

Nouvelle méthode de codage par plan de bits pour les signaux de parole et audio

Thi Minh Nguyet Hoang¹

Stéphane Ragot¹

Marie Oger[‡]

Marc Antonini²

¹ Orange Labs/TECH/SSTP, Av. Pierre Marzin, 22307 Lannion Cedex
Tel: 33 2 96 05 34 58, Fax: 33 2 96 05 35 30

{thiminhnguyet.hoang, stephane.ragot}@orange-ftgroup.com

² Lab. I3S-UMR 6070 CNRS et Université de Nice Sophia-Antipolis,
2000 rte des Lucioles, F-06903 Sophia Antipolis
Tel: 33 4 92 94 27 18, Fax: 33 4 92 94 28 98

am@i3s.unice.fr

Résumé

Une nouvelle technique de codage par plan de bits pour le codage par transformée des signaux de parole et audio est proposée. Cette technique décompose une séquence entière à coder en une succession de plans de bits, des bits les plus significatifs (MSB) aux bits les moins significatifs (LSB). Chaque plan de bits est ensuite converti en une séquence quinaire (+, -, 0, 1, EoP), où le symbole "EoP" (End of Plane) indique la fin du plan courant. Un codage arithmétique contextuel est finalement appliqué sur cette séquence quinaire. Pour exploiter la corrélation entre plans de bits successifs, les plans ne sont pas codés de façon séquentielle (du premier bit au dernier bit), mais en deux passes, en fonction des plans précédemment codés. Une application aux signaux de parole et audio en bande élargie, échantillonnés à 16 kHz est présentée. Les résultats montrent que la technique proposée est équivalente – en terme de performance/complexité – au codage non scalable de type stack-run, tout en permettant un train binaire scalable.

Mots clefs

Codage audio, codage de parole, codage entropique.

1 Introduction

Avec l'évolution des technologies et des services de communications, les réseaux de communication et les terminaux sont devenus très hétérogènes. Par ailleurs, il existe de très nombreux systèmes de codage de parole et audio normalisés, chacun étant adapté à des applications ou réseaux spécifiques. Ces standards sont en général peu flexibles et ne s'adaptent pas facilement à des contraintes diverses (ex : fréquence d'échantillonnage, taille de trame, qualité, débit, retard, complexité, taux de perte des paquets, etc.).

Cette problématique de l'hétérogénéité et du manque de flexibilité est étudiée par le projet européen IST FP6 Flex-Code (*Flexible Coding for Heterogenous Networks*) [1]. L'objectif de Flexcode est de créer un codeur source-canal flexible, optimal et pouvant s'adapter instantanément à différentes contraintes. Ces objectifs sont atteints en utilisant des outils de codage flexible comme la quantification scalaire suivie par un codage arithmétique, et surtout en s'appuyant sur une *approche de codage basé modèle* [2, 3], où le codage est optimisé à partir de paramètres issus de *modèles génériques* de source, canal et perception. Des exemples de codage basé modèle sont le codage des coefficients LPC (*Linear Predictive Coding*) par GMM (*Gaussian Mixture Models*) [2] ou le codage par transformée à partir d'un modèle gaussien généralisé [3]. La théorie de débit-distorsion asymptotique en débit (*high-rate distortion theory*) sert à optimiser analytiquement les paramètres de codage de source.

Dans cet article, on propose une nouvelle technique de codage par plan de bits qui rentre dans le cadre du projet Flex-Code. Le codage par plans de bits consiste à décomposer une séquence entière sous forme binaire, du bit le plus significatif (MSB) au bit le moins significatif (LSB), et à appliquer un codage entropique à chaque plan de bits. Le train binaire obtenu est par nature scalable (ou *hiérarchique*) [4]. Le codage par plan de bits est déjà utilisé en codage audio (ex. MPEG-4 BSAC [5], MPEG-4 SLS [6], codeurs audio propriétaires [7, 8]), codage d'image (ex. JPEG2000 [9]) et codage vidéo (ex. MPEG-4 FGS [10]). Ici, on se concentre sur le codage de parole et audio pour les applications conversationnelles. Des résultats ont déjà été présentés dans [11], avec un codage par plan de bits est basé sur un modèle de probabilité de la source gaussienne généralisée. On étend ces travaux en proposant une nouvelle méthode flexible de codage par plan de bits. Cette méthode est plus efficace et de plus faible complexité que celle décrite dans [11].

Le principe de codage type stack-run et de codage par plan de bits qui sont deux méthodes de codage entropique est

*Marie Oger (marie.oger.sa@gmail.com) était à Orange Labs quand ces travaux ont débuté.

†Ce travail a été en partie financé par le projet européen Flexcode (FP6-2002-IST-C 020023-2), auquel France Télécom/Orange participe.

présenté à la section 2. On propose une nouvelle technique de codage par plan de bits à la section 3. Son application au codage par transformée des signaux de parole et audio est illustrée à la section 4. Les performances obtenues y sont comparées avec celles de codeurs de référence (codage stack-run, codage par plan de bits basé modèle) avant de conclure dans la section 5.

2 Codage entropique de séquences entières

On considère ici le problème du codage entropique d'une séquence à valeurs entières signées $\mathbf{Y} = [y_1 \dots y_N]$ de dimension $N \geq 1$. Cette séquence est par exemple issue de la quantification scalaire d'un signal $\mathbf{X} = [x_1 \dots x_N]$. Dans ce qui suit on présente deux méthodes flexibles de codage entropique. D'autres méthodes telles que le codage de Golomb ou Golomb-Rice [12] sont envisageables.

2.1 Codage stack-run

Le codage stack-run est une technique de codage sans perte de séquences entières développée pour le codage d'images par ondelettes [13]. Il consiste à convertir une séquence d'entiers en une séquence à alphabet quaternaire (0, 1, +, -) et ensuite à coder cette séquence par un codage arithmétique contextuel [14].

La figure 1 (a) présente le principe du codage stack-run. Les éléments de la séquence \mathbf{Y} sont partitionnés en deux contextes : *run* et *stack*. Un *run* est une série de zéros, dont la longueur est codée de la même façon qu'en codage par plage (*run-length coding*). Un *stack* est un entier non nul. Un *stack* correspond à une colonne de bit du MSB au LSB. Le principe de la conversion en symbole quaternaire (0, 1, +, -) est détaillé dans [13]. On en résume ici les règles. Les symboles "+" et "-" sont utilisés pour le signe d'un stack et la longueur des runs exprimée en binaire, tandis que "0" et "1" sont utilisés pour la valeur absolue des stacks. Le MSB de chaque stack est remplacé par un "+" quand le coefficient associé est positif et "-" dans le cas contraire.

Les figures 1 (b) et (c) présentent un exemple de conversion d'une séquence entière

$$\mathbf{Y} = [0, 0, 0, +35, +4, 0, 0, 0, 0, 0, 0, 0, 0, 0, -11] \quad (1)$$

en une séquence quaternaire (0, 1, +, -). La séquence \mathbf{Y} commence par un run de 3 zéros ; la représentation binaire de 3 étant "11", on obtient le code "++". La longueur du run étant égale à $2^k - 1$, le dernier "+" (MSB) n'est pas supprimé [13]. L'élément suivant est un stack de valeur +35, dont la valeur absolue est incrémentée [13] pour obtenir 36 ; la représentation de 36 est "001001" et donc le code associé à +35 est "00100+". Les mêmes opérations sont effectuées par la suite. Ainsi, les 16 éléments entiers de \mathbf{Y} sont représentés par 18 symboles quaternaires "++00100+10+-+0001-".

Un exemple de codage basé modèle utilisant le codage stack-run pour les signaux en bande élargie (échantillonnés à 16 kHz) est présenté dans [3].

