

Ecole Nationale Polytechnique
DER de Génie Electrique & Informatique

THESE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Présentée par

Mohand Saïd DJOUADI

Pour l'obtention du titre de

MAGISTER EN ELECTRONIQUE
Option : Acquisition et traitement de l'information

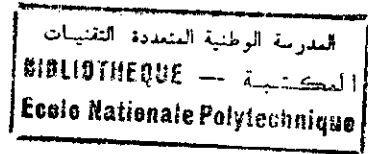
Thème

***Design d'un Quantificateur Vectoriel Hybride
Pour le Codage d'Images***

Soutenue le 20 juin 1998 devant le jury :

Mme GUERTI M.	Maître de conférence (ENP)	Président
Mr BERKANI D.	Maître de conférence (ENP)	Rapporteur
Mr BELOUCHRANI A.	Docteur (ENP)	Examineur
Mr SAADAOU L.	Chargé de cours (ENP)	Examineur
Mme HAMAMI L.	Chargé de cours (ENP)	Examineur
Mlle MERAZKA F.	Magister (ENP)	Invitée

Remerciements



Je tiens à remercier vivement mon directeur de thèse Monsieur Daoud BERKANI pour l'honneur qu'il ma fait en acceptant de diriger cette thèse.

Je tiens également à remercier Madame GUERTI de l'honneur qu'elle me fait en acceptant de présider le jury de cette thèse, ainsi que Madame HAMAMI , Monsieur SAADAoui et Monsieur BELOUCHRANI d'avoir accepter de faire partie de ce jury.

Je remercie également tous les cadres de la DER Génie électrique et informatique, en particulier Monsieur BOUCHERIT pour leurs témoignages de sympathie et d'amitié qui ont été pour moi une aide précieuse.

Je remercie spécialement Monsieur Mohand Amokrane BOUSLIMANI, pour ses encouragements et son soutien indéfectible. Qu'il trouve ici mon témoignage de ma profonde gratitude.

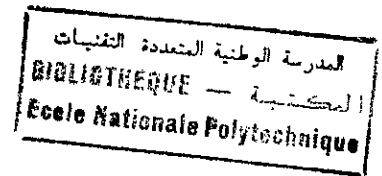
Je ne saurais oublier de remercier Monsieur Hacene REZINE, Monsieur Saleh KELLALI et Monsieur Ahmed DAMOU pour leur soutien et leurs encouragements sans limite. Qu'ils me soit permis de leur témoigner ma profonde gratitude.

الطريقة الإحصائية للتكميم الشعاعي، معاجم أشعة تسمح لتدفق محدد، ترميز مثالي. و لكن هذه المعاجم تخلو من أي بنية، و تتطلب مجالات ضخمة من الذاكرة لتخزينها و وقت كبير لعملية الترميز. إن الشبكات النقطية تسمح بتوليد عدد كبير من النقط من خلال عدد ضئيل من الأشعة، الشيء الذي يمكن أن يؤدي إلى حل مشكلة الطريقة الإحصائية. و لكن للشبكات النقطية مشاكلها الخاصة، بحيث لا تستعمل إلا في حالة المنابع ذات التوزيع المنتظم. من أجل تجنب عوائق الطريقتين، لابد من اقتراح طريقة أخرى. هذه الدراسة مبدؤها إنشاء مكتم حديد يسمى هجين بالتنسيق بين الطريقتين. أولا نعين بغض النقاط الإحصائية باستعمال خوارزمية LBG، ثم كل نقطة توجهنا نحو شبكة Gosset. من أجل رفع النوعية البصرية للصورة المتحصل عليها، نطبق على المكتم المهجن عملية نشر التشويه بالطرح. يمكن القول بأن النتائج المحصل عليها جد مرضية، مثلما تبينته الصور المقدمة.

Abstract Statistic approach of vector quantization uses code-books of rounded vectors which allows quasi-optimal coding for a given rate. Therefore these codebooks have no structure and require big memory size for their storage and a great deal of time for coding. Regular lattices of points gives the possibility of generating a great number of points from a short number of vectors. This can permit us to solve the problems of the statistic approach. However the lattices presents their own problem; effectively they are only applied for uniform distribution sources. In order to avoid these two kinds of problems, another approach is envisaged. This study consists of designing a new quantizer, which we tell hybrid, by combining the two techniques. Using the LBG algorithm performs first few statistic vectors of the source, then each of them leads to a Grosset lattice. In order to increase the perceptual quality of the reconstructed image, a subtractive dithering technique is applied to the hybrid quantizer. Like it could be seen in the presented images, the obtained results are very satisfying.

Résumé La quantification vectorielle par l'approche statistique, utilise des dictionnaires de vecteurs arrondis permettant à un débit donné un codage quasi-optimal. Cependant ces dictionnaires ne présentent aucune structure et nécessitent des espaces mémoires de tailles importantes pour leur stockage et un temps élevé pour le codage. Les réseaux réguliers de points donnent la possibilité de génération d'un grand nombre de points à partir d'un nombre réduit de vecteurs. Ceci peut nous permettre de résoudre le problème de l'approche statistique. Cependant les réseaux de points présentent leur propre problème. En effet, ils sont applicables uniquement dans le cas des sources à distribution uniforme. Pour éviter ces deux types de problèmes, une autre approche est envisagée. Cette étude consiste en le design d'un nouveau quantificateur nommé hybride, par la combinaison des deux techniques. En premier lieu, on détermine quelques points statistiques par l'algorithme du LBG, ensuite chacun de ces points servira d'aiguille vers un réseau de Gosset. Pour améliorer la qualité perceptuelle de l'image reconstruite, on applique au quantificateur hybride la technique de diffusion d'erreur par soustraction. Comme peuvent le montrer les images présentées, les résultats obtenus sont très satisfaisants.

Table des Matières



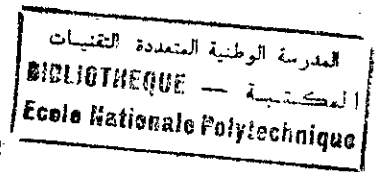
Introduction Générale	3
Partie I. Théorie de l'Information et Quantification	6
1 Eléments de la théorie de l'information	
1.1. Notion d'entropie.....	7
1.1.1. <i>Entropie d'un dictionnaire</i>	7
1.2. Fonction débit distorsion.....	8
1.3. Source à amplitude discrète.....	8
1.4. Source à amplitude continue.....	9
1.4.1. <i>Distorsion</i>	9
1.4.2. <i>Information mutuelle</i>	10
1.5. Source à amplitude discrète et sans mémoire.....	12
1.6. Source à amplitude discrète avec mémoire.....	13
1.7. Codage sans perte d'une source à amplitude discrète.....	14
1.8. Source à amplitude continue et sans mémoire.....	14
1.9. Source à amplitude continue avec mémoire.....	15
2 La Quantification	
2.1. Quantification Scalare (SQ).....	16
2.1.1. <i>Quantification Scalare optimale</i>	18
2.2. Quantification Vectorielle (VQ).....	18
2.2.1. <i>QV optimal (approche intuitive)</i>	22
2.2.2. <i>QV optimal</i>	22
2.2.3. <i>QV appliquée à la compression</i>	23
2.2.4. <i>QV appliquée au codage</i>	23
2.2.5. <i>Design d'un quantificateur vectoriel statistique</i>	25
2.2.6. <i>Organigramme du LBG</i>	27
2.2.7. <i>Complexité de construction du dictionnaire</i>	28
2.2.8. <i>Complexité de la recherche au sein du dictionnaire</i>	28
2.2.9. <i>Design d'un quantificateur vectoriel algébrique</i>	29

2.2.10. Quantification vectorielle sphérique.....	30
2.2.11. Avantage de la quantification vectorielle sphérique.....	32

Partie II. Réseaux réguliers de points, Approche Hybride et Diffusion par soustraction

3 Les Réseaux réguliers de points	
3.1. Définition d'un réseau.....	35
3.2. Réseaux et codes correcteurs d'erreurs.....	35
3.3. Réseaux sphériques.....	37
3.4. Problème d'empilement des sphères dans un espace.....	39
3.5. Problème de recouvrement d'espace par des sphères.....	41
3.6. Problème du nombre de contacts (« kissing number »).....	42
3.7. Introduction des séries θ	43
3.8. Les réseaux réguliers les plus importants.....	44
3.9. Le réseau de Gosset E_8	46
3.9.1. Quantification rapide avec un réseau de Gosset E_8	47
3.9.2. Complexité de la quantification dans E_8	47
3.9.3. Quantification sphérique avec le réseau de Gosset E_8	47
3.9.4. Algorithme de quantification pour une sphère de E_8	49
3.9.4.1. Algorithme de codage avec une sphère de E_8	51
3.9.4.2. Algorithme de décodage.....	51
4 Approche Hybride et technique de Diffusion	
4.1. Approche hybride de quantification (Statistique / Algébrique).....	52
4.1.1. Algorithme de codage avec le quantificateur hybride.....	53
4.1.2. Algorithme de décodage avec le quantificateur hybride.....	54
4.1.3. Résultats expérimentaux.....	54
4.2. Quantification vectorielle avec dithering (diffusion d'erreurs).....	56
4.2.1. Diffusion par soustraction.....	59
4.2.2. Résultats expérimentaux.....	60
Conclusion générale.....	62
Références bibliographiques.....	64
Annexe.....	66

Introduction Générale



De nos jours, la compression d'images s'avère être nécessaire et incontournable au vu de ses diverses applications, notamment au niveau des transmissions TV, vidéo conférences, images graphiques, images satellitaires, au niveau des avions de reconnaissance, etc.

Ceci d'une part, d'autre part la compression d'images trouve son intérêt dans l'élaboration de codes efficaces pour réduire la capacité mémoire nécessaire aux stockage d'images tout en maintenant une certaine fidélité, chose nécessaire notamment pour les images multispectrales et les dessins etc.

Le but fondamental de la compression d'images, est la réduction du débit de transmission et de la capacité mémoire nécessaire à leurs stockage tout en gardant une fidélité acceptable ou une qualité appréciable de ces images après restitution.

Plusieurs techniques de compression satisfaisante ont alors vu le jour comme la DPCM, (Differential Pulse Code Modulation), Codage par Transformée, Codage Hybride , et des techniques adaptatives de celles ci [34].

Ces techniques exploitent souvent l'aspect psychovisuel et stochastique (redondances) de l'image pour réduire le débit. Néanmoins l'un des inconvénients de ces méthodes conventionnelles, et que le quantificateur utilisé est **Scalaire**, c'est à dire, on quantifie chaque échantillon de la forme d'onde ou chaque pixel de l'image individuellement [35,38]. Le problème majeur de toutes ces méthodes, est que les échantillons utilisés sont toujours plus ou moins corrélés entre eux, de ce fait elles restent toujours sous-optimales.

Selon la théorie de SHANNON-FANO [38] sur le rapport **débit / distorsion** on obtient toujours de meilleures performances en codant des vecteurs à la place des scalaires, résultat valable même pour le cas de sources sans-mémoire.

Pour faire la conception d'un quantificateur, deux approches sont généralement utilisées :

1. Approche statistique
2. Approche algébrique

La première approche exploite la distribution de la source de signaux ou de données à coder, elle trouve alors son intérêt dans le cas où la distribution de cette dernière est connue.

La seconde approche quant à elle, ignore la nature de la distribution de la source à coder, d'où la non nécessité de la connaissance de cette dernière.

Néanmoins chaque approche présente ses propres inconvénients :

Pour le cas de l'approche statistique, souvent la nature de la distribution de la source à coder est inconnue, elle peut être donc quelconque. Pour lever cet obstacle on cherche des vecteurs statistiques en utilisant des séquences d'apprentissage [3] de la source à coder. L'inconvénient de cette procédure, est qu'il est nécessaire de stocker en mémoire tout ces vecteurs dont le nombre doit être souvent important pour garantir une certaine qualité du signal à la restitution. Cette façon d'opérer induit en réalité un autre problème autre que celui de la capacité mémoire; en effet, ces vecteurs statistiques n'ont aucune structure ni aucune régularité, une recherche exhaustive s'avère alors obligatoire lors de l'opération de codage de tout vecteur d'entrée, ce qui entraîne un temps de calcul et de recherche élevé.

Pour le cas de l'approche algébrique [35,36,38], même si elle ne nécessite pas le stockage d'un grand nombre de vecteurs (uniquement les vecteurs formant la base génératrice du réseau algébrique sont stockés). Elle ne s'applique que pour le cas des sources à distribution uniforme, et ce vu l'uniformité et la régularité des points formant tous réseau algébrique. Comme dans le cas réel, les sources informationnelles ne sont généralement jamais uniformes, les résultats que peut donner un codeur à base d'un quantificateur algébrique ne peuvent donc être que de moindre qualité.

Vu ces différents problèmes et inconvénients des deux approches sus citées, les chercheurs dans le domaine des codeurs essayent de trouver une autre approche alternative aux deux précédentes.

Le travail présenté au niveau de cette thèse rentre dans cette optique; en effet, l'apport de cette thèse consiste, en l'élaboration et en la proposition d'une nouvelle approche de design de quantificateurs vectoriels, en essayant de combiner les approches statistiques et algébriques en une approche médiane profitant des avantages de chacune d'entre elles et évitant leurs inconvénients respectifs, cette nouvelle approche porte le nom d'**approche hybride**.

Cette thèse consistera donc en la description d'un nouveau quantificateur vectoriel destiné à prendre place au niveau d'une chaîne de codage pour la compression de données ou de signaux informationnels. L'innovation réside dans la mise en œuvre d'une technique *hybride* Algébrique / Statistique, en utilisant des réseaux algébriques dits de **Gosset** comme réseaux réguliers de points dont les centroïdes sont des points statistiques de la source à coder déterminés par apprentissage grâce à l'algorithme du **LBG**.

Ce travail se divise en deux grandes parties.

La première partie sera consacrée à des généralités et à des définitions sur la théorie de l'information et de la quantification scalaire et vectorielle.

La seconde partie sera consacrée à l'introduction et à la description de la nouvelle approche et des outils mathématiques utilisés pour sa mise en œuvre; les résultats obtenus sur des images en utilisant plusieurs variantes des codeurs selon le nombre de points statistique utilisés seront présentés. Une autre technique dite de dithering ou de diffusion d'erreurs [36] sera également présentée et appliquée à un codeur algébrique à base d'un réseau de Gosset, seront également présentés les résultats obtenus sur des images standards de personnes. Enfin je terminerai par des conclusions situant les résultats obtenus et donnant les perspectives d'amélioration des performances de l'approche hybride.

PARTIE I

Théorie de l'Information et Quantification

Chapitre 1

Éléments de la théorie de l'information

1.1. Notion d'entropie

Soit une source à temps discret et ergodique $\{x(n)\}$, $n = 0, \pm 1, \pm 2, \dots$

Le flux de **symboles** $x(n)$ forme une séquence aléatoire $\{X(n)\}$, $n = 0, \pm 1, \pm 2, \dots$ dont les réalisations x_k appartiennent à l'alphabet de la source $A_k = \{x_k \mid k = 1, 2, \dots, K\}$ (1.1)

Si K est fini la source est dite à **amplitude discrète**, si K est infini la source est dite à **amplitude continue**.

La source est **sans mémoire** si les échantillons successifs sont statistiquement indépendants.

1.1.1. Entropie d'un dictionnaire

Soit un dictionnaire de longueur L et de taille k , à chaque vecteur représentant est associé un index i parmi les L index du dictionnaire. Chaque index i se présente comme un mot de code binaire c_i de longueur b_i bits.

L'entropie de ce dictionnaire est donnée par la relation suivante :

$$H(D) = \frac{1}{k} \cdot \sum_{i=1}^L P_i \cdot (-\log_2 P_i) \quad (1.2)$$

Avec :

- P_i : Probabilité de sélectionner le vecteur représentant y_i du dictionnaire D ,
- $(-\log_2 P_i)$: l'appréciation numérique sur l'incertitude de sélectionner y_i , l'unité d'information étant le bit.

L'entropie du dictionnaire précise 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 remarquera que l'entropie est toujours inférieure à la longueur moyenne des index :

$$H(D) \leq \sum_{i=1}^L P_i \cdot b_i \quad (1.3)$$

1.2. Fonction débit distorsion

En pratique la source à coder est souvent continue. On souhaite alors que le système codeur-décodeur assure la transmission de l'information avec un débit R (en bits par 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 stipule qu'il existe une **fonction débit-distorsion** $D(R)$ qui fournit une borne aux performances du système de codage [38,41]. Cette fonction $D(R)$ indique la distorsion minimale théorique pour le codeur avec le débit R :

$$D(R) \leq D'(R) \quad (1.4)$$

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 :

$$R'(D) \geq R(D) \quad (1.5)$$

1.3. Source à amplitude discrète

Un codage entropique est réalisable.

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

$$\text{Min}\{R\} = R(0) = H(X) \quad (1.6)$$

$R'(0) = h(X)$ est atteint à l'aide d'un code entropique. Si la source est avec mémoire le calcul du code est plus simple.

$D(R)$ est une fonction monotone décroissante avec R [38,41]. 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 zéros, 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.

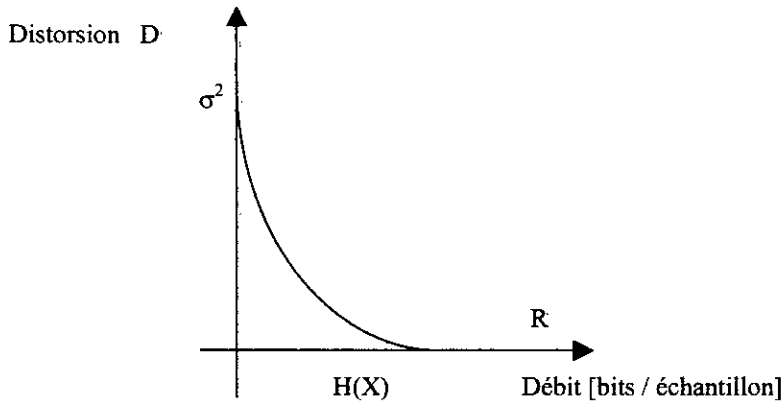


Fig 1.1 : Exemple de la courbe débit-distorsion d'une source à amplitude discrète

1.4. Source à amplitude continue

1.4.1. Distorsion

Soit $\{X(n)\}$, $n = 0, \pm 1, \pm 2, \dots$ 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 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 vecteur $\mathbf{x}$ 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:

$$E[d_N(X, Y)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} d_N(x, y) \cdot p(x, y) dx \cdot dy \quad (1.7)$$

avec:

$$d_N(x, y) = \frac{1}{N} \cdot \sum_{k=1}^N (x(k) - y(k))^2 \quad (1.8)$$

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

- La statistique de la source $p(x)$,
- La probabilité des transitions $p(y / x)$.

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

1.4.2. 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'informations entre le codeur et le décodeur [38,41], elle apprécie la quantité moyenne d'informations qu'implique la "réception" des vecteurs par rapport à ceux émis.

$$I(X, Y) = \lim_{N \rightarrow \infty} I_N(X, Y) \quad (1.9)$$

Avec:

$$\begin{aligned} I_N(X, Y) &= \frac{1}{N} \cdot \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p(x, y) \cdot \log_2(p(y|x) \cdot p(x)) dx \cdot dy \\ &= \frac{1}{N} \cdot \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p(x, y) \cdot \log_2(p(x|y) \cdot p(y)) dx \cdot dy \end{aligned} \quad (1.10)$$

$I(X, Y)$ dépend donc également de la statistique de la source et de celle des transitions.

L'adjectif **mutuel** vient de l'égalité :

$$I(X, Y) = I(Y, X) \quad (1.11)$$

L'information mutuelle calcule le débit minimal R nécessaire pour avoir une fidélité de reconstruction D . En effet, pour une densité $p(x)$ donnée, considérons:

$$S = \{ p(y / x) : I_N(X, Y) \leq R \} \quad (1.12)$$

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 $E[d_N(X,Y)]$. Nous recherchons alors celui assurant le minimum de distorsion:

$$D_N(R) = \min_{p(y|x) \in S} E[d_N[X, Y]] \quad (1.13)$$

Le codeur réalisant $D_N(R)$ est optimal, il assure la distorsion moyenne minimale sous une contrainte de débit inférieur à R . On remarquera que $D_N(R)$ 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:

$$D(R) = \lim_{N \rightarrow \infty} D_N(R) \quad (1.14)$$

La fonction $D(R)$ fournit, pour un débit R donné, 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 X et les sorties Y du système sont des vecteurs. Il faut également savoir que dès des tailles appréciables, 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 source à amplitude continue $R(0) = +\infty$ car, pour $D = 0$:

$$\bullet \quad P(y/x) = \begin{cases} 1 & \text{si } y = x \\ 0 & \text{ailleurs} \end{cases} \quad (1.15)$$

$$\bullet \quad H(X) = H(Y) = I(X,Y) = +\infty \quad (1.16)$$

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

$$\bullet \quad D > 0 \quad (1.17)$$

$$\bullet \quad I(X,Y) < +\infty \quad (1.18)$$

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

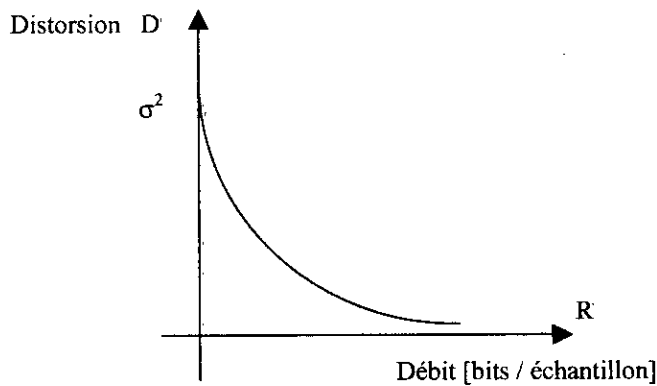


Fig 1.2 : Courbe débit-distorsion pour une source à amplitude continue.

1.5. Source à amplitude discrète et sans mémoire

Soit P_k la probabilité d'occurrence du symbole x_k : $P_k = P_r \{ X(n) = x_k \}$. (1.19)

Une appréciation numérique de l'information reçue est donnée par :

$$I(x_k) = -\log_2 P_k \quad (\text{en bits}) \quad (1.20)$$

La quantité d'information moyenne reçue ou **entropie** est alors (en bit / échantillon) :

$$H(X) = E(I(X)) = -\sum_{k=1}^K P_k \cdot \log P_k \quad (1.21)$$

On a alors :

$$0 \leq H(X) \leq \log_2 K \quad (1.22)$$

- 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 :

$$R(X) = \log_2 K - H(X) \quad (1.23)$$

1.6. Source à amplitude discrète avec mémoire

Dans ce cas, Il existe des dépendances statistiques entre les échantillons successifs de la source [41,42].

Soit $\mathbf{x} = \mathbf{x}(n) = (x(n), x(n+1), \dots, x(n+N-1))^T$

Soit l'entropie des vecteurs aléatoires (en bits / échantillon) :

$$H_N(\mathbf{X}) = \frac{1}{N} \cdot E(-\log_2 P(\mathbf{X})) = -\frac{1}{N} \cdot \sum_x \sum_x \dots \sum_x P(x) \cdot \log_2 P(x) \quad (1.24)$$

avec :

$$H_N(\mathbf{X}) = \lim_{N \rightarrow \infty} H_N(\mathbf{X}) \quad (1.25)$$

On considère l'entropie conditionnelle d'un symbole à l'instant n étant donnés les $(N-1)$ symboles précédents :

$$H(X(n) / X(n-1), X(n-2), \dots, X(n-N+1))$$

On pourra démontrer que :

$$H(X(n) / (X(n-1), X(n-2), \dots, X(n-N+1))) \leq H_N(X) \quad (1.26)$$

et que :

$$\lim_{N \rightarrow \infty} H(X(n) / (X(n-1), X(n-2), \dots, X(n-N+1))) = H(X) \quad (1.27)$$

Les deux fonctions $H_N(X)$ et $H(X(n) / (X(n-1), X(n-2), \dots, X(n-N+1)))$ sont décroissantes avec N .

On a également pour deux sources ayant le même alphabet et la même probabilité pour les symboles, on pourra prouver que :

$$(H(X) / \text{source avec mémoire}) \leq (H(X) / \text{source sans mémoire}) \quad (1.25)$$

1.7. Codage sans perte d'une source à amplitude discrète

En théorie un codage sans perte d'informations avec débit binaire proche de l'entropie est possible pour une source à amplitude discrète [38,41,42]. Ce codage est dit **entropique** ou *run lenght coding* (exemple : codage de Huffman).

Au niveau de ce code, les statistiques du signal sont exploitées en affectant les mots code les plus courts aux représentants les plus fréquents.

Ce codage est efficace si les symboles de la source ont des probabilités pouvant être approchées par des puissances négatives de deux .

1.8. Source à amplitude continue et sans mémoire

Soit $\{ X(n) \}$ une source stationnaire centrée et sans mémoire.

Soient $p_x(x)$ sa fonction de densité de probabilité, $\sigma_x^2 = E(X^2(n))$ sa variance et

$R_{xx} = \sigma_x^2 \cdot \delta(k)$ sa fonction d'autocorrélation.

Un tel signal a une **entropie absolue** infinie

En effet : $P(x_k) = 0$ d'où $H(X) = +\infty$

C'est pour cette raison qu'on introduit l'**entropie différentielle** (qui peut être positive, négative ou nulle en fonction de l'amplitude de la source) :

$$h(X) = E(-\log_2 p_x(X)) = - \int_{-\infty}^{+\infty} p_x(x) \cdot \log_2 p_x(x) dx \quad (1.29)$$

On démontre que l'entropie différentielle est maximale si la source suit la loi Gaussienne [38,41,42]

$f(0, \sigma_x^2)$, dans ce cas on aura :

$$p_x(x) = \frac{1}{2 \cdot \pi \cdot \sigma_x^2} \cdot \exp\left(\frac{-x^2}{2 \cdot \sigma_x^2}\right) \quad (1.30)$$

On aura donc :

$$h(X)_G = \log_2 \sqrt{2 \cdot \pi \cdot e \cdot \sigma_x^2} \quad (1.31)$$

On définit aussi la puissance entropique qui traduit la répartition de l'information par unité de variance du signal :

$$Q = \frac{1}{2 \cdot \pi \cdot e} \cdot 2^{2 \cdot h(X)} \quad (1.32)$$

Dans le cas d'une source gaussienne $Q = \sigma_x^2$, pour une source non-gaussienne $Q < \sigma_x^2$.

1.9. Source à amplitude continue avec mémoire

Soit un vecteur $\mathbf{x}$ constitué de N échantillons successifs de la source. Ce vecteur est défini par sa fonction de densité de probabilité $p_{\mathbf{x}}(\mathbf{x})$ et on définit également l'entropie différentielle par :

$$h(\mathbf{X}) = \lim_{N \rightarrow \infty} h_N(\mathbf{X}) \quad (1.33)$$

avec :

$$\begin{aligned} h_N(\mathbf{X}) &= \frac{1}{N} \cdot E(-\log_2 p_{\mathbf{x}}(\mathbf{X})) \\ &= -\frac{1}{N} \cdot \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} p_{\mathbf{x}}(\mathbf{x}) \cdot \log_2 p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (1.34)$$

D'une manière générale :

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

La borne supérieure $(\frac{1}{2} \cdot \log_2(2 \cdot \pi \cdot e \cdot \sigma_x^2))$ est atteinte dans le cas d'une source gaussienne [38], pour le cas des entropies inférieures à cette valeur, elles sont dues aux redondances de la source considérée (sa densité de probabilité n'est pas gaussienne et/ou la source présente une mémoire induisant un spectre de puissance non « plat » ou « blanc »).

La puissance entropique d'une source gaussienne avec mémoire (colorée) est donnée par :

$$Q = \gamma_x^2 \cdot \sigma_x^2 \quad (1.36)$$

où γ_x^2 est la mesure de *platitude du spectre* (spectre plat $\equiv$ spectre blanc), avec :

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

Si $\gamma_x^2 = 1$, on a une source gaussienne sans mémoire,

Si $\gamma_x^2 < 1$, la source en question présente une mémoire, le signal est plus au moins prédictible.

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

Chapitre 2

La Quantification

Généralement les formes d'onde des signaux informationnels sont continues en amplitude et continues dans le temps. Une conversion analogique digitale est alors nécessaire pour produire une représentation discrète de la forme d'onde afin de permettre son traitement. Cette opération de conversion commence par un échantillonnage du signal continu, cependant les échantillons obtenus restent toujours continus en amplitude, une opération de quantification devient alors nécessaire pour limiter le nombre possible d'amplitudes à une valeur finie.

La quantification, est la procédure qui transforme l'amplitude d'un signal donné $x(n)$ à un instant n en une amplitude $y(n)$ donné par un ensemble fini d'amplitudes. Lorsque la quantification est faite sur un seul échantillon elle est dite **Scalaire** (SQ), lorsqu'elle est faite sur un bloc ou plusieurs échantillons elle est dite **Vectorielle** (VQ). Elle peut être également définie comme étant un processus de transformation qui fait correspondre à la variable d'entrée x une valeur approchée y_i choisie dans un ensemble fini de N valeurs prédéterminées.

2.1. Quantification scalaire (SQ)

La quantification scalaire consiste à quantifier séparément chaque échantillon du signal d'entrée [35,38]. Comme l'illustre la figure 2.1, un échantillon x du signal d'entrée est spécifié par l'indice k s'il se trouve dans l'intervalle suivant :

$$I_k : \{ x_k < x < x_{k+1} \}; \quad k=1,2, \dots, N \quad (2.1)$$

x_k et x_{k+1} sont les niveaux de décision ou seuils.

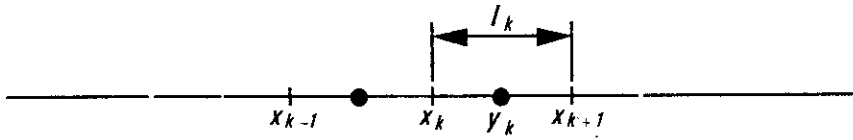


Fig 2.1 : Quantification scalaire

Tous les échantillons dans l'intervalle I_k seront remplacés par une valeur y_k appelée *niveau de reconstruction* ou *représentant*. On peut par conséquent utiliser N niveaux de reconstruction pour représenter les échantillons du signal d'entrée compris entre les valeurs x_1 et x_{N+1} .

Les niveaux de reconstruction peuvent s'écrire:

$$y = y_k \quad \text{si } x \in I_k \quad (2.2)$$

ou s'exprimer par la fonction suivante:

$$y = Q(x) . \quad (2.3)$$

L'expression (2.3), qui montre la relation entre x et y , est appelée *caractéristique de quantification*. La figure 2.2 illustre $y = Q(x)$, typiquement en marche d'escalier [35], pour la quantification scalaire.

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

- le **bruit granulaire** ou de **quantification** qui se produit lorsque la valeur d'entrée x se situe dans l'une des zones $[x_i, x_{i+1}]$, 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 $[x_0, x_L]$. La valeur de reproduction est alors soit y_1 soit y_L , et l'erreur résultante supérieure à un demi pas de quantification.

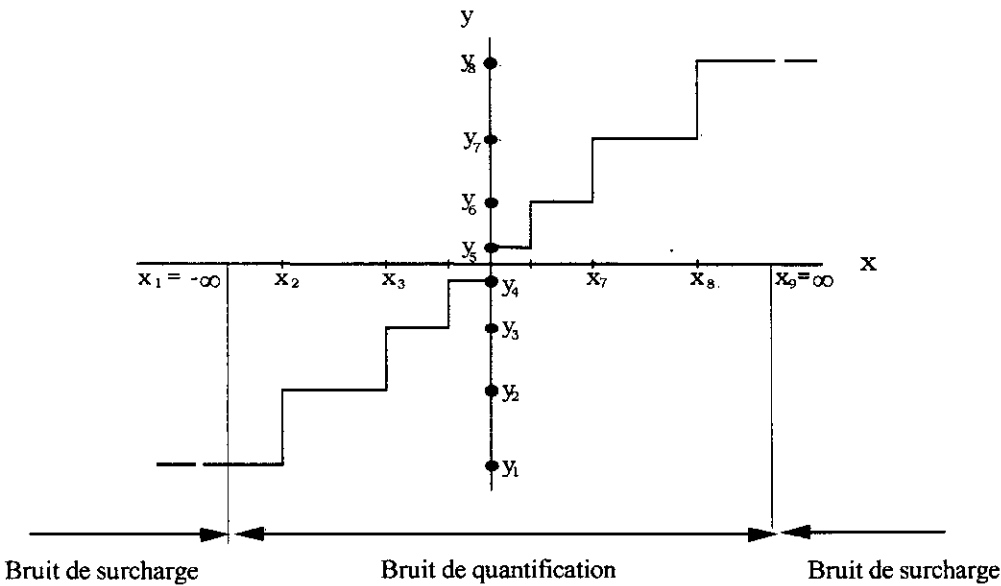


Fig 2.2: Caractéristique de la quantification scalaire.

2.1.1. Quantificateur scalaire optimal

Le quantificateur scalaire optimal est celui qui minimise, pour une source donnée et à 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. On remarquera que la concentration des niveaux de reconstruction est plus importante dans la zone de l'espace où la densité de probabilité des points d'entrée est plus élevée.

La solution à ce problème d'optimisation (elle est unique), est connue sous le nom de *quantificateur de Lloyd-Max* [35,41].

En donnant 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. Malheureusement, ce gain n'atteint pas le maximum donnée par la théorie de l'information.

2.2. Quantification vectorielle (VQ)

La quantification vectorielle consiste à quantifier un jeu d'échantillons (vecteur) au lieu d'un seul échantillon. C'est une application surjective de l'ensemble des vecteurs d'entrée $\mathbf{x}$, généralement de l'espace réel euclidien à k dimensions $\mathbf{R}^k$, dans un sous-

ensemble fini C appelé *dictionnaire* de $\mathbf{R}^k$, [2,3;34]. Le dictionnaire C se compose de N éléments distincts que l'on appelle *mots de code* ou *vecteurs de reproduction*.

$$C = \{ \mathbf{y}_i ; i=1,2, \dots , N \} .$$

(2.4)

Le débit R d'un quantificateur vectoriel utilisant le dictionnaire C est défini par:

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

(2.5)

Le débit représente le nombre de bits par échantillon à coder. $\log_2 N$ bits sont nécessaires pour représenter l'indice d'un vecteur à k dimensions. Le débit en quantification vectorielle, peut être fractionnaire ($R<1$). La figure 2.3 illustre le principe de la quantification vectorielle.

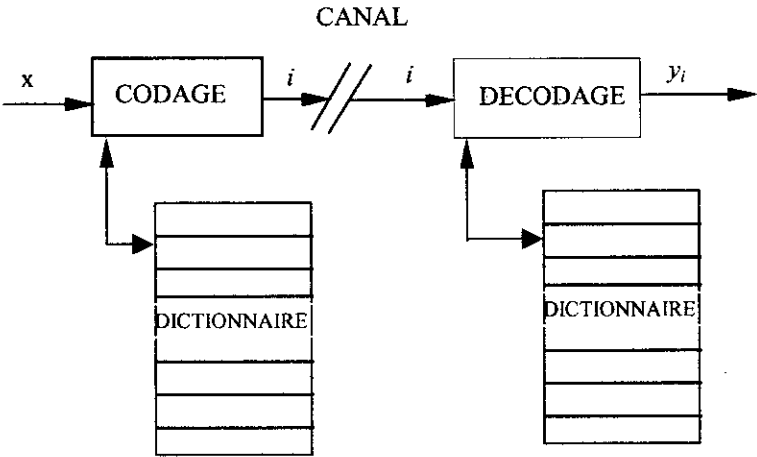


Fig 2.3: Principe de la quantification vectorielle.

Dans le codeur, on associe au vecteur d'entrée $\mathbf{x}$ un mot de code $\mathbf{y}_i$ du dictionnaire C selon le critère du plus proche voisin. Seul l'indice i est transmis au décodeur. On a N indices correspondant aux N mots de code du dictionnaire. Le même dictionnaire est utilisé au décodage. Le mot de code $\mathbf{y}_i$, représentant du vecteur $\mathbf{x}$, est retrouvé à partir de l'indice i reçu. Il est clair que la quantification scalaire est un cas particulier de la quantification vectorielle, pour une dimension k du vecteur égale à 1.

Dans la quantification vectorielle, on exploite le fait que les composantes du vecteur d'entrée ont généralement une corrélation et qu'elles se répartissent principalement dans une région donnée de l'espace $\mathbf{R}^k$. On peut atteindre une plus petite valeur de distorsion

avec le même nombre de vecteurs de reproduction qu'en quantification scalaire en construisant convenablement le dictionnaire. Autrement dit, on obtiendra la même distorsion avec ces deux types de quantification en utilisant un plus petit nombre de vecteurs de reproduction. Le signal d'entrée pourra donc être représenté et transmis à plus bas débit.

Même si le signal d'entrée est non corrélé, ou que les composantes du vecteur sont indépendantes, il y a avantage à utiliser la quantification vectorielle [38]. La figure 2.4 montre la performance en terme d'Erreur Quadratique Moyenne (EQM) d'un quantificateur vectoriel à deux dimensions en pourcentage de la performance d'une paire de quantificateurs scalaires [35]. Les résultats sont donnés pour différentes valeurs de la corrélation ρ et différents débits par dimension.

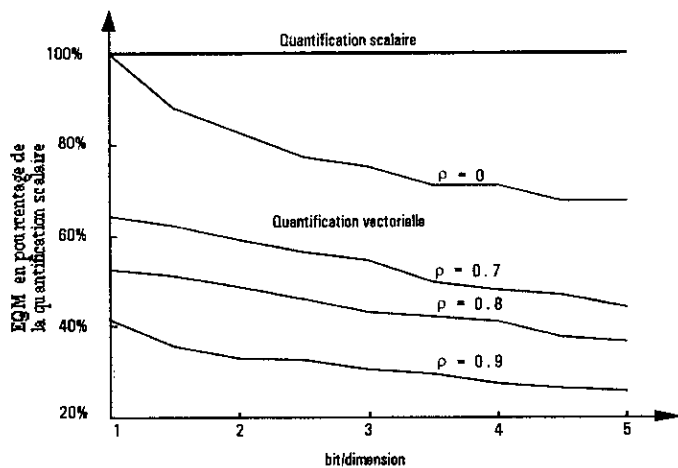


Fig 2.4: Comparaison des performances entre une paire de SQ et un VQ appliqués à un vecteur à deux composantes gaussiennes ayant la corrélation ρ .

Un quantificateur vectoriel est complètement défini par un ensemble $\mathbf{R}_i$ de vecteurs arrondis et une distance $d(\mathbf{x}, \mathbf{y}_i)$ définissant le degré de distorsion qui s'introduit lorsqu'un vecteur d'entrée $\mathbf{x}$ est approximé par un vecteur arrondi $\mathbf{y}_i$.

Pour qu'il soit optimal, il doit satisfaire les deux conditions nécessaires suivantes:

- La distorsion entre le vecteur de reproduction $\mathbf{y}_i$ et tous les vecteurs d'entrée $\mathbf{x}$ représentés par $\mathbf{y}_i$ doit être minimale. Un codeur effectue une partition de $\mathbf{R}^k$ en sous-espaces où tous les vecteurs d'entrée $\mathbf{x}$ représentés par $\mathbf{y}_i$ sont groupés.

Cette partition, appelée *région de Voronoï* ou de *Dirichlet* (voir figure 2.5), est établie selon le critère du plus proche voisin s'exprime de la manière suivante:

$$\mathbf{R}_i \supset \mathbf{x} \in \mathbf{R}^k : d(\mathbf{x}, \mathbf{y}_i) < d(\mathbf{x}, \mathbf{y}_j), j \neq i \}$$
(2.6)

- Pour une partition fixe $\mathbf{R}_i$, le vecteur de reproduction optimal du décodeur doit être le centroïde de la partition

$$\text{cent}(\mathbf{R}_i) = \min_{\mathbf{x}_i}^{-1} E\{d(\mathbf{x}, \mathbf{x}_i) | \mathbf{x} \in \mathbf{R}_i\}$$
(2.7)

où le minimum inverse "min⁻¹" dénote un choix de x_i minimisant la valeur de $E\{d(\mathbf{x}, x_i) | \mathbf{x} \in \mathbf{R}_i\}$.

Le critère de distorsion le plus utilisé est définie par:

$$d(\mathbf{x}, \mathbf{y}_i) = \|\mathbf{x} - \mathbf{y}_i\|^2 = \sum_{l=1}^k (x_l - y_l)^2.$$
(2.8)

Lorsque l'expression (2.8) est utilisée comme critère de distorsion, les deux conditions nécessaires précédentes sont représentées simplement par les relations suivantes.:

$$\mathbf{R}_i \supset \mathbf{x} \in \mathbf{R}^k : \|\mathbf{x} - \mathbf{y}_i\|^2 < \|\mathbf{x} - \mathbf{y}_j\|^2, j \neq i \}$$
(2.9)

et

$$\text{cent}(\mathbf{R}_i) = \frac{1}{n_i} \sum_{j: \mathbf{x}_j \in \mathbf{R}_i} \mathbf{x}_j$$
(2.10)

où n_i est le nombre de vecteurs dans la partition $\mathbf{R}_i$.

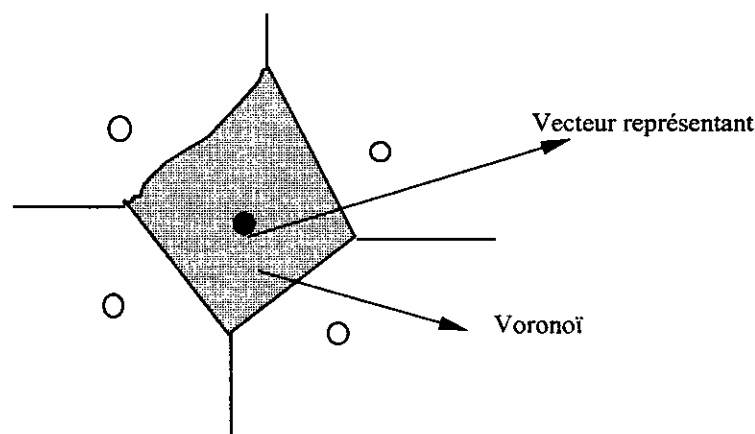


Fig 2.5 : Exemple d'un Voronoï

2.2.1. Quantificateur vectoriel optimal (approche intuitive)

L'efficacité d'un quantificateur se mesure à la qualité de restitution du signal source qui doit être la plus fidèle possible (c.à.d avec une erreur de restitution minimale).

La quantification vectorielle se trouve supérieure par rapport à la quantification scalaire car elle permet :

1. Une meilleure utilisation de la *mémoire* de la source (exploitation des corrélations existantes entre les coordonnées des vecteurs) ;
2. Une facilité pour la répartition dans l'espace des représentants.

Le quantificateur optimal est celui qui répartit les vecteurs de reproduction en tenant compte de la distribution des vecteurs à coder dans l'espace vectoriel de travail [38](en tenant compte de leur densité de probabilité multidimensionnelle).

2.2.2. Quantificateur optimal

Pour une distribution statistique et un débit donnés de la source :

1. le *quantificateur globalement optimal* est celui dont la distorsion moyenne est minimale ;
2. un *quantificateur localement optimal* a un dictionnaire pouvant être perturber sans que la distorsion moyenne n'augmente.

La théorie qui établit la supériorité de la quantification vectorielle sur la quantification scalaire n'est pas constructive [38,41], elle ne nous donne pas la façon de construire le *dictionnaire globalement optimal*. Par contre des propriétés suffisantes sont connues qui nous permettent de construire des *dictionnaires localement optimaux*.

Un quantificateur se décompose en deux éléments:

1. un codeur
2. un décodeur

Le quantificateur (localement) optimal est celui qui satisfait ces conditions :

- le *codeur optimal* (pour un dictionnaire fixé), ce dernier respecte la règle du « plus proche voisin » : $\forall j, si d(x, y_i) < d(x, y_j) \Rightarrow C(x) = y_i$
cette règle détermine la répartition de l'espace $\mathbf{R}^k$ en régions ;
- le *décodeur optimal* (pour une répartition de $\mathbf{R}^k$ donnée), le vecteur y_i doit minimiser la distorsion associée au Voronoï C_i , y_i sera donc le centroïde de cette région : $y_i = cent(C_i)$;
- la probabilité d'avoir un vecteur à coder à la même distance de deux

représentants soit nulle, sinon ce vecteur est affecté à l'un des deux représentants, la partition optimale de l'espace considéré n'existera plus, et donc la condition du décodeur optimal sera impossible.

2.2.3. Quantification vectorielle appliquée à la compression

La compression de données vise à réduire la quantité des éléments binaires nécessaires à la représentation de l'information utile contenue dans le signal à transmettre ou à stocker. Cette réduction peut soit engendrer ou non des pertes d'informations

Principe :

Soit $\{x_n\}$ l'ensemble des échantillons du signal source qu'on veut coder :

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

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

Si $D^{-1} \neq C$, il s'agit d'un code avec perte d'information dit **irréversible**.

Pour réaliser une bonne compression, deux conditions doivent être satisfaites :

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

2.2.4. Quantification vectorielle appliquée au codage

La quantification vectorielle permet la combinaison des applications de codage et de décodage. En effet, considérons l'ensemble **I** des **L** index des vecteurs mots codes du dictionnaire considéré : $I = \{1, 2, \dots, L\}$.

La loi de codage sera alors définie par :

$$C : R^k \longrightarrow I$$

$$x \qquad \qquad C(x) = i$$

Le décodage s'effectuera de la manière suivante :

$$\begin{array}{ccc} D & : & I \longrightarrow D \\ & & i \qquad \qquad D(i) = y_i \end{array}$$

On aura alors obtenu une application : $Q = D \circ C$.

Un quantificateur vectoriel se décompose alors en deux applications qui sont :

- Un codeur
- Un décodeur

Le codeur

Le codeur consiste à rechercher dans un dictionnaire noté D , pour tout vecteur du signal d'entrée x , le code-word y_i qui lui est le plus proche.

Nous introduisons alors la notion de la **métrique** L_p , où la norme d'un vecteur x de dimension k est donnée par :

$$L_p(x) = \|x\|^p = \sum_{j=1}^k |x_j|^p \quad (2.11)$$

la distance entre deux vecteurs x_1 et x_2 sera alors :

$$\begin{aligned} \|x_1 - x_2\|^2 &= d_p(x_1, x_2) \\ &= \sum_{j=1}^k |x_{1j} - x_{2j}|^p \end{aligned} \quad (2.12)$$

La notion de proximité la plus utilisée, est la distance euclidienne :

$$d(x, y_i) = \sum_{j=1}^k (x_j - y_{ij})^2 \quad (2.13)$$

Cette distance particulière $p=2$, mesure la **distorsion** entre les vecteurs x et y_i .

Les régions de Voronoï sont alors données par :

$$C_i = \{x \in R^k \mid Q(x) = y_i, \text{ si } d(x, y_i) \leq d(x, y_j), \forall j \neq i\} \quad (2.14)$$

Chaque région contient un ensemble de vecteurs de R^k et tous les vecteurs d'entrées x qui appartiennent à C_i sont représentés par le même mot code y_i du dictionnaire D considéré.

La compression d'informations se produit à ce niveau car c'est uniquement l'index du code-word y_i qui minimise le critère de distorsion qui sera transmis ou stocké au lieu que se soit un vecteur. Cette application qui perd de l'information, fait que la quantification vectorielle soit une **opération irréversible** [22,32,38], le signal d'entrée ou le signal original ne pourra plus jamais 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 par **bits par dimension**, ou encore par **bits par échantillon** :

$$R = \frac{1}{k} \cdot \log_2 L \tag{2.15}$$

Le décodeur

Le **décodeur** joue le rôle d'un récepteur dédié à la reconstruction du vecteur source, pour cette raison il dispose d'un dictionnaire D, identique à celui utilisé par le codeur pour l'opération de codage, qu'il consulte à chaque fois qu'il reçoit un index pour produire le code-word correspondant; le décodeur réalise l'opération de décompression.

2.2.5. Design d'un quantificateur vectoriel statistique

Cette approche est basée sur un algorithme itératif, *l'algorithme de la K-moyennes* ou de **LBG**, du nom de ses auteurs [3]. A l'aide d'une grande quantité de données d'entrée, on fait l'apprentissage à plusieurs reprises afin de prendre compte des caractéristiques statistiques du signal d'entrée. On utilise donc l'histogramme du signal au lieu de sa distribution (que l'on ne connaît pas) pour la conception du quantificateur vectoriel.

L'algorithme de LBG converge vers un optimum satisfaisant les deux conditions nécessaires précédentes.

Soit $\mathbf{X} = \{ \mathbf{x}_j ; j = 1, 2, \dots, M \}$ une séquence d'apprentissage composée de M vecteurs à k dimensions, $\mathbf{Y}_0 = \{ \mathbf{y}_i ; i = 1, 2, \dots, N \}$ un dictionnaire initial, et $D_0 = \infty$ la distorsion initiale. La conception d'un quantificateur vectoriel statistique se décompose en trois étapes suivantes [3]:

- 1- Selon le critère de distorsion de l'expression (2.8), on associe à chaque vecteur $\mathbf{x}_j$ un vecteur de reproduction $\mathbf{y}_i$ du dictionnaire $\mathbf{Y}_0$. La partition qui minimise la distorsion, $\{ \mathbf{R}_i i = 1, 2, \dots, N \}$ est donc déterminée.

2- La distorsion moyenne globale D est calculée par l'équation suivante:

$$D = \frac{1}{M} \sum_{j=1}^{M'} \min_{\mathbf{y} \in \mathbf{Y}_0} \|\mathbf{x}_j - \mathbf{y}\|^2 \quad (2.16)$$

Si la distorsion D ne varie plus, le calcul est terminé et $\mathbf{Y} = \{\mathbf{y}_i; i=1,2, \dots, N\}$ sont les vecteurs de reproduction optimaux. Sinon, on continue.

3- On calcule les centroïdes à l'aide de la relation (2.10). Les mots de code

$\mathbf{Y} = \{\mathbf{y}_i; i=1,2, \dots, N\}$ sont remplacés par les centroïdes obtenues.

Notons que si dans la partition $\mathbf{R}_i$ le nombre de vecteurs $n_i=0$, on met $\text{cent}(\mathbf{R}_i) = \mathbf{y}_i$, l'ancien mot de code. On répète les étapes 1 et 2.

Le quantificateur obtenu dépend fortement des conditions de départ, celles-ci doivent être choisies soigneusement. On peut utiliser la méthode dite du "*splitting*" pour effectuer la sélection du dictionnaire initial[3,35]. On trouve d'abord un seul mot de code optimal en calculant le centroïde de toute la séquence d'apprentissage. A partir du mot de code optimal, on produit un dictionnaire de taille 2, en perturbant faiblement ce centroïde dans deux directions opposées. L'application de l'algorithme de LBG à ce dictionnaire, permet d'obtenir un meilleur dictionnaire à deux mots de code optimaux qui sont à leur tour divisés en deux pour constituer un dictionnaire de taille 4.

Le procédé est poursuivi jusqu'à l'obtention d'un dictionnaire de la taille voulue. La figure 2.6 illustre le design d'un dictionnaire composé de quatre vecteurs de reproduction à deux dimensions par "*splitting*".

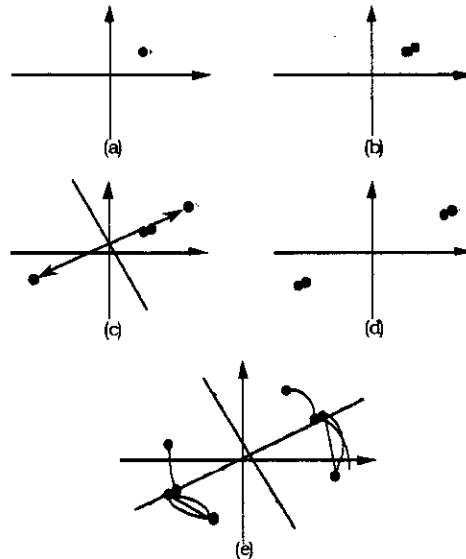


Fig 2.6: Design du dictionnaire initial par "*splitting*".

La technique de quantification vectorielle statistique permet de bien rendre compte de la distribution du signal à coder, mais la taille du dictionnaire et la dimension du vecteur d'entrée sont sévèrement limitées par des contraintes technologiques en matière de stockage et de complexité de calcul.

2.2.6. Organigramme du LBG

Cette méthode, consiste à déduire de chaque vecteur représentant y_i deux nouveaux vecteurs $y_i + \epsilon$ et $y_i - \epsilon$ (ϵ : vecteur de perturbation de faible énergie), avant d'appliquer au nouveau dictionnaire obtenu les itérations dites de **Lloyd**, qui consistent à modifier (à chaque itération) localement le dictionnaire, réduire ou laisser inchangées la distorsion moyenne. L'algorithme converge en un nombre fini d'itérations vers le minimum local le plus proche correspondant au dictionnaire initial.

Le dictionnaire ainsi obtenu est alors le centroïde de la séquence d'apprentissage, ensuite l'algorithme génère une suite de dictionnaires (à chaque boucle le nombre de vecteurs de reproduction se trouve multiplié par 2).

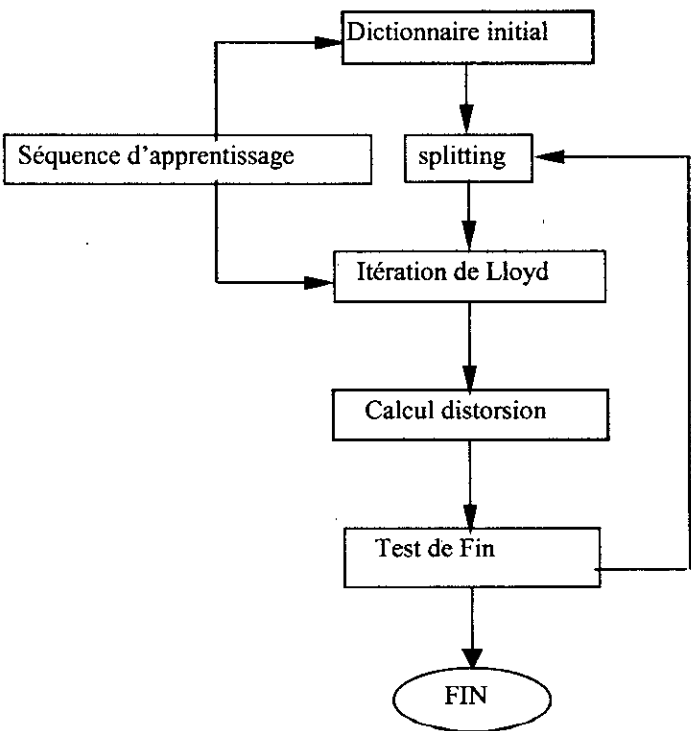


Fig 2.7 : Schéma de déroulement de l'algorithme de LBG [38]

2.2.7. Complexité de construction du dictionnaire

La complexité de l'algorithme est fonction :

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

Lors de la construction du dictionnaire, la phase des itérations de Lloyd est la plus coûteuse en terme de calcul.

En effet chaque vecteur de la source doit être comparé à chacun des L représentants, il y a autant de calculs de distorsions dans R^k et cela à chaque itération. Ainsi si n itérations sont exécutées (en considérant la métrique euclidienne) :

- Le nombre de multiplications effectuées est $N_x = k \cdot L \cdot M \cdot n$
- Le nombre d'additions effectuées est $N_+ = (2 \cdot k - 1) \cdot L \cdot M \cdot n$
- Le nombre de comparaisons effectuées est $N_c = (L - 1) \cdot M \cdot n$

Il faut ajouter aussi à cela :

- Le calcul des nouveaux centroïdes des classes nécessite $k \cdot (M - L) \cdot n$ additions et $k \cdot L \cdot n$ multiplications ;

Enfin $k \cdot (M + L) + M$ cases mémoires sont nécessaires pour stocker les vecteurs de la séquence d'apprentissage, leurs classes (auquel appartient chaque vecteur) et les vecteurs du dictionnaire.

2.2.8. Complexité de la recherche au sein du dictionnaire

Le dictionnaire ainsi obtenu ne possède aucune structure topologique pouvant faciliter le codage. Ainsi pour trouver le représentant correspondant au vecteur d'entrée de la source à coder, il est nécessaire d'effectuer L calculs de distorsion. Une telle procédure de recherche exhaustive au sein du dictionnaire a comme complexité d'ordre $L = 2^{R \cdot k}$, (R étant le débit binaire).

En résumant :

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

Ce bilan a conduit à la recherches de nouvelles méthodes de quantification vectorielle permettant un codage avec des coûts de calcul moindres.

Autrement dit, on cherche des méthodes qui consistent à imposer au dictionnaire des *contraintes structurelles* dans un but de simplifier l'opération de codage (rapidité de décision).

Une méthode alternative consiste à choisir un dictionnaire dont les vecteurs de reproduction appartiennent à un réseau algébrique de points.

2.2.9. Design d'un quantificateur vectoriel algébrique

Un quantificateur vectoriel algébrique, est un quantificateur dont les régions de décision sont uniformément répartis et le centroïde de chaque région représente un point d'une constellation ou d'un nuage de points réguliers et structurés [1,33,35,36]. Cette régularité des points présente un grand intérêt, en effet à partir d'une position donnée ou d'un point donné on peut s'orienter vers n'importe quelle direction ou vers n'importe quel point de l'espace. La particularité la plus importante des réseaux algébriques réside dans le fait que tout réseau régulier de point possède une base génératrice de tous ses points, en d'autres termes, chaque point du réseau peut être obtenu à l'aide d'une combinaison linéaire d'un ensemble réduit de vecteurs linéairement indépendants.

Cette approche offre des avantages très importants [35] dont on peut citer :

1. Le dictionnaire n'a pas à être stocké car les vecteurs y_i s'obtiennent par des combinaisons linéaires des vecteurs formant la base du réseau. Seulement cette base qui est formée par un nombre réduit de vecteurs est stockée en mémoire.
2. La recherche du mot de code optimal y_i est accélérée car elle s'opère seulement au niveau des vecteurs de la base.
3. La conséquence des deux points ci-dessus est que l'approche algébrique permet d'envisager des dictionnaires beaucoup plus grands ou de dimension plus importante que dans l'approche statistique.

2.2.10. Quantification vectorielle sphérique

Pour quantifier un signal échantillonné, nous le découpons en bloc de M échantillons. Nous considérons ensuite les M échantillons successifs de ces blocs comme les M composantes de vecteurs à M dimensions, soit X un de ces vecteurs :

$$X = [x_1, x_2, \dots, x_M]^T$$

La quantification vectorielle sphérique consiste à quantifier séparément la norme G (gain) et l'orientation x (phase) du vecteur X [33,35,38,39], soit :

$$- G = \sqrt{\sum x_i^2} \quad \text{représente la norme,} \quad (2.17)$$

$$- x = X / G \quad \text{représente l'orientation ou vecteur normalisé.} \quad (2.18)$$

Soit à quantifier d'abord l'orientation x . Cette quantification met en œuvre un premier quantificateur Q_0 auquel est associé un dictionnaire ou table de décodage S_0 définie par :

$S_0 = [y_i | i = 1, 2, \dots, N_0 - 1]$. Ce quantificateur fonctionne à un débit $R_0 = \log_2(N_0)$ et fournit le vecteur y_i qui minimise la quantité :

$$\begin{aligned} d(x, y_i) &= \|x - y_i\|^2 \\ &= \|x\|^2 + \|y_i\|^2 - 2 \cdot x^T y_i \\ &= 2 - 2 \cdot x^T y_i \end{aligned} \quad (2.19)$$

Il est évident que pour minimiser cette distance il faut maximiser le produit scalaire $x^T y_i$, produit qui n'est rien d'autre que la projection orthogonale de l'orientation x sur le vecteur y_i (ou vice versa).

Soit P cette projection maximale :

$$\max_i \{x^T y_i\} = \max_i \{\text{proj}(x)/y_i\} = P. \quad (2.20)$$

d'une manière similaire, il est possible de calculer le produit scalaire directement à partir du vecteur non normalisé X . Les calculs sont simplement mis à l'échelle par le facteur G .

$$\max_i \{X^T y_i\} = \max_i \{\text{proj}(X)/y_i\} = PG. \quad (2.21)$$

La figure 2.8 illustre une situation typique où le vecteur normalisé x est approximé par y_i .

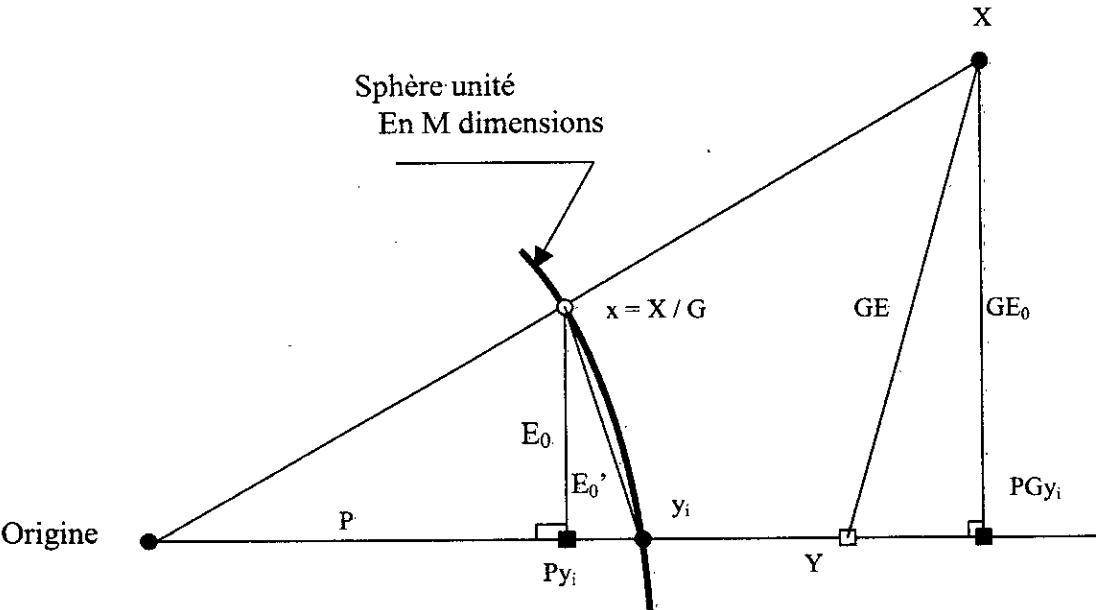


Fig 2.8 : illustration typique de la quantification vectorielle sphérique[35].

La sphère de rayon unité est représentée par un arc de cercle.

Le vecteur y_i qui sera retenu, est celui qui aura la plus grande projection P

En fait, comme il apparaît sur la figure ci dessus, une meilleure approximation de x est donnée par le vecteur Py_i étant donné que l’erreur commise E_0 reste toujours inférieure à E_0' . Il s’en suit alors, qu’une meilleure approximation pour le vecteur initial X est PGy_i , si la quantité PG pouvait être connu sans distorsion. En réalité, on quantifie également cette quantité. C’est le rôle qui sera dédié au quantificateur Q_G .

Ce quantificateur, opère avec un débit R_G et a pour but d’approximer la quantité PG avec le minimum d’erreur et de distorsion possible.

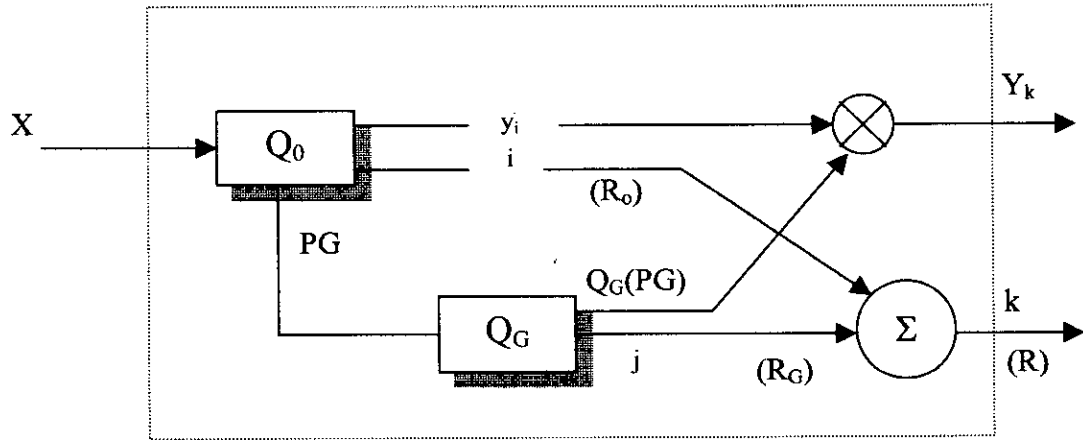


Fig 2.9: Diagramme de la quantification vectorielle sphérique

Les propriétés de la quantification vectorielle sphérique, peuvent être résumées comme suit :

Soit Q le quantificateur équivalent (quantificateur global)

$$Y_k = Q(X) = Q_0(x) Q_G(PG) \quad (2.22)$$

$$R = R_0 + R_G \quad (2.23)$$

3.2.11. Avantages de la quantification vectorielle sphérique

Ce type de quantification présente plusieurs avantages [35] dont on peut énumérer quelques uns :

1. Traiter la norme seule réduit de un la dimension du problème, on passera donc d'un volume à une surface dans un espace R^k , et permet donc l'utilisation de dictionnaires de tailles réduites.
2. Les avantages les plus importants, viennent du fait que la norme présente certaines propriétés exploitables. Dans la plupart des cas la norme varie de façon lente d'un vecteur à un autre; Il est alors aisée d'exploiter cette forme de redondance par un codage de type prédictif.
3. Il est plus facile de croire que quantifier vectoriellement un vecteur X en le considérant comme étant issu d'une source uniforme dans un hypercube de dimension k soit la meilleure approche. Mais en fait, loin de tendre vers l'uniformité, la distribution de la source est concentrée dans une **coquille sphérique** centrée à l'origine; phénomène appelé « **sphere hardening** » qui se produit lorsque les composantes du vecteur ne sont pas de natures différentes mais au contraire elles émanent d'une même source.

PARTIE II

Réseaux réguliers de points, Approche
Hybride et Diffusion par Soustraction

Chapitre 3

Les Réseaux réguliers de points

L'étude des réseaux, est étroitement liée au problème de l'empilement des sphères [39] de même rayon dans $\mathbf{R}^M$. Ici, il s'agit de disposer les sphères de manière à obtenir l'empilement le plus dense (rapport du volume des sphères sur le volume occupé). Par exemple en dimension 2, la façon la plus dense «d'empiler» des sphères sur un plan, est l'arrangement hexagonal. Pour le cas de 3 dimensions (exemple : le cas des boules de billard), il est possible de les disposer sur un plan de façon optimale en formant une première couche selon l'arrangement hexagonal. Il est possible ensuite de poser une deuxième couche identique mais décalée de manière à ce que les boules occupent les « creux » de la première couche; on continue ainsi à placer la troisième puis la quatrième ainsi de suite. La figure 3.1 montre deux manière de placer la troisième couche [39]. Les deux empilements ont la même densité, mais seul l'empilement **b** correspond à un réseau régulier.

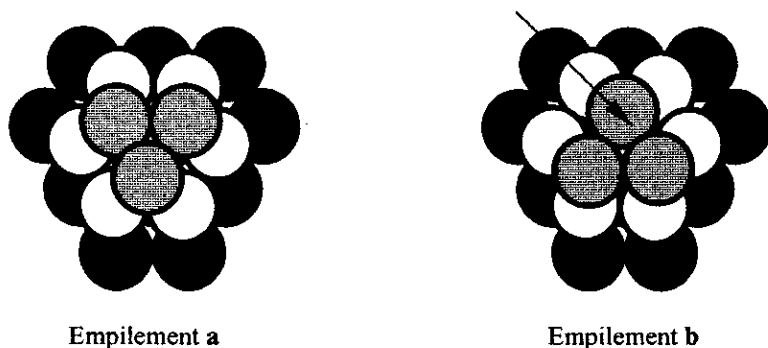


Fig 3.1 : Deux empilements de sphères en 3 dimensions obtenus en superposant des couches d'arrangement hexagonal. La troisième couche peut être en coïncidence avec la première ou non.

3.1. Définition d'un réseau

Un réseau régulier de points dans $\mathbf{R}^M$ est l'ensemble de points x qui s'obtiennent par une combinaison linéaire de M vecteurs de bases indépendants [1,35,38], $z_1, z_2, z_3, \dots, z_M$, avec des coefficients de proportionnalité entiers, $k_1, k_2, \dots, k_M$:

$$L_M = \{ x \mid x = \sum_i k_i z_i \} ; \quad k_i : \text{entier} \quad (3.1)$$

On peut facilement remarquer qu'un réseau régulier de points reste invariant lorsqu'on le translate pour amener son origine à l'un de ses points.

3.2. Réseaux et codes correcteurs d'erreurs

Plusieurs réseaux de points, ont été établis à partir des codes correcteurs d'erreurs [35,41]. En effet un code correcteur d'erreurs (soit le cas binaire), est un sous-ensemble de N M -uplets binaires parmi les 2^M possibles (sommets de l'hypercube unité dans $\mathbf{R}^M$). N est généralement une puissance de 2.

Exemple : Code à répétition

$N=2$, $M=3$; (voir figure 3.2 [35])

Lors de la transmission d'un mot code, la valeur binaire peut se trouver inversée (erreur qui peut être dû au canal de transmission), sur quelques positions du triplet.

On appelle la *distance de Hamming*, d_H , entre deux triplets le nombre de positions sur lesquels ceux ci diffèrent. Par exemple, la distance de Hamming entre les deux triplets suivants est $d_H([1\ 0\ 1], [0\ 1\ 1]) = 2$.

On appelle aussi *poids* d'un mot de code le nombre de « 1 » qu'il contient.

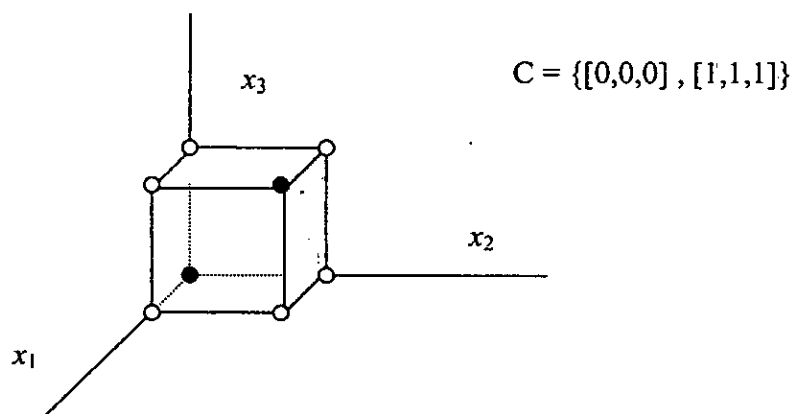


Fig 3.2 : Interprétation géométrique du code correcteur à répétition $C[3,1,3]$

Un code peut corriger t erreurs si la plus petite distance de Hamming $d_H \min$, entre paire de M -uplets est supérieure à $t / 2$. Enfin, les propriétés globales d'un code se résument par les trois entiers : $[M, \log_2 N, d_H \min]$. Ainsi le code de la figure ci dessus est $[3, 1, 3]$.

Leech et Sloane ont proposé diverses constructions de réseaux à partir d'un code correcteur d'erreurs, $C[M, \log_2 N, d_H \min]$.

La construction A, consiste à prendre tous les points de Z_M qui sont congrus, modulo 2 par composante, à un mot du code C:

$$L = \{ x \mid x = [x_1, x_2, ..., x_M] \in Z_M \text{ et } [x_1 \bmod 2, x_2 \bmod 2, ..., x_M \bmod 2] \in C \}. \quad (3.2)$$

Interprétation :

Si l'on écrit les composantes entières x_i en représentation binaire « complément à deux » le M -uplet formé par le bit de plus faible poids de chaque composante est un mot du code C. On pourra également remarquer, qu'on peut écrire : $x = y + c$; où $c \in C$ et où y est un vecteur du réseau à composantes paires : $2 Z_M$. Le vecteur y étant obtenu en forçant à 0 les bits de plus faible poids des composantes de x .

Il s'ensuit alors que le réseau L peut s'écrire comme l'union de copies du réseau $2 Z_M$, translatées par un vecteur du code :

$$L = \bigcup_i 2Z_M + c_i \quad ; \quad \text{où } c_i \in C. \quad (3.3)$$

On peut également considérer cette relation comme étant l'union des copies du code C translatées aux sommets du réseau à composantes paires comme l'illustre la figure 3.3 [35]:

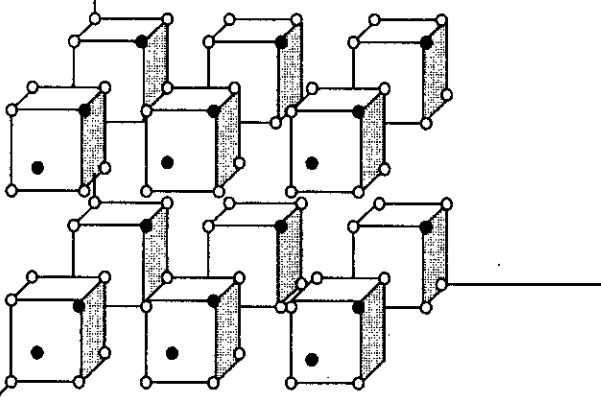


Fig. 3.3 : Réseau régulier obtenu par la construction A appliquée au code à répétition de la figure 3.2

Un autre réseau important est le D_M [35,42]; il contient les points de Z_M tels que la somme de leurs composantes soit paire.

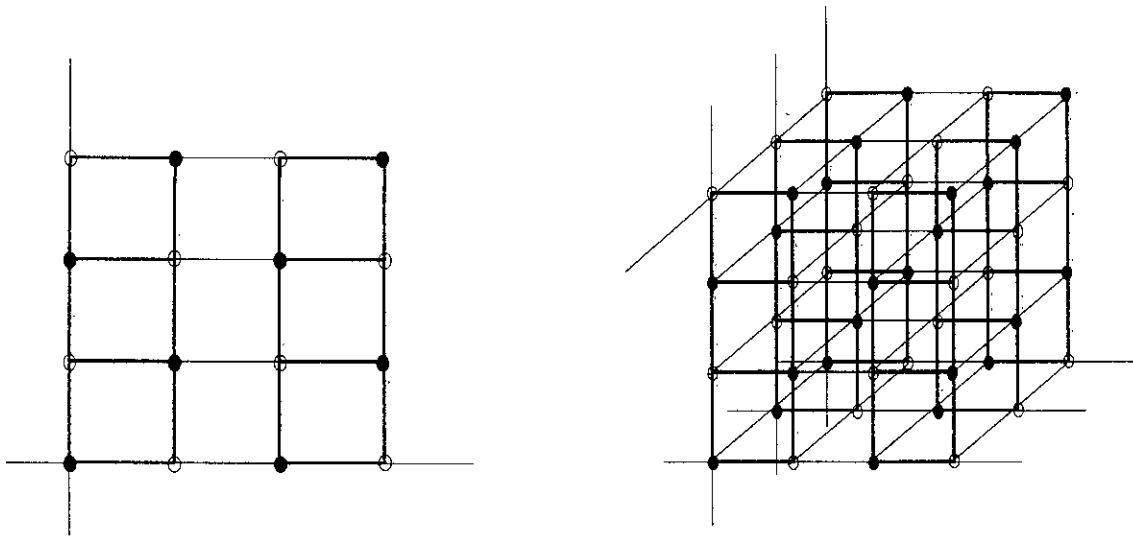


Fig.3.4 : Réseau D_M pour 2 et 3 dimensions.

Grâce à D_M , on peut obtenir des réseaux (par la *construction en B*); on considère un code linéaire C dont le poids des mots est divisible par 4. Cette construction, consiste à prendre tous les points de D_M qui sont congrus modulo 2 par composante à un mot de C . Où encore, on peut former le réseau comme l'union de réseaux $2 D_M$ translatés par chaque mot c_i du code C .

$$L = \bigcup_i 2 D_M + c_i \quad ; \quad \text{où } c_i \in C \tag{3.4}$$

Les réseaux les plus connus et les plus intéressants pour la quantification vectorielle sont :

1. Le réseau de Gosset E_8 en 8 dimensions,
2. Le réseau de Barnes-Wall Λ_{16} en 16 dimensions,
3. Le réseau de Leach Λ_{24} en 24 dimensions.

3.3. Réseaux sphériques

Soit $\mathbf{x}$ un *point* de R^k : $\mathbf{x} = (x_1, x_2, \dots, x_k)^T$

Soit une *sphère* de centre $\mathbf{u} = (u_1, u_2, \dots, u_k)^T$ et de rayon ρ qui représente l'ensemble des points $\mathbf{x}$ tel que :

$$\sum_{i=1}^k (x_i - u_i)^2 = \rho^2 \tag{3.5}$$

Dans l'espace de dimension k , on considère un empilement régulier de sphères identiques de rayon ρ . Un *réseau régulier* (lattice ou treillis) Λ est alors constitué de l'ensemble des centres de ces sphères [33,38,39]. Par convention $\mathbf{O}$ est considéré comme le *centre du réseau*.

Il est alors aisé de montrer qu'il existe, en plus de $\mathbf{O}$, k sphères de centres $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{ik})^T$ telles que tout point du réseau peut être déterminé par la somme :

$$\sum_{i=1}^k \varepsilon_i \cdot \mathbf{v}_i \quad | \quad \varepsilon_i \in \mathbb{Z} \quad (3.6)$$

Les vecteurs $\mathbf{v}_i$ forment la **base du réseau**, ces vecteurs sont *linéairement indépendants*. On détermine le parallélogramme fondamental du treillis Λ [38] comme la région formée par :

$$\sum_{i=1}^k \theta_i \cdot \mathbf{v}_i \quad | \quad 0 \leq \theta_i \leq 1 \quad (3.7)$$

L'espace entier peut être recouvert en sommant ces régions (qui ne se recouvrent pas)

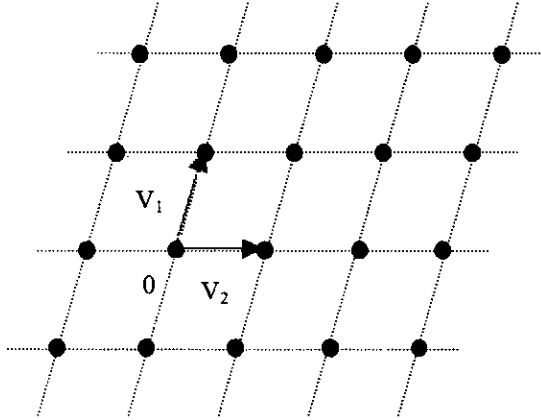


Fig 3.5 : Réseau bidimensionnel

La *matrice génératrice* du réseau est donnée par :

$$\mathbf{M} = \begin{pmatrix} \mathbf{V}_{11}, & \mathbf{V}_{12}, & \dots, & \mathbf{V}_{1m} \\ \mathbf{V}_{21}, & \mathbf{V}_{22}, & \dots, & \mathbf{V}_{2m} \\ \vdots & & & \vdots \\ \mathbf{V}_{k1}, & \mathbf{V}_{k2}, & \dots, & \mathbf{V}_{km} \end{pmatrix} = \begin{pmatrix} \mathbf{V}_1^T \\ \vdots \\ \mathbf{V}_k^T \end{pmatrix}$$

Il est évident qu'il est plus facile de d'écrire un réseau k -dimensionnel à l'aide de vecteurs de m coordonnées.

Ainsi un vecteur y du réseau est déterminé par :

$$y = M^T \cdot \varepsilon \quad \text{avec} \quad \varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_k)^T \quad \mid \quad \varepsilon_i \in \mathbb{Z} \quad (3.8)$$

On peut également définir un réseau régulier Λ comme étant l'ensemble des points y de $\mathbb{R}^k$ tel que :

$$\Lambda = \left\{ y \in \mathbb{R}^k \quad \mid \quad \exists \varepsilon \in \mathbb{Z}^k, \quad y = M^T \cdot \varepsilon = \sum_{i=1}^k \varepsilon_i \cdot v_i \right\} \quad (3.9)$$

On définit la *matrice de Gram* du réseau par : $A = M \cdot M^T$,

Son déterminant est noté (M : matrice carrée) : $\det \Lambda = \det A = (\det M)^2$,

Le volume d'un voronoï (parallélotope) est donc : $\det M = (\det \Lambda)^{1/2}$,

Deux réseaux, Λ_1 et Λ_2 , peuvent être **équivalents** ou **similaires** après et/ou réflexion et/ou changement d'échelle. Ceci est noté : $\Lambda_1 \cong \Lambda_2$.

Un réseau Λ_k admet un *dual* ou *réciroque* Λ_k^* qui est donné par :

$$\Lambda_k^* = \{ y \in \mathbb{R}^k \quad \mid \quad y^T \cdot u \in \mathbb{Z}, \quad \forall u \in \Lambda_k \} \quad (3.10)$$

Les réseaux réguliers de points sont, à l'origine, des solutions envisagées aux problèmes suivants [39]:

1. Empilement des sphères dans un espace donné
2. Recouvrement de l'espace par des sphères
3. Recherche du «nombre de contacts» (kissing number)

3.4. Problème d'empilement des sphères dans un espace

On recherche, dans un espace de dimension k donnée, l'empilement le plus dense de sphères (toutes sont équivalentes et 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 [33,38,39] (la densité maximale peut être obtenue avec des réseaux non réguliers).

Dans ce cas, on veut maximiser la **densité du réseau** définie par :

$$\Delta = \text{proportion occupée par les sphères}$$

$$= \frac{\text{Volume d'une sphère}}{\text{Volume d'une région fondamentale}}$$

$$= \frac{V_k \cdot \rho^k}{(\det \Lambda)^{1/2}} \tag{3.11}$$

où :

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

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

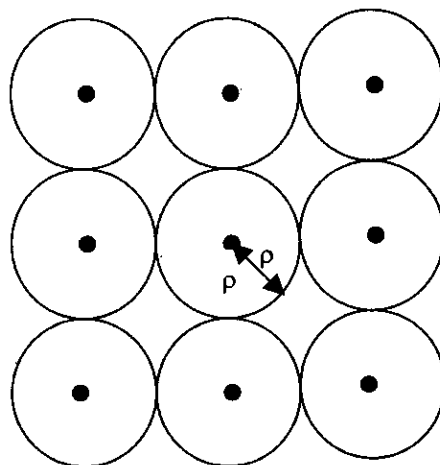


Fig 3.6: Un empilement de sphères dans l'espace bidimensionnel

3.5. Problème de recouvrement d'espace par des sphères

Dans ce cas, on désire obtenir le recouvrement le plus économique de l'espace euclidien de dimension k avec des sphères [33,38,39]. Ces dernières sont toutes identiques recouvrent $\mathbb{R}^k$, et se recouvrent également entre elles, ce dernier recouvrement doit être minimal. De la même manière que dans le cas précédent la solution n'est connue, pour $k \geq 3$, que si on considère les centres des sphères appartenant à un réseau régulier de points.

Soit r le rayon des ces sphères ; on veut minimiser θ , la **densité du recouvrement** («épaisseur ») définit par :

$$\begin{aligned} \theta &= \frac{\text{nombre moyen de sphères contenues dans l'espace}}{\text{Volume d'une sphère}} \\ &= \frac{\text{Volume d'une région fondamentale}}{\text{Volume d'une sphère}} \\ &= \frac{V_k \cdot r^k}{(\det \Lambda)^{1/2}} \end{aligned} \quad (3.12)$$

r définit le **rayon de recouvrement** du réseau, il correspond à la borne *supérieure minimale* de la distance entre un point x de l'espace $\mathbb{R}^k$ et le plus proche point du réseau.

Posons y_i les points du réseau, et $d(x, y_i)$ la distance entre les deux points :

$$r = \sup_{x \in \mathbb{R}^k} \inf_{y_i \in \Lambda} d(x, y_i) \quad (3.13)$$

Ces points isolés, à la distance r de leurs plus proches voisins dans le réseau, représentent des **trous**.

Le problème de recouvrement ainsi posé apparaît comme le dual de celui de l'empilement [38], cependant ils sont différents :

- Pour le premier, il s'agit de minimiser r
- Pour le second, il s'agit de maximiser ρ

Ainsi, pour une dimension donnée, les réseaux solution de chacun des ces problèmes sont souvent différents.

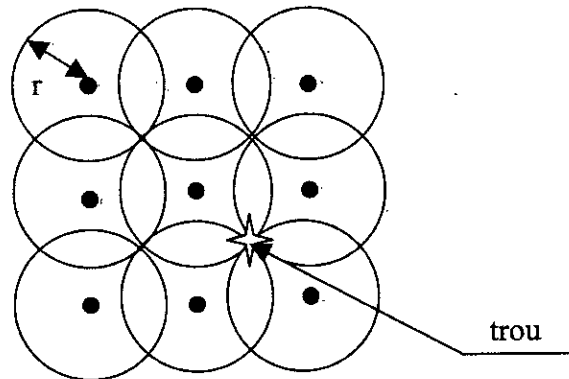


Fig.3.7: Un regroupement de l'espace bidimensionnel par des sphères.

3.6. Problème du nombre de contacts (« kissing number »)

Dans ce cas, 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 [33,38,39]. 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.

Le nombre τ est variable si l'on considère des réseaux non réguliers, il est fixe dans le cas où l'on considérerait des réseaux réguliers de points.

τ sera donc connu que si on considère des points appartenant à un réseau régulier, d'une manière exacte, la solution optimale n'est connue que pour des espaces de dimensions $k=1,2,3,8$ et 24 . La détermination du nombre de contacts intervient 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, il s'ensuit que la recherche du nombre τ permet de mieux les répartir.

Le calcul de τ est facilité par un puissant outil de calcul mathématique : se sont les **séries thêta** [38] qui indiquent le nombre de points sur une surface d'une sphère.

3.7. Introduction des séries thêta

Soit y un point du réseau Λ : $y = M^T \cdot \varepsilon$

La norme euclidienne L_2 de y est :

$$\|y\|^2 = \sum_{i=1}^k y_i^2 = y^T \cdot y = \varepsilon^T \cdot M \cdot M^T \cdot \varepsilon = \varepsilon^T \cdot A \cdot \varepsilon = f(\varepsilon) \quad (3.14)$$

Cette norme donne la **forme quadratique** associée au réseau.

En théorie des nombres un problème se pose : on désire connaître le nombre de façons

d'écrire un entier m comme la somme de k carrés (ex : $m = \sum_{i=1}^k y_i^2$).

Les réseaux présentent du fait de leur forme quadratique, une formulation à ce genre de problème :

Considérant un réseau régulier de points Λ , soit N_m le **nombre de vecteurs y de Λ de norme m** (c.a.d : $y^T \cdot y = m$), N_m représente également le nombre de fois que la forme quadratique associée à Λ représente m .

Le calcul de N_m est facilité par les séries thêta du réseau Λ qui sont définies comme suit:

$$\begin{aligned} \theta_{\Lambda}(z) &= \sum_{y \in \Lambda} q^{y^T \cdot y} \\ &= \sum_{m=0}^{+\infty} N_m \cdot q^m \quad \text{où} \quad q = e^{i \cdot \pi \cdot z} \end{aligned} \quad (3.15)$$

Les séries thêta produisent le nombre de points qu'il y a à chaque distance de l'origine (le nombre de points sur chaque sphère de rayon ou encore, sur chaque surface d'énergie m). Les points étant répartis sur des sphères concentriques, pour connaître le **nombre total** N_T de points dans la sphère de rayon, il faut faire la somme des points sur chacune des sphères de rayon inférieur :

$$N_T = \sum_{i=0}^m N_i \quad (3.16)$$

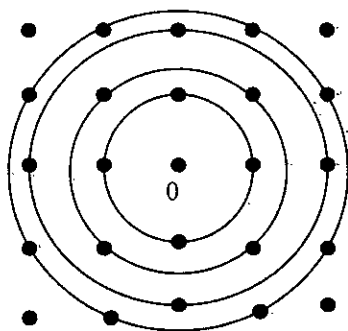


Fig .3.8: Sphères concentriques d'un réseau bidimensionnel.

La plupart des réseaux s'expriment, lors du développement de leurs séries thêta, comme des fonctions des **séries thêta jacobi** [38] :

$$\theta_2(z) = \sum_{m=-\infty}^{+\infty} q^{(m+1/2)^2} = 2 \cdot q^{1/4} + 2 \cdot q^{9/4} + 2 \cdot q^{25/4} + \dots$$

$$\theta_3(z) = \sum_{m=-\infty}^{+\infty} q^{m^2} = 1 + 2 \cdot q + 2 \cdot q^4 + 2 \cdot q^9 + \dots$$

$$\theta_4(z) = \sum_{m=-\infty}^{+\infty} (-q)^{m^2} = 1 - 2 \cdot q + 2 \cdot q^4 - 2 \cdot q^9 + \dots$$

On remarque qu'elles s'expriment comme un développement en puissance de q donnant le nombre de vecteurs sur les sphères successives autour de l'origine.

3.8. Les réseaux réguliers les plus importants

Le tableau ci dessous illustre quelques réseaux solutions des différents problèmes évoqués [38]:

k	1	2	3	4	5	6	7	8	12	16	24	
Meilleur empilement		A ₂	A ₃	D ₄	D ₅	E ₆	E ₇	E ₈	K ₁₂	Λ ₁₆	Λ ₂₄	
Plus grand Nombre contacts	de		A ₂	A ₃	D ₄	D ₅	E ₆	E ₇	E ₈	P _{12a}	Λ ₁₆	Λ ₂₄
		2	6	12	24	40	72	126	240	840	4320	196560
Meilleur recouvrement		A ₂	A ₃ [*]	A ₄ [*]	A ₅ [*]	A ₆ [*]	A ₇ [*]	A ₈ [*]	A ₁₂ [*]	A ₁₆ [*]	Λ ₂₄	
Meilleur quantificateur		A ₂	A ₃ [*]	D ₄	D ₅ [*]	E ₆ [*]	E ₇ [*]	E ₈	K ₁₂	Λ ₁₆	Λ ₂₄	

Explication du tableau ci dessus :

- Les réseaux encadrés offrent la solution optimale globale (parmi les réseaux réguliers ou non)
- Les réseaux à gauches du trait vertical sont optimaux parmi les réseaux réguliers
- Les Z^k ($k \geq 1$) représentent les réseaux cubiques
- Les A_k ($k \geq 1$) représentent les réseaux « racines » communément appelés « root lattices »
- La majorité de ces réseaux appartiennent à la famille des réseaux par « strates » noté Λ_k , on notera également qu'il existe de nombreuses équivalences : $\Lambda_1 \cong Z \cong A_1 \cong A^*_1 \cong D_1$; $\Lambda_2 \cong A_2 \cong A^*_2$; $\Lambda_3 \cong A_3 \cong D_3$; $\Lambda_4 \cong D_4 \cong D^*_4$; $\Lambda_5 \cong D_5$; $\Lambda_6 \cong E_6$; $\Lambda_7 \cong E_7$; $\Lambda_8 \cong E_8 \cong E^*_8$; ...
- A_2 représente le réseau hexagonal
- A_3 représente le réseau cubique à face centrée
- A^*_3 représente le réseau cubique à corps centré
- K_{12} représente le réseau de Coxeter-Todd
- Λ_{16} représente le réseau de Barnes-Wall
- E_8 représente le réseau de Gosset
- Λ_{24} représente le réseau de Leech.

3.9. Le réseau de Gosset E_8 .

Le réseau E_8 porte de nom de **Gosset** [35,39] ou encore de réseau en «**diamant**», il est défini par la relation :

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

On remarque donc que le réseau E_8 correspond à l'union du réseau D_8 avec le *réseau décalé* (appelé **coset** [25,26]) $\left[\frac{1}{2} \right] + D_8$ ou encore :

$$E_8 = \{ \mathbf{y}' = (y_1', y_2', \dots, y_8')^T, \mathbf{y}'' = (y_1'', y_2'', \dots, y_8'')^T /$$

$$y_i' \in \mathbb{Z} \text{ et } \sum_{i=1}^8 y_i' = 0 \pmod{2}, y_i'' \in (\mathbb{Z} + \frac{1}{2}) \text{ et } \sum_{i=1}^8 y_i'' = 0 \pmod{2} \} \quad (3.17)$$

Avec :

$$\begin{cases} \det A = 1, \\ \tau = 240 \\ \rho = \frac{1}{\sqrt{2}} \\ r = \rho \cdot \sqrt{2} = 1 \\ \theta_{E_8}(z) = \frac{1}{2} \cdot (\theta_2(z)^8 + \theta_3(z)^8 + \theta_4(z)^8) = 1 + 240 \cdot q^2 + 2160 \cdot q^4 + \dots \end{cases}$$

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

Tableau 3.1 : Nombre de points de contact des sphères $E_8(m)$

3.9.1. Quantification rapide avec un réseau de Gosset E_8 :

On a vu que le réseau E_8 n'est rien d'autre que l'union des réseaux D_8 et D_8 décalé de $[1/2]$, ainsi pour quantifier un vecteur $\mathbf{x} = (x_1, x_2, \dots, x_8)^T \in \mathbb{R}^8$, on procède comme suit [38]:

1. Calcul des vecteurs $f(\mathbf{x})$ et $g(\mathbf{x})$, ensuite choix de celui dont la somme des coordonnées est paire, soit y_0 ce vecteur ; (pour le calcul de $f(\mathbf{x})$ et $g(\mathbf{x})$, voir annexe).
2. Calcul des vecteurs $f(\mathbf{x}-[1/2])$ et $g(\mathbf{x}-[1/2])$, puis sélection de celui dont la somme des coordonnées est paire, on ajoute ensuite au vecteur choisit $[1/2]$ et on note ce vecteur y_1 ;
3. Calcul des normes $d(y_0, \mathbf{x})$ et $d(y_1, \mathbf{x})$, on retient alors le vecteur y_i le plus proche de $\mathbf{x}$.

3.9.2. Complexité de la quantification dans E_8

La complexité de la quantification dans E_8 , sera donc la somme des complexités suivantes :

- Complexité de la quantification dans D_8
- Complexité de la quantification dans D_8 décalé
- Recherche du minimum des deux normes.

3.9.3. Quantification sphérique avec le réseau de Gosset E_8

On notera par $E_8(m)$ l'ensemble des points d'une sphère et par $N_8(m)$, le nombre de points de cet ensemble :

$$E_8(m) = \{ \mathbf{y} | \mathbf{y} \in E_8 \cap S_m \} \quad (3.18)$$

où S_m représente la sphère de rayon, $E_8 = \bigcup_m E_8(m)$

Si on utilise $E_8(m)$ comme quantificateur, il y a tout un choix de débits par dimension suivant la sphère m utilisée. Le tableau 3.2 [35] donne pour chaque sphère m entre 1 et 15 le nombre $N_8(m)$ et le débit correspondant en bit/dimension.

m	P_m	$N_8(m)$	Débit/dim
1	8	240	0.988
2	15	2160	1.385
3	24	6720	1.589
4	42	17520	1.762
5	54	30240	1.861
6	57	60480	1.986
7	103	82560	2.042
8	119	140400	2.137
9	124	181680	2.184
10	162	272160	2.257
11	220	319680	2.286
12	208	490560	2.363
13	279	527520	2.376
14	305	743040	2.438
15	343	846720	2.461

Tableau 3.2 : Tableau donnant le nombre de leaders, le nombre de points de contacts et le débit par dimension pour chaque sphère d'énergie m .

Ayant choisi une sphère m , nous pouvons ordonner les points entre 0 et $N_8(m)-1$, ce qui représente les vecteurs y_i du quantificateur.

Pour exploiter convenablement ce quantificateur, deux problèmes restent posés :

- Trouver le vecteur le plus proche voisin y du vecteur x à quantifier
- Déterminer l'indice i de ce voisin.

3.9.4. Algorithme de quantification avec une sphère de E_8

Les réseaux de Gosset $E_8(m)$ possèdent une propriété très intéressante, rendant la recherche des vecteurs du réseau très simple à élaborer et à implémenter.

En effet, toute permutation des coordonnées d'un vecteur de cet ensemble est encore un vecteur de cet ensemble [33,35,38].

Cette propriété nous permet de définir une relation d'équivalence entre vecteurs que l'on notera $\approx$.

On dira alors que les vecteurs y et z de $E_8(m)$ sont équivalents, $y \approx z$, s'ils se déduisent l'un de l'autre par permutation des composantes.

Exemple :

$[2,0,0,-2,0,0,0,0] \approx [0,0,-2,0,0,2,0,0]$.

Cette relation d'équivalence induit une partition de l'ensemble $E_8(m)$ en P_m classes. Ce nombre P_m est tabulé dans le tableau 3.3 suivant :

j	K(j)	H(j)	Leaders (m=1)
1	28	0	2 2 0 0 0 0 0 0
2	56	28	2 0 0 0 0 0 0 -2
3	1	84	1 1 1 1 1 1 1 1
4	28	85	1 1 1 1 1 1 -1 -1
5	70	113	1 1 1 1 -1 -1 -1 -1
6	28	183	1 1 -1 -1 -1 -1 -1 -1
7	28	211	0 0 0 0 0 0 -2 -2
8	1	239	-1 -1 -1 -1 -1 -1 -1 -1

Tableau 3.3 : Liste des leaders (directeurs) des 8 classes d'équivalence de la sphère $m=1$, avec le nombre d'éléments $K(j)$ de chaque classe ainsi que le cumul $H(j)$ des classes précédentes.

Dans chacune des classes on fait des permutations des composantes des vecteurs selon l'ordre lexicographique, et l'on considère le plus grand d'entre eux comme «leader de la classe » (toujours par rapport à l'ordre lexicographique).

Les P_m leaders sont alors eux même ordonnés dans l'ordre lexicographique par l'indice $j = 1,2,...,P_m$. Le tableau 3.3 donne pour la sphère $m=1$ la liste des $P_m=8$ directeurs dans l'ordre lexicographique.

Soit q le nombre de valeurs distinctes que prennent les composantes du leader. On considère alors, w^0 , le nombre de répétitions de la plus grande composante a_0 , w^1 , le nombre de répétitions de la deuxième composante de plus grande valeur a_1 et ainsi de suite jusqu'à w^{q-1} , le nombre de répétitions de la plus petite valeur a_{q-1} .

La cardinalité d'une classe est donnée par la relation suivante dénombrant les permutations avec répétitions :

$$K(j) = \frac{8!}{(w^0! w^1! w^2! \dots w^{q-1}!)}$$

(3.19)

Pour indexer les $K(j)$ permutations de la classe, il est nécessaire de généraliser les nombres w^d en leur affectant un index k de la façon suivante.

Un vecteur quelconque de la classe j compte toujours q valeurs distinctes dans ses composantes qui peuvent être dans le désordre.

Considérant un indice quelconque de composante, d , on définit alors w^d_k comme le nombre des répétitions de la valeur a_d qui reste dans les positions k à 8.

Dans l'exemple suivant, on illustrera le calcul des valeurs w^d_k , dans le cas d'un vecteur y de la sphère $m = 9$, soit le leader suivant, $[4,4,4,4,2,2,0,0]^T$ la fenêtre à considérer dans le cas de l'indice $k=6$, est montrée par un rectangle en pointillé comme le montre l'exemple ci dessous :

Position $k =$

1

2

3

4

5

6

7

8

↓

↓

↓

↓

↓

↓

↓

↓

X=[

4,

2,

4,

0,

0,

4,

4,

2]

,

a_d	w^d_k	1	2	3	4	5	6	7	8
4	0	4	3	3	2	2	2	1	0
2	1	2	2	1	1	1	1	1	1
0	2	2	2	2	2	1	0	0	0

Exemple 3.1 : Calcul des valeurs w^d_k , dans le cas d'un vecteur y de la sphère $m = 9$

Pour retrouver l'index lexicographique t d'un vecteur parmi les $K(j)$ membres de sa classe à partir de ses nombres w_k^d , on utilise la formule du rang de **Schalkwijk** ; ce rang est donné par la relation suivante où $d(k)$ représente l'indice d de la k -ième composante : $a_d(k)$.

$$t = \sum_{k=1}^8 \sum_{d=0}^{d(k)-1} \frac{(M - k)}{(w_k^d - 1) \prod_{\substack{i=0 \\ i \neq d}}^{q-1} (w_k^i)} \quad (3.20)$$

N.B : $0! = 1$, $(-1)! = \infty$ et $M = 8$;

3.9.4.1. Algorithme de codage avec une sphère de E_8

Soit $\mathbf{x} = [x_1, x_2, \dots, x_8]^T$ un vecteur à huit composantes réelles à quantifier par $E_8(m)$. On peut décomposer l'algorithme en quatre étapes :

1. Réordonner les composantes du vecteur par ordre décroissant des valeurs (c.à.d, la plus grande composante en première position), ensuite on appellera $\tilde{\mathbf{x}}$ ce nouveau vecteur.
2. Trouver le leader j^* le plus proche en cherchant parmi tous les leaders celui qui conduit au plus grand produit scalaire :

$$\tilde{\mathbf{x}}^T \tilde{\mathbf{y}}_{j^*} = \max_j \left(\{ \tilde{\mathbf{y}}^T \tilde{\mathbf{x}}_j \mid j = 1, 2, \dots, N_8(m) \} \right) \quad (3.21)$$

3. Déterminer le rang t du vecteur à l'intérieur de la classe j^* par la formule du rang de Schalkwijk.
4. Déterminer l'indice i du vecteur quantifié y_i par la relation suivante :

$$i = t + H(j^*) = 0, 1, \dots, N_8(m) - 1 \quad (3.22)$$

3.9.4.2. Algorithme de décodage

L'algorithme de décodage se décompose en trois étapes :

1. Retrouver la classe à laquelle appartient le vecteur j^* tel que : $H(j^*) \leq i < H(j^* + 1)$
2. Retrouver le rang t dans la classe j^* vérifiant : $t = i - H(j^*)$
3. Faire subir au leader de la classe j^* la permutation inverse qui le fera passer du rang $K(j) - 1$ au rang t .

Chapitre 4

Approche Hybride et technique de Diffusion

4.1. Approche hybride de quantification (Statistique / Algébrique)

Cette nouvelle technique que nous proposons, est une approche intermédiaire entre les approches statistique et algébrique. En effet, elle a pour but d'éviter les inconvénients de l'approche statistique qui se résument par, les contraintes de stockage, de complexité de calcul et de recherche au sein du dictionnaire ; et par ceux de l'approche algébrique qui sont essentiellement dus au fait qu'elle ne s'applique que pour des sources à distributions uniformes, cas rare voir inexistant dans le cas des sources de signaux informationnels. Ceci d'une part, d'autre part elle vise à exploiter les avantages des deux techniques sus citées, qui sont : utilisation dans le cas de sources à distributions inconnues (non-uniformes), des dictionnaires générés algébriquement donc non stockés en mémoire, tout en tenant compte des statistiques de la source.

Pour réaliser un tel quantificateur, on s'est posé les deux hypothèses suivantes :

- On considère que la source à coder est **globalement** non-uniforme (distribution quelconque),
- On considère que cette source est **localement** uniforme.

Pour exploiter ces deux hypothèses, on se doit d'abord de déterminer des zones d'uniformité de la source considérée, on procède alors de la manière suivante :

En utilisant l'algorithme du LBG, on détermine quelques points statistiques (4,8,16, ...) de la source représentant les zones de l'espace où les nuages de points sont les plus concentrés ou encore, les zones où la probabilité d'avoir un point de la source est la plus grande ; on considère alors que tout autour de ces points la source présente une certaine uniformité.

Ce dictionnaire statistique ainsi établi possède un nombre réduit de code-words, ce qui évite le problème de contraintes de stockage, de complexité de calcul et de recherche au sein de ce dernier.

Une fois ce dictionnaire établi, chaque point de ce dernier sera considéré comme le centre d'un réseau algébrique de points, dans notre cas on a considéré le réseau de Gosset $E_8(1)$ possédant 240 points de contacts; de cette manière, chaque point statistique servira d'aiguilleur vers 240 points de l'espace; points qui seront générés algébriquement.

De cette manière, si par exemple on ne prend que quatre points statistiques de la source, on pourra générer 960 points d'une manière algébrique, c'est à dire suffisamment de points pour couvrir une grande partie de l'espace de travail considéré, ainsi on aura également pu éviter le problème des réseaux algébriques qui ne s'appliquent que pour des sources à distributions uniformes, vu que les points centres des réseaux considérés, sont des points qui suivent les statistiques de la source considérée.

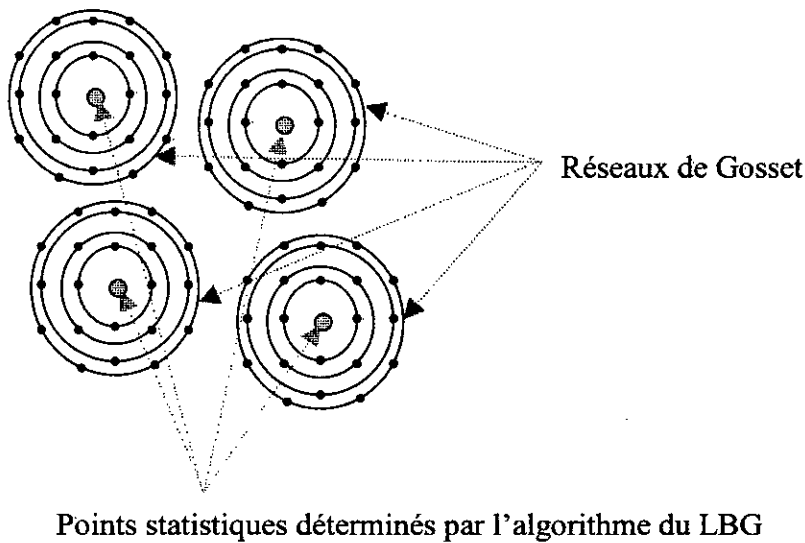


Fig4.1 : Structure du quantificateur hybride proposé.

4.1.1 Algorithme de codage avec le quantificateur hybride

La première étape consiste à chercher un dictionnaire statistique, on opère alors comme suit :

- Fixer la longueur N (N réduit) du dictionnaire,
- Appliquer l'algorithme du LBG,
- Stocker les points statistiques obtenus dans un fichier.

Après avoir établi ce dictionnaire, on procède à l'étape suivante, qui se résume de la manière suivante :

Soient $\mathbf{x}$ un vecteur d'entrée de dimension 8, et y_i $i = 1, 2, \dots, N$ les code-words du dictionnaire statistique;

- Calculer les distance euclidiennes $d_i(\mathbf{x}, y_i) = \sum_{j=1}^8 (x_j - y_{ij})^2$,
- Choisir l'indice i du code-word pour lequel la distance euclidienne est minimale $d_i(\mathbf{x}, y_i) < d_j(\mathbf{x}, y_j) \forall j = 1, 2, \dots, N$ et $i \neq j$,
- Translater le vecteur d'entrée $\mathbf{x}$ vers y_i ,
 $\tilde{\mathbf{x}} = ((x_1 - y_{i1}), (x_2 - y_{i2}), \dots, (x_N - y_{iN}))$,
- Calculer les distances euclidiennes entre $\tilde{\mathbf{x}}$ et les vecteurs du réseau de Gosset centré en y_i ,
- Choisir l'indice j du code-word pour lequel la distance euclidienne est minimale
- Le couple (i, j) ainsi obtenu sera le code du vecteur d'entrée $\mathbf{x}$.

4.12. Algorithme de décodage avec le quantificateur hybride

Soit (i, j) le code reçu d'un vecteur à décoder, on procède alors comme suit :

- Génération du vecteur du réseau de Gosset à partir de l'indice j , soit g_j ce vecteur,
- Recherche du vecteur statistique dans le dictionnaire à partir de l'indice i , soit y_i ce vecteur,
- Calcul du vecteur à restituer, soit $\tilde{\mathbf{x}} = g_j + y_i$.

4.1.2. Résultats expérimentaux

La figure 4.2 représente les résultats obtenus avec la technique hybride pour différents nombres de points statistique, on remarque que plus on augmente ce nombre (tout en le gardant très inférieur à celui requis par l'approche statistique), l'image restituée devient de qualité perceptuelle de plus en plus intéressante.



a : Image originale codée sur 8 bits/pixel



b : Image codée par un réseau de Gosset
E8(1) seul, 1.5 bits/pixel, taux de
compression 81.25%



c : Image codée avec un réseau de Gosset
E8(1) + 4 points statistiques, 1.625 bit/p,
taux de compression : 79.69%



d : Image codée avec un réseau de Gosset
E8(1) + 8 points statistiques, 1.75 bit/p,
taux de compression : 78.125%



e : Image codée avec un réseau de Gosset $E_8(1)$ + 16 points statistiques, 1.875 bit/p, taux de compression : 76.56%



f : Image codée avec un réseau de Gosset $E_8(1)$ + 32 points statistiques, 2 bits/pixel, taux de compression : 75%

Fig 4.2 : « a » représente l'image originale, « b » l'image codée par $E_8(1)$ seul, « c » l'image codée par $E_8(1)$ + 4 points statistiques, « d » l'image codée par $E_8(1)$ + 8 points statistiques, « e » l'image codée par $E_8(1)$ + 16 points statistiques, « f » l'image codée par $E_8(1)$ + 32 points statistiques.

4.2. Quantification vectorielle avec dithering (diffusion d'erreurs)

Les quantificateurs à réseaux algébriques, attirent de plus en plus l'attention, vu leur facilité d'implémentation, et vu que dans la plupart des cas ils permettent d'éviter le problème de complexité comme le cas de l'algorithme statistique du L.B.G.

La régularité géométrique des réseaux permet la mise au point d'algorithmes de quantifications très rapides et très efficaces.

La technique du dithering ou de diffusion d'erreur a été appliquée pour la première fois par Roberts [36] pour le codage d'images. En effet, il a été remarqué que le fait d'ajouter une séquence de variables aléatoires indépendantes dite « dither » avant l'opération de quantification, et ensuite la soustraire après quantification, avait un effet appréciable sur la qualité perceptuelle et visuelle de l'image. De cette idée, plusieurs travaux de recherche ont été abordés pour mieux maîtriser cette technique.

Dans notre travail nous avons appliqué cette méthode de diffusion d'erreur sur le réseau de Gosset E_8 . Soit V la matrice génératrice du réseau algébrique considéré, et soit P_0 le Voronoï autour de l'origine 0 du réseau considéré. On note par $Q(P_0, V)$ le

quantificateur algébrique défini par le Voronoï central (autour de l'origine du réseau) P_0 et par la matrice génératrice V [36].

On définit le vecteur d'erreur d'un tel quantificateur par :

$e = x - Q(x)$; ce vecteur d'erreur appartient nécessairement à P_0 . Chaque vecteur d'erreur e est produit par une infinité de vecteur d'entrées de la forme $e + Vn$, n appartenant à Z^k (voir figure 4.1), et l'on peut noter également que ces vecteurs d'erreurs peuvent être eux même des vecteurs d'entrées [36]. La fonction densité de probabilité du vecteur d'erreur est donnée par :

$$f_E(e) = \begin{cases} \sum f_x(e + Vn), & e \in P_0 \\ 0 & \text{ailleurs} \end{cases} \quad (4.1)$$

Le développement de cette fonction en série de Fourier donnera l'expression suivante :

$$\tilde{f}_E(e) = \frac{1}{|\det V|} \cdot \sum \phi_x(Un) \cdot e^{-j e^T U_n} \quad (4.2)$$

avec $\phi_x(Un)$, les coefficients du développement en série de Fourier.

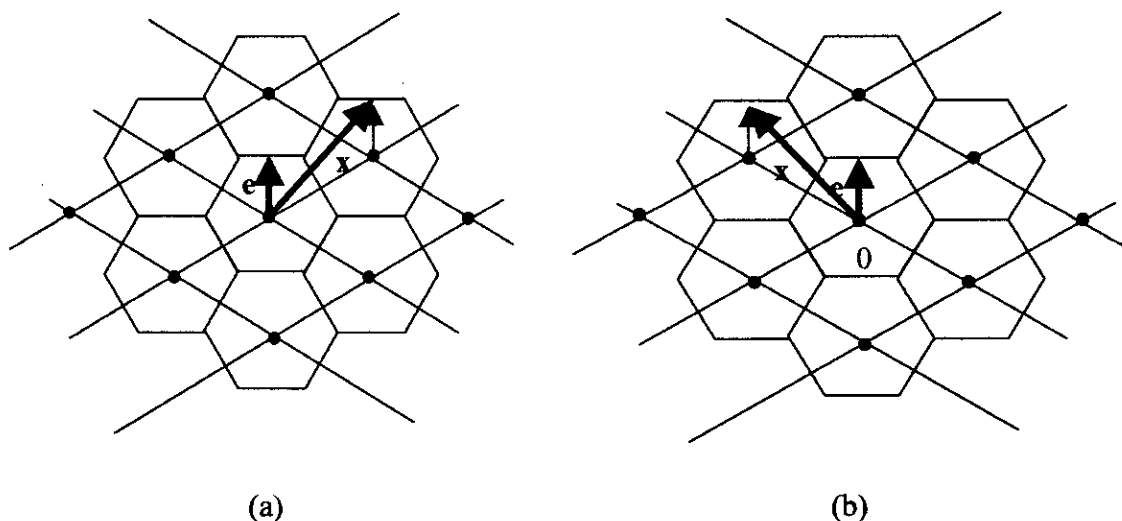


fig4.1 : Erreur de quantification par réseau algébrique.

(a) représente un vecteur d'erreur e produit par le vecteur d'entrée x .

(b) représente un vecteur d'entrée différent produisant le même vecteur d'erreur .

Théorème 1:

L'erreur de quantification d'un réseau $Q(P_0, V)$, est **uniforme sur P_0** , et a une fonction de densité de probabilité définie par :

$$f_E(e) = \begin{cases} \frac{1}{|\det V|}, & e \in P_0 \\ 0 & \text{ailleurs} \end{cases} \quad (4.3)$$

Si et seulement si Le vecteur d'entrée x est **Nyquist-V**, c'est à dire tous les coefficients du développement en série de Fourier de la fonction densité de probabilité des vecteurs d'erreur sont égaux à une impulsion de Dirac.

Théorème 2:

Des vecteurs aléatoires x sont Nyquist-V et ont par conséquent une distribution d'erreur de quantification uniforme dans un réseau régulier de points défini par $Q(P_0, V)$:

1. x est uniforme dans tous Voronoï P du réseau
2. x est uniforme dans toute union arbitraire de Voronoï du réseau, c'est à dire :

$$f_x(x) = \begin{cases} c_i, & x \in P_i \\ 0 & \text{ailleurs} \end{cases} \quad (4.4)$$

$$\text{avec les } c_i \text{ positifs et tel que : } \sum_i c_i = \frac{1}{|\det V|} \quad (4.5)$$

3. x est la somme de vecteurs aléatoires indépendants, où un parmi eux est Nyquist-V .

Démonstration :

1. Soit $Q(P_0, V)$ un quantificateur algébrique avec le Voronoï $P_0 = P$. Alors, $Q(x)=0$, et $e = x$, c'est à dire l'erreur e est uniforme dans P_0 . D'après le théorème 1, si x est Nyquist-V, alors l'erreur de quantification dans P_0 est uniforme même si la quantification est faite par un réseau régulier $Q(P_0, V)$ avec $P_0 \neq P$.

2. On a :

$$\begin{aligned}\phi_x(U_n) &= \int f_x(x) e^{jx^T U_n} dx \\ &= \sum_i \int_{p_i} c_i e^{jx^T U_n} dx \\ &= |\det V| \sum_i c_i \delta(n) \\ &= \delta(n)\end{aligned}$$

3. Soit $x = v + z$, où v et z sont indépendants, alors on pourra écrire :

$$\begin{aligned}\phi_x(\Omega) &= \phi_v(\Omega) \cdot \phi_z(\Omega) \quad \text{on aura donc :} \\ \phi_x(U_n) &= \phi_v(U_n) \cdot \phi_z(U_n) = \delta(n) \quad \text{si : } \phi_v(U_n) = \delta(n) \text{ ou } \phi_z(U_n) = \delta(n)\end{aligned}$$

d'ici on pourra tirer que si v ou z est Nyquist-V alors la somme est également Nyquist-V, il est alors aisé de faire l'extrapolation pour le cas de la somme d'un nombre arbitraire de vecteurs aléatoires.

4.2.1. Dithering par soustraction

Soit v un vecteur aléatoire, statistiquement indépendant du vecteur d'entrée x . On ajoute le vecteur v au vecteur d'entrée x avant l'opération de quantification, puis on le soustrait une fois cette dernière effectuée [36], comme le montre la figure 4.2. Cette façon d'opérer porte le nom de quantification algébrique avec dithering ou avec diffusion d'erreur. Le vecteur d'erreur est donné par : $e = x - (Q(x + v) - v) = x + v - Q(x + v)$, on peut remarquer que cette erreur est la même que celle donnée par le quantificateur conventionnel pour un vecteur d'entrée $x + v$.

La fonction caractéristique de la somme $x + v$ est $\phi_x(\Omega)\phi_v(\Omega)$ où $\phi_x(\Omega)$ et $\phi_v(\Omega)$ représentent respectivement les fonctions caractéristiques de x et de v . Il est alors évident que si l'on se réfère aux théorèmes 1 et 2, pour avoir une distribution d'erreur de quantification uniforme pour une entrée de la forme $x + v$, il suffit d'avoir v ou x Nyquist-V, puisque les vecteurs x sont générés par la source à coder et qui peut avoir une distribution quelconque, en d'autres termes on ne peut pas contrôler les vecteurs générés par la source. On ne peut que contrôler alors que les vecteurs de dither v , qu'on choisira alors Nyquist-V.

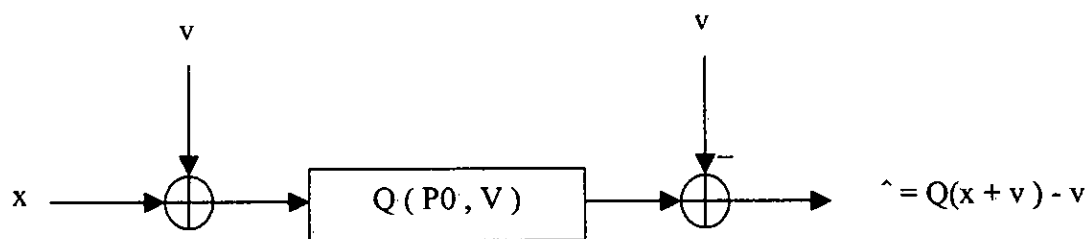


Fig4.2: Quantificateur par réseau régulier avec dithering

2.2. Résultats expérimentaux

La figure 4.3 montre l'amélioration de l'effet psychovisuel et de la qualité perceptuelle de l'image en utilisant comme quantificateur un réseau régulier de points (dans notre cas on a utilisé un réseau de Gosset) avec dithering ou diffusion d'erreur.

La figure (4.3.a) représente l'image originale codée sur 8 bits/pixel, la figure (4.3.b) représente l'image à la sortie du quantificateur algébrique sans dithering, elle est codée sur 1.5 bits/pixel ; la figure (4.3.c) représente l'image à la sortie du quantificateur algébrique avec dithering, elle est toujours codée sur 1.5 bits/pixel ; néanmoins cette dernière image présente une qualité perceptuelle meilleure que celle sans dithering.

Les figures (4.3.d) et (4.3.e) représentent les images d'erreurs respectivement dues à la quantification algébrique sans dithering et avec dithering, on remarque qu'au niveau de cette dernière l'erreur est uniformément distribuée sur toute l'image ce qui améliore l'effet psychovisuel de l'image.



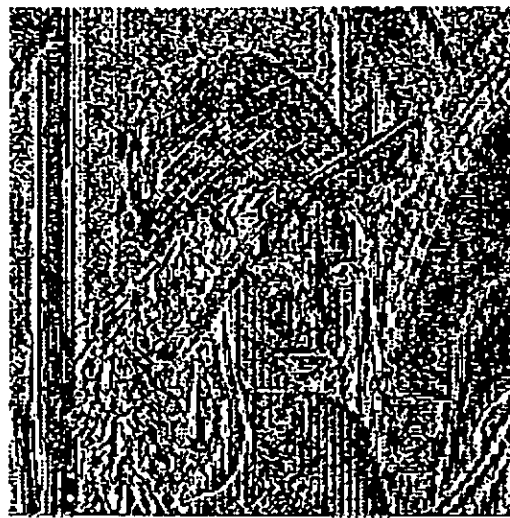
a : Image originale codée sur 8 bits/pixel



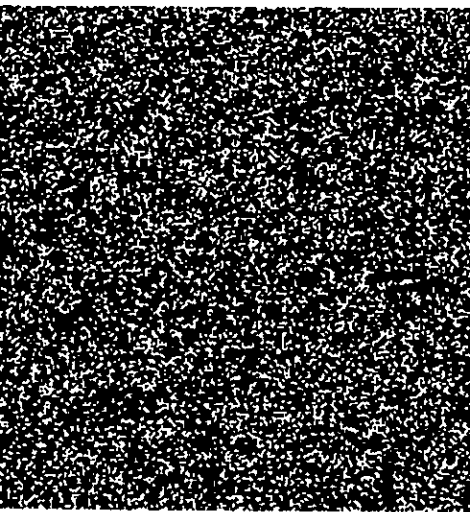
b : Image codée par un réseau de Gosset E8(1) seul, 1.5 bits/pixel, taux de compression 81.25%



c : Image codée par un réseau de Gosset $E_8(1)$ + dithering, 1.5 bits/pixel, taux de compression 81.25%



d: Image d'erreur causée par le codage avec un réseau de Gosset seul



e : Image d'erreur causée par le codage avec un réseau de Gosset + dithering

Fig 4.3 : « a » représente l'image originale, « b » représente l'image codée par le réseau de Gosset $E_8(1)$, « c » représente l'image codée par $E_8(1)$ plus dithering, « d » représente l'erreur causée par le réseau $E_8(1)$ seul et « e » celle causée par $E_8(1)$ plus dithering.

Conclusion générale

Dans ce travail, nous avons proposé deux stratégies pour le codage des signaux informationnels :

- Une approche hybride de quantification vectorielle,
- Une technique de diffusion d'erreurs ou de dithering appliquée à un quantificateur algébrique.

La première stratégie utilisée pour le codage de sources de signaux informationnels, est une nouvelle approche destinée à éviter les problèmes de codage rencontrés avec les approches de quantification ou de codage classique, qui sont les approches statistique et algébrique. Cette nouvelle technique, en fait n'est qu'un compromis ou une technique intermédiaire entre les deux, elle permet donc de tirer leurs avantages et d'éviter leurs inconvénients respectifs. En effet, les résultats obtenus sur des images type le prouve, la qualité visuelle des images obtenues est très proche de celle donnée par les quantificateurs statistiques, malgré le nombre négligeable de vecteurs statistiques utilisés, d'où gain considérable en espace mémoire nécessaire au stockage du dictionnaire, et par conséquent réduction du nombre d'opérations lors de la recherche au sein du dictionnaire, induisant une réduction considérable du temps de calcul et de complexité.

La qualité des images obtenues avec le réseau de Gosset seul est déjà meilleure que celle que pouvait donner un autre quantificateur non sphérique, chose qui est dû au « sphère hardening », en d'autres termes des vecteurs dont les composantes sont issus d'une même source, sont uniformément distribués sur le contour d'une sphère, d'où l'utilisation des réseaux de Gosset comme quantificateur algébrique. La qualité des images devient de plus en plus appréciable quant on augmente le nombre de points statistiques, tout en restant négligeable par rapport au nombre minimum requis par l'approche statistique. Néanmoins, on peut reprocher à cette nouvelle technique le fait qu'elle nécessite un débit binaire dépassant ceux offerts par les méthodes statistiques.

La seconde technique utilisée, vise à améliorer l'effet psychovisuel des images codées par uniquement des réseaux algébriques de points. Cette méthode consiste à diffuser l'erreur sur toute l'image, c'est à dire, elle oblige l'erreur de quantification à être uniformément distribuée sur toute l'image. Pour cela la technique du dithering par soustraction a été appliquée

à un quantificateur algébrique à base du réseau de Gosset ; avant d'injecter le signal à coder à l'entrée du codeur algébrique, un signal aléatoire Nyquist-V lui est ajouté, ce dernier est ensuite soustraie à la sortie du décodeur. Le résultat obtenu offre une meilleure qualité perceptuelle et visuelle à l'image.

Perspectives

Comme il a été précisé le quantificateur hybride offre une alternative aux méthodes classiques, cependant au détriment d'un débit binaire plus élevé. Comme perspective, il serait important d'essayer de trouver un moyen permettant de réduire ce débit tout en gardant toutes les performances d'un tel quantificateur. Parmi les possibilités envisageables, on peut citer :

- Application de la méthode hybride dans le domaine transformé (DCT, FFT, etc....),
- Utilisation de réseaux toujours solutions du problème d'empilement des sphères, mais de dimensions plus grandes tel que le réseau de Leech.

Références bibliographiques

- [1] J. H. Conway et N. J. A. Sloane, "Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization", reprinted from IEEE Trans. Inform. Theory, March 1982.
- [2] A. Gersho et B. Ramamurthi, "Image Coding Using Vector Quantization", reprinted from proc. IEEE Conf. Acoust., Speech, Signal Processing, May 1982.
- [3] Y. Linde, A. Buzo et R.M. Gray, "An Algorithm for Vector Quantizer Design", IEEE Transactions on Communications, January 1980.
- [4] J. H. Conway et N. J. A. Sloane, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes", reprinted from IEEE Trans. Inform. Theory, March 1982.
- [5] T. Murakami, K. Asai et A. Itch, "Vector Quantization of Color Images", reprinted from proc. IEEE Conf. Acoust., Speech, Signal Processing, April 1986.
- [6] V. Ramamoorthy et N. S. Jayant, "High Quality Image Coding With Model-Testing Vector Quantizer and Human Visual System Model", reprinted from IEEE Conf. Acoust., Speech, Signal Processing, April 1988.
- [7] M. Goldberg, P. R. Boucher et S. Shlien, "Image Compression Using Adaptive Vector Quantization", reprinted from IEEE Trans. Commun., Vol. Com-34, pp. 180-187, Feb. 1986.
- [8] A. Gersho, "On the Structure of Vector Quantizers", reprinted from IEEE Trans. Inform. Theory, March 1982.
- [9] K. Aizawa, H. Harashima et H. Miyakawa, "Adaptive Discrete Cosine Transform Coding With Vector Quantization for Color Images", reprinted from proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, April 1986.
- [10] J. Makhoul, S. Roucos et H. Gish, "Vector Quantization in Speech Coding", reprinted from proc. IEEE, Nov. 1985.
- [11] W. Li, "A New Algorithm to Compute The DCT and its Inverse", IEEE Transactions on Signal Processing, June 1991.
- [12] N. Demassieux et F. Jutand, "A real Time Discrete Cosine Transform Chip", L'écho des Recherches-English issue, 1990.
- [13] K. S. Thyagarajan, S. Parthasarathy et H. Abut, "A Matrix Quantizer Incorporating The Human Visual Model", reprinted from proc. IEEE Conf. Acoust., Speech, Signal Processing, March 1985.
- [14] C. W. Rutledge, "Variable Block Vector DPCM: Vector Predictive Coding of Color Images", reprinted from proc. IEEE, June 1987.
- [15] A. A. El Gamal, L. A. Hemachandra, I. Shperling, et Victor K. Wei, "Using Simulated Annealing to Design Good codes", reprinted from IEEE Trans. Inform. Theory, Jan. 1987.
- [16] W. H. Equitz, "A New Vector Quantization Clustering Algorithm", reprinted from proc. IEEE Trans. Acoust., Speech, Signal Processing, Oct. 1989.
- [17] J. Foster, R. M. Gray et M. Ostendorf Dunham, "Finite - State Vector Quantization for Waveform Coding", reprinted from IEEE Trans. Inform. Theory, May 1985.
- [18] M. Goldberg et H. Sun, "Image Sequence Coding Using Vector Quantization", reprinted from IEEE Trans. Commun., July 1986.
- [19] R. L. Baker et R. M. Gray, "Image Compression Using Non-Adaptive Spatial Vector Quantization", reprinted from Conf. Record of 16th Asilomar Conf. Circuits, Systems, Computers. Oct. 1982.
- [20] H-h Shen et R.L. Baker, "A Finite State/Frame Difference Interpolative Vector Quantizer For Low Rate Image Sequence Coding", reprinted from proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, April 1988.
- [21] K. S. Thyagarajan, H. Bohlmann et H. Abut, "Image Coding Based on Segmentation Using Region Growing", reprinted from IEEE Conf. Acoust., Speech, Signal Processing, April 1987.
- [22] A. Gersho, "Adaptive Vector Quantization", Ann, Télécommun., 1986.
- [23] C-C Lu et J.G. Dunham, "Highly Efficient Coding Schemes for Contour Lines Based on Chain Code Representations", IEEE Transactions on Communications, October 1991.
- [24] M.T. Orchard et C.A. Bouman, "Color Quantization of Images", IEEE Transactions on Signal Processing, December 1991.
- [25] G.D Forney, "Coset Codes-Part I: Introduction and Geometrical Classification", IEEE Transactions on Information Theory, September 1988.
- [26] G.D Forney, "Coset Codes-Part II: Binary Lattices and Related Codes", IEEE Transactions on Information Theory, September 1988.
- [27] F. Kuhlmann et J.A. Bucklew, "Piecewise Uniform Vector Quantizers", IEEE Transactions on Information Theory. September 1988.

- [28] K. Sayood, J.D. Gibson et M.C. Rost, "An Algorithm for Uniform Vector Quantizer Design", reprinted from IEEE Trans. Inform. Theory, Nov. 1984.
- [29] T.R. Fisher et R.M. Dicharry, "Vector Quantizer Design for Memoryless Gaussian, Gamma, and Laplacian Sources", reprinted from IEEE Trans. Commun., Sept. 1984.
- [30] T.R. Fisher, "A Pyramid Vector Quantizer", reprinted from IEEE Trans. Inform. Theory, July 1986.
- [31] P.F. Swaszek et Tsu W. Ku, "Asymptotic Performance of Unrestricted Polar Quantizers", reprinted from IEEE Trans. Inform. Theory, March 1986.
- [32] P.A. Chou, T. Lookabaugh et R.M. Gray, "Entropy- Constrained Vector Quantization", reprinted from IEEE Trans. Acoust., Speech, Signal Processing, Jan. 1983.
- [33] J.P. Adoul et M. Barth, "Nearest Neighbor Algorithm for Spherical codes from the Leech Lattice", reprinted from IEEE Trans. Inform. Theory, September 1988.
- [34] N.M. Nasrabadi et R.A. King, "Image Coding Using Vector Quantization", reprinted from IEEE Trans. Commun., Aug.1988.
- [35] J.P Adoul, "La Quantification Vectorielle des signaux: approche algébrique", ANN. Télécommun., 1986.
- [36] A. Kirac et P.P. Vaidyanathan, "Results on Lattice Vector Quantization with Dithering", IEEE Trans. On Circuits and Systems –II: Analog and Digital Signal Processing, December 1996.
- [37] S.A. McClellan et J.D. Gibson, "Group (Source) Coding of The Prediction Residual", Submitted to the IEEE Transactions on Information Theory, February 1991.
- [38] V. Ricordel et C. Labit, "Quantification Vectorielle par emboîtement d'une hiérarchie de réseaux réguliers de points", IRISA, Publication Interne, Juillet 1995.
- [39] N. Sloane, "Les Empilements de spheres", Pour la science, Mars 1984.
- [40] P.F. Swaszek, "A Vector Quantizer for The Laplace Source", IEEE Int'l. Symp. On Information Theory, San Diego, CA, Jan. 1990.
- [41] J. H. Conway et N. J. A. Sloane, "Sphere packings, lattices and groups" by springer – verlag New york Inc.
- [42] N.S. Jayant, "Digital coding of waveform", by Mc Graw Hill edition in 1988.
- [43] M.S Djouadi et D. Berkani, "Image coding using hybrid vector quantization", submitted to COMAEI'98, 8 and 9 December 1998.

Annexe

Introduction des fonctions f et g , utilisées pour l'opération de quantification avec le réseau de Gosset :

Soit $x = (x_1, x_2, \dots, x_k)^T$ un vecteur appartenant à $\mathbb{R}^k$, soit f la fonction qui appliquée au réel x_i nous donne l'entier le plus proche. Dans le cas où x_i est équidistant de deux entiers, f nous fournit l'entier ayant la valeur absolue la plus petite (c.a.d si $x_i = l_i + 0.5 \mid l_i \in \mathbb{Z}$, alors, $f(x_i) = l_i$).

Soit $\delta(x_i) = x_i - f(x_i)$ l'erreur de quantification faite par la fonction f avec $|\delta(x_i)| \leq \frac{1}{2}$

Soit w la fonction, qui appliquée à x_i , donne le second entier le plus proche qu'on appelle communément « wrong way » :

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

Ainsi, étant donné un vecteur x , soit l'entier n ($1 \leq n \leq k$) tel que :

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

x_n représente donc la composante pour laquelle l'erreur de quantification est la plus grande.

Soit g la fonction définit par :

$$g(x) = (f(x_1), f(x_2), \dots, w(x_n), \dots, f(x_k))^T$$

Comparativement à $f(x)$, on a remplacé $f(x_n)$ par $w(x_n)$. Les deux vecteurs $f(x)$ et $g(x)$ diffèrent par une seule composante, et la somme de leurs composantes diffère d'une unité. Le point appartenant à D_k sera alors celui dont la somme des composantes est *paire*.

Pour trouver le points y de D_k le plus proche de x , il faut suivre la procédure suivante :

1. Calcul de $f(x)$, si la somme de ses composantes est paire alors on choisi $y = f(x)$,
2. Sinon, on calcul $g(x)$ et on posera $y = g(x)$.

Dans le cas où l'on trouve $y = f(x)$, 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 où l'on trouve $y = g(x)$, il faut effectuer en tout : $3.k + 2$ opérations : il faut calculer en plus k différences (les $\delta(x_i)$), une recherche de maximum, un arrondi ($w(x_i)$).

Exemples (avec le réseau D_4) :

(1)

$$x = (0.6, -1.1, 1.7, 0.1)^T$$

$$f(x) = (1, -1, 2, 0)^T$$

$$\delta(x) = (-0.4, -0.1, -0.3, 0.1)^T$$

$$g(x) = (0, -1, 2, 0)^T$$

La somme des composantes de $f(x)$ est paire, par contre celle des composantes de $g(x)$ est impaire, donc $y = f(x)$ est le point de D_4 le plus proche de x .

(2)

$$x = (0.5, 0.5, 0.5, 0.5)^T$$

$$f(x) = (0, 0, 0, 0)^T$$

$$g(x) = (1, 0, 0, 0)^T$$

Dans ce cas, on voit directement que $y = g(x)$ est le point de D_4 le plus proche de x .

ملخص: تستعمل الطريقة الإحصائية للتكميم الشعاعي، معاجم أشعة تسمح لتدفق محدد، ترميز مثالي. و لكن هذه المعاجم تخلو من أي بنية، و تتطلب مجالات ضخمة من الذاكرة لتخزينها و وقت كبير لعملية الترميز. إن الشبكات النقطية تسمح بتوليد عدد كبير من النقط من خلال عدد ضئيل من الأشعة، الشيء الذي يمكن أن يؤدي إلى حل مشكلة الطريقة الإحصائية. و لكن للشبكات النقطية مشاكلها الخاصة، بحيث لا تستعمل إلا في حالة المنابع ذات التوزيع المنتظم. من أجل تجنب عوائق الطريقتين، لابد من اقتراح طريقة أخرى. هذه الدراسة مبدؤها إنشاء مكتم حديد يسمى هجين بالتنسيق بين الطريقتين. أولا نعين بعض النقاط الإحصائية باستعمال خوارزمية LBG، ثم كل نقطة توجهنا نحو شبكة Gosset. من أجل رفع النوعية البصرية للصورة المتحصل عليها، نطبق على المكتم المهجن عملية نشر التشويه بالطرح. يمكن القول بأن النتائج المحصل عليها جد مرضية، مثلما تبينته الصور المقدمة.

Abstract Statistic approach of vector quantization uses code-books of rounded vectors which allows quasi-optimal coding for a given rate. Therefore these codebooks have no structure and require big memory size for their storage and a great deal of time for coding. Regular lattices of points gives the possibility of generating a great number of points from a short number of vectors. This can permit us to solve the problems of the statistic approach. However the lattices presents their own problem; effectively they are only applied for uniform distribution sources. In order to avoid these two kinds of problems, another approach is envisaged. This study consists of designing a new quantizer, which we tell hybrid, by combining the two techniques. Using the LBG algorithm performs first few statistic vectors of the source, then each of them leads to a Gosset lattice. In order to increase the perceptual quality of the reconstructed image, a subtractive dithering technique is applied to the hybrid quantizer. Like it could be seen in the presented images, the obtained results are very satisfying.

Résumé La quantification vectorielle par l'approche statistique, utilise des dictionnaires de vecteurs arrondis permettant à un débit donné un codage quasi-optimal. Cependant ces dictionnaires ne présentent aucune structure et nécessitent des espaces mémoires de tailles importantes pour leur stockage et un temps élevé pour le codage. Les réseaux réguliers de points donnent la possibilité de génération d'un grand nombre de points à partir d'un nombre réduit de vecteurs. Ceci peut nous permettre de résoudre le problème de l'approche statistique. Cependant les réseaux de points présentent leur propre problème. En effet, ils sont applicables uniquement dans le cas des sources à distribution uniforme. Pour éviter ces deux types de problèmes, une autre approche est envisagée. Cette étude consiste en le design d'un nouveau quantificateur nommé hybride, par la combinaison des deux techniques. En premier lieu, on détermine quelques points statistiques par l'algorithme du LBG, ensuite chacun de ces points servira d'aiguille vers un réseau de Gosset. Pour améliorer la qualité perceptuelle de l'image reconstruite, on applique au quantificateur hybride la technique de diffusion d'erreur par soustraction. Comme peuvent le montrer les images présentées, les résultats obtenus sont très satisfaisants.