classification and analysis of multivariate observations. – In *Proc. of the Fifth Berkeley Symposium on Math. Stat. and Prob.*, volume 1, pages 281–296, 1967.

[30] J. Max. – Quantizing for minimum distorsion. – *IEEE Transactions on Information Theory*, IT-6:7–12, March 1960.

[31] P. Monet and C. Labit. – Codebook replenishment in classified pruned tree-structured vector quantization of image sequences. – In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, pages 2285–2288, 1990.

[32] A.N. Netravali and B.G. Haskell. – *Digital pictures: Representation and Compression.* – Plenum Press, New York, 1988.

[33] E.A. Riskin and R.M Gray. – A greedy tree growing algorithm for the design of variable rate vector quantizers. – *IEEE Transactions on Signal Processing*, 39(11):2500–2507, November 1991.

[34] C.E. Shannon. – A mathematical theory of communication. – *Bell System Technical Journal*, pages 379–423, 623–656, 1948.

[35] C.E. Shannon. – Coding theorems for a discrete source with a fidelity criterion. – *IRE National Convention Record, part 4*, pages 142–163, 1959.

[36] C.E. Slepian (ed.). – *Key Papers in the Development of Information Theory.* – IEEE Press, New York, 1973.

[37] A.J. Viterbi and J.K. Omura. – *Principles of Digital Communication and Coding.* – McGraw-Hill, New York, 1979.

[38] P. Zador. – Asymptotic quantization error of continuous signals and their quantization dimension. – *IEEE Transactions on Information Theory*, IT-28:139–149, March 1982.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399