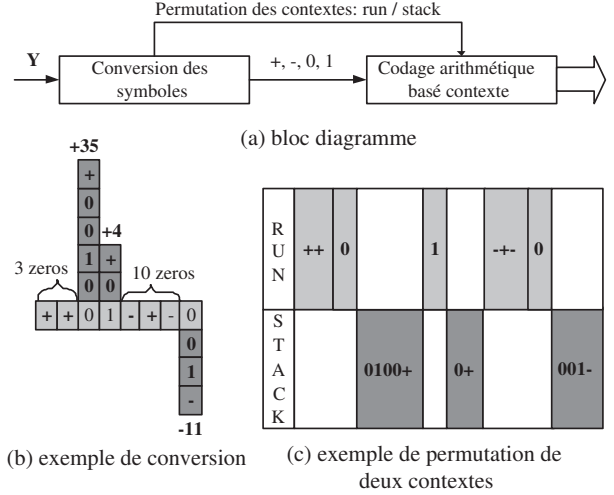


FIG. 1 – Codage stack-run : principe et exemple.

2.2 Codage par plan de bits

La décomposition en plans de bits d'une séquence entière \mathbf{Y} revient d'abord à séparer signes et valeurs absolues. Cette décomposition peut être représentée sous forme d'une matrice binaire $(K + 1) \times N$:

$$\mathbf{P}(\mathbf{Y}) = \begin{bmatrix} s_1 & \dots & s_N \\ b_{K-1}(y_1) & \dots & b_{K-1}(y_N) \\ \vdots & & \vdots \\ b_0(y_1) & \dots & b_0(y_N) \end{bmatrix} = \begin{bmatrix} \mathbf{S}(\mathbf{Y}) \\ \mathbf{P}_{K-1}(\mathbf{Y}) \\ \vdots \\ \mathbf{P}_0(\mathbf{Y}) \end{bmatrix} \quad (2)$$

où K est le nombre de plans de bits maximal pour la décomposition de l'ensemble des $|y_i|$, s_i est le bit de signe de y_i et $b_k(y_i)$ est le $k^{i\text{ème}}$ bit de la décomposition binaire de $|y_i|$. On se restreint ici à la décomposition linéaire naturelle, cependant d'autres représentations comme le codage de Gray sont possibles.

Le codage s'effectue sur les plans de bits $\mathbf{P}_k(\mathbf{Y})$ ($0 \leq k < K$) successifs, en allant du MSB ($k = K - 1$) au LSB ($k = 0$). En général, le nombre total de plan de bits K doit être codé et transmis. Pour ne pas transmettre de bits de signe inutiles, les bits de signe s_i dans le plan $\mathbf{S}(\mathbf{Y})$ ne sont transmis que si la valeur absolue correspondante $|y_i|$ est non nulle. Pour permettre un décodage partiel (dit aussi *hiérarchique*) des plans de bits, le bit de signe s_i doit être transmis dès qu'un des bits décodés $b_k(y_i)$ vaut 1. Les valeurs y_i non nulles sont dites *significatives*.

Un exemple de décomposition en plan de bits, pour la

séquence \mathbf{Y} de l'équation 1, est donné ci-dessous :

$$\mathbf{P}(\mathbf{Y}) = \begin{bmatrix} 0000000000000001 \\ 0001000000000000 \\ 0000000000000000 \\ 0000000000000001 \\ 0000100000000000 \\ 0001000000000001 \\ 0001000000000001 \end{bmatrix} = \begin{bmatrix} \mathbf{S}(\mathbf{Y}) \\ \mathbf{P}_5(\mathbf{Y}) \\ \mathbf{P}_4(\mathbf{Y}) \\ \mathbf{P}_3(\mathbf{Y}) \\ \mathbf{P}_2(\mathbf{Y}) \\ \mathbf{P}_1(\mathbf{Y}) \\ \mathbf{P}_0(\mathbf{Y}) \end{bmatrix} \quad (3)$$

Dans ce cas, il y a $K = 6$ plans. Les 16 éléments entiers de \mathbf{Y} sont représentés avec 99 bits au total (16×6 bits pour les valeurs absolues + 3 bits de signe significatifs). Comparée au codage stack-run, la représentation par plan de bits est plus redondante. La technique décrite à la section 3 a pour but de rendre cette représentation plus compacte pour atteindre une efficacité de codage proche du codage stack-run.

3 Nouvelle méthode de codage par plan de bits

Les notations introduites à la section 2 sont réutilisées dans cette section. La technique de codage par plan de bits proposée ici est fondée sur quelques observations :

- En codage de signaux de parole et audio, en supposant que la séquence entière \mathbf{Y} correspond à un spectre discret, l'ordre naturel de codage des bits $b_k(y_i)$ dans chaque plan $\mathbf{P}_k(\mathbf{Y})$ est séquentielle, de $b_k(y_1)$ à $b_k(y_N)$, des basses fréquences aux hautes fréquences. En théorie, l'ordre de codage des bits $b_k(y_i)$ doit cependant respecter l'importance perceptuelle des bits [8].
- Les travaux publiés [9, 11] montrent que les performances obtenues avec le codage par plans de bits sont en général inférieures à celle du codage non hiérarchique. Cette pénalité est due en grande partie à la difficulté d'exploiter l'information mutuelle entre plans de bits successifs.
- Les séries de zéros dans la séquence d'entiers \mathbf{Y} produisent des colonnes de bits nulles dans $\mathbf{P}(\mathbf{Y})$. Alors que le codage stack-run représente ces séries de zéros sous forme d'un codage de plage avec les symboles “+” ou “-”, le codage par plan de bits répète le codage de séries de zéros pour chaque plan de bits.
- Si la séquence d'entier \mathbf{Y} représente des coefficients fréquentiels d'un signal audio, en général ce spectre concentre l'énergie en basse fréquence plus qu'en haute fréquence. Ainsi, pour les plans de bits les plus significatifs (k est proche de $K - 1$) la probabilité, $P[b_k(y_j) = 0] \approx 1$ à la fin de chaque plan.

Ces observations ont motivé l'emploi d'un codage en plusieurs passes (pour exploiter efficacement les corrélations entre plans) et l'introduction d'un symbole spécial (“*EoP*”), comme décrit ci-après.

3.1 Représentation en 5 symboles et codage en une passe du MSB

Dans la technique proposée, le plan MSB est converti en 5 symboles : “+”, “-”, “0”, “1” et “*EoP*” d'une façon similaire au codage stack-run mais en ajoutant un symbole spécial “*EoP*” et en tenant compte de la nature binaire “signée” du plan MSB. On cherche d'abord la position p_k du symbole “*EoP*” dans chaque plan $\mathbf{P}_k(\mathbf{Y})$. Celle-ci est donnée par :

$$p_k = \arg \min_{1 \leq i \leq nb} \{ \forall j \geq i \ b_k(y_j) = 0 \} \quad (4)$$

Ainsi, le symbole “*EoP*” indique que tous les bits $b_k(y_j)$ dans le plan $\mathbf{P}_k(\mathbf{Y})$ sont nuls pour $j \geq p_k$. Donc, on code les bits s'arrêtant à cette position : $[b_k(y_1) \dots b_k(p_k)]$.

Dans le plan MSB $\mathbf{P}_{K-1}(\mathbf{Y})$, les suites consécutives de “0” (à part celle terminant éventuellement le plan) sont converties en longueurs de plage (runs) sous la forme de “+” et “-” : le nombre de “0” consécutifs est calculé, écrit en binaire, 0 est remplacé par “-” tandis que 1 est remplacé par “+”. Comme en codage stack-run, si le nombre de zéros consécutifs n'est pas de la forme $2^n - 1$ (n entier), le premier bit de cette décomposition binaire (correspondant forcément à un “+” à la fin de décomposition) n'est pas pris en compte [13]. Un bit “1” dans $\mathbf{P}_{K-1}(\mathbf{Y})$ est remplacé par un symbole “0” ou “1” suivant que la valeur entière associée à la position du bit de valeur “1” est respectivement positive ou négative.

A titre d'exemple, le MSB dans la décomposition en plan de bits donné à l'équation 3 est : $\mathbf{P}_5(\mathbf{Y}) = [0001000000000000]$. Ce plan est converti en séquence quinaire $[+ + 0 \text{ EoP}]$, où “++” représente les trois premiers zéros du plan (en retenant que 3 vérifie $3 = 2^2 - 1$) [13]; “0” représente la valeur 1 du bit de poids fort de la composante +34, laquelle est positive et le dernier symbole “*EoP*” représente la série de zéros en fin de plan.

3.2 Codage des autres plans en deux passes

Pour exploiter la corrélation entre plans de bits successifs, les plans $\mathbf{P}_k(\mathbf{Y})$ ($k < K - 1$), sont codés en deux passes, en fonction des plans précédemment codés $\mathbf{P}_{K-1}(\mathbf{Y}), \dots, \mathbf{P}_{k+1}(\mathbf{Y})$.

La première passe commence par une recherche des positions significatives dans le plan en cours et à coder les bits associés à ces positions. La deuxième passe code les bits restants. Ainsi, pour un plan $\mathbf{P}_k(\mathbf{Y})$ ($k < K - 1$), on calcule un indicateur $significant_k(i)$ défini par :

$$significant_k(i) = \left\lfloor \frac{|y_i|}{2^{k+1}} \right\rfloor = \sum_{m=k+1}^{K-1} b_m(y_i) 2^m \quad (5)$$

où $\lfloor \cdot \rfloor$ correspond à l'arrondi à l'entier inférieur. La valeur de $significant_k(i)$ calculée pour le plan $\mathbf{P}_k(\mathbf{Y})$ est utilisée pour diviser ce plan en deux parties : une partie

“significative”

$$\mathbf{P}_k^{sig}(\mathbf{Y}) = \{b_k(y_i), i \leq 1 \leq N | significant_k(i) \neq 0\} \quad (6)$$

et une partie “non significative”

$$\mathbf{P}_k^{nonsig}(\mathbf{Y}) = \{b_k(y_i), i \leq 1 \leq N | significant_k(i) = 0\} \quad (7)$$

Dans la première passe, on code $\mathbf{P}_k^{sig}(\mathbf{Y})$ sur deux symboles (“0” et “1”). Dans la seconde passe, $\mathbf{P}_k^{nonsig}(\mathbf{Y})$ est converti en une séquence à 5 symboles (“+”, “-”, “0”, “1” et “*EoP*”) puis codée en arithmétique. La conversion en 5 symboles est la même que pour le MSB (voir section 3.1).

k	$\mathbf{P}_k^{sig}(\mathbf{Y})$	$\mathbf{P}_k^{nonsig}(\mathbf{Y})$	1 ^{ère} passe	2 ^{ème} passe
5		0001000000000000		+ + 0 <i>EoP</i>
4	0	0000000000000000	0	<i>EoP</i>
3	0	0000000000000001	0	- + +1
2	00	0001000000000000	00	+ + 0 <i>EoP</i>
1	101	0000000000000000	101	<i>EoP</i>
0	101	0000000000000000	101	<i>EoP</i>

Un exemple pour la décomposition en plan de bits de l'équation 3 est montré ci-dessus. Le plan MSB est converti avec les règles définies dans la section 3.1 en une séquence [+ + 0 *EoP*]. Les plans suivants sont codés en deux passes. Dans la première passe, on cherche d'abord la position du ou des bits dont la valeur est de “1” dans le plan précédent. Dans l'exemple, il s'agit du seul quatrième bit du plan $\mathbf{P}_4(\mathbf{Y})$. Sa valeur est “0”, donc $\mathbf{P}_4^{sig}(\mathbf{Y}) = [0]$ et on code un seul symbole “0”. Pour les bits restants $\mathbf{P}_4^{nonsig}(\mathbf{Y}) = [0000000000000000]$, qui deviennent en représentation quinaire le symbole “*EoP*” et on code ce symbole. Pour le plan $\mathbf{P}_3(\mathbf{Y})$, on a encore le quatrième bit à comptabiliser pour la première passe et sa valeur est “0”, donc $\mathbf{P}_3^{sig}(\mathbf{Y}) = [0]$ et on code un seul bit “0”. Dans la deuxième passe, on a les bits restants $\mathbf{P}_3^{nonsig}(\mathbf{Y}) = [0000000000000001]$, ce qui correspond à 14 zéros, soit “-+++” en représentation quinaire, toutefois, le dernier symbole “+” n'a pas besoin d'être codé et il reste finalement “-++”. De plus, le bit de valeur 1 est remplacé par le bit de signe négatif “1”, correspondant $y_i = -11$. On notera ici l'absence du symbole “*EoP*” car ce plan se termine par un “1”. Il en va de même pour les autres plans. En conséquence, 24 symboles sont utilisés pour représenter la séquence d'entiers \mathbf{Y} .

4 Exemple d'application

La figure 2 présente le schéma bloc d'un codeur prédictif par transformée de signaux de parole et audio (mono). Le signal d'entrée $x(n)$ est échantillonné à 16 kHz. Il est découpé en blocs successifs (ou trames) avec une longueur de trame est de 20 ms et une mémorisation du signal futur *lookahead* de 25 ms. Le retard algorithmique est donc de 45 ms.

L'entrée $x(n)$ est traitée par un filtrage passe-haut (HP) afin de retirer les fréquences inférieures à 50 Hz. Le signal est

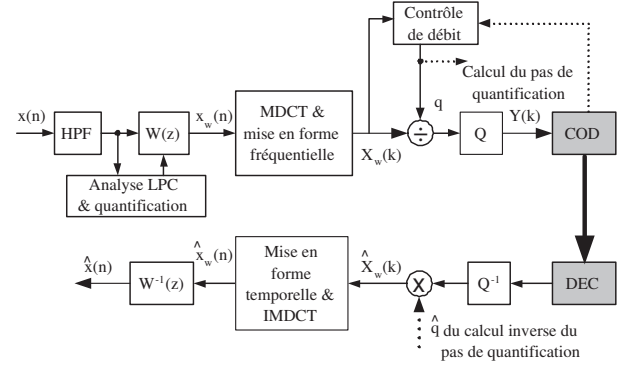


FIG. 2 – Architecture de codage prédictif par transformée.

ensuite pondéré perceptuellement par filtrage $W(z)$, dont les coefficients sont adaptés au moyen d'une analyse LPC comme décrit dans [3]. Le signal temporel pondéré $x_w(n)$ est transformé en spectre fréquentiel $X_w(k)$ par analyse MDCT et mis en forme fréquentiellement [3]. Ce spectre $X_w(k)$ est quantifié en scalaire, avec un pas de quantification q . La séquence d'entiers $Y(k)$ résultante est codée soit par codage stack-run, soit par codage par plan de bits. L'allocation des bits aux paramètres de codage (coefficients LPC pour $W(z)$, pas de quantification q , spectre \mathbf{Y}) est identique à [11] mis à part qu'aucun bit n'est réservé ici pour l'injection de bruit : 43 bits pour les paramètres LPC, 7 bits pour le pas de quantification, les bits restants sont alloués au codage entropique.

Le schéma de la figure sépare la perception (masquage fréquentiel) et le codage. En effet le filtrage $W(z)$ et la mise en forme fréquentielle après MDCT servent à conditionner le signal d'entrée pour le mettre dans un domaine où s'applique l'erreur quadratique. L'utilisation d'un seul pas de quantification dans le domaine MDCT suit le principe de remplissage inverse des eaux (*reverse waterfilling*) [15].

Deux types de codage sont testés :

- Le codage à débit fixe (*constrained resolution*) – dans ce cas le pas de quantification q est adapté pour assurer la contrainte de débit fixe.
- Le codage à qualité constante, qui donne un débit variable – sous l'hypothèse que la pondération perceptuelle par le filtre $W(z)$ est optimale et permet d'appliquer le critère d'erreur quadratique dans le domaine du spectre ($X_w(k)$), on fixe le pas q de telle sorte que le rapport signal à bruit dans le domaine MDCT soit constant par trame.

4.1 Résultats pour le codage à débit fixe

La mesure de qualité objective WB-PESQ [16] a été utilisée pour comparer les performances de codage et évaluer l'influence du codage entropique. Seuls des échantillons de parole ont été traités pour calculer la note moyenne MOS-LQO (*Mean Opinion Score - Listening Quality Objective*) à différents débits. Le débit varie de 16 à 40 kbit/s avec un

pas de 4 kbit/s. La figure 3 présente les notes WB-PESQ obtenues pour le codeur de la figure 2 où COD et DEC (blocs grisés) correspondent au codage stack-run [3], codage par plan de bits basé modèle [11], ou à la technique proposée ici. Mis à part le codage de référence G.722.1, tous les codeurs testés partagent le même schéma de codage ; la différence se trouve dans la méthode du codage entropique. Pour comparer sans biais l'influence du codage entropique, les améliorations telles que la réduction de pré-écho ou l'injection/substitution du bruit ne sont pas utilisées. Les notes du G.722.1 sont données pour l'information, même si WB-PESQ n'est pas recommandé pour comparer les différents modèles de codage.

La figure 3 confirme les résultats de [11] où il a été déjà montré que la performance du codage par plan de bits basé modèle est très proche du codage stack-run, sauf à bas débit (ex. 16 kbit/s) où il y a un écart environ 0.2 MOS-LQO. La nouvelle méthode proposée ici réduit cet écart à 0.1 MOS-LQO à bas débit. La différence minimale entre le codage stack-run et la méthode proposée peut être expliquée à partir des exemples donnés précédemment dans les sections 2 et 3, où la séquence \mathbf{Y} donnée à l'équation 1 est représentée par 18 et 24 symboles quinaires respectivement.

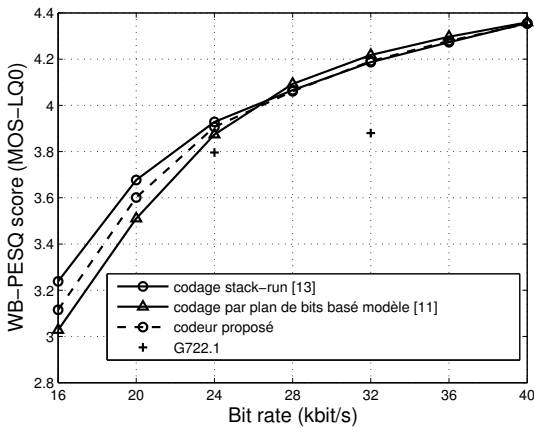


FIG. 3 – Notes WB-PESQ pour le codage à contrainte de débit fixe (sans injection de bruit).

Des tests subjectifs informels pour des signaux de parole et de musique codés à 24, 32, 40 kbit/s ont été effectués par 4 sujets experts. Quatre signaux de test, échantillonnés à 16 kHz ont été utilisés : deux échantillons de parole (une voix d'homme, une voix de femme) et deux échantillons de musique (piano, trompette). Les résultats pseudo-MUSHRA* ont montré que la qualité obtenue avec la méthode proposée ou le codage stack-run est équivalente à 32 et 40

*Les tests subjectifs ont été réalisés en suivant la définition de la méthodologie MUSHRA [17], mais avec un nombre limité de conditions, sans "ancres" (ex. référence filtrée à 3.5 kHz). Les sujets ont pu écouter la référence échantillonnée à 16 kHz, et noter 9 conditions sur une échelle de 0 à 100. Ces 9 conditions ont été présentées dans un ordre aléatoire : référence cachée, signal codé par stack-run à 24, 32, 40 kbit/s, signal codé par G.722 à 24, 32 kbit/s et signal codé par la méthode proposée à 24, 32, 40 kbit/s.

kbit/s, ce qui n'est pas le cas à 24 kbit/s, où légèrement plus d'artefacts ont été notés dans le cas du codage par plan de bits. Il est donc nécessaire par la suite d'améliorer la qualité à bas débit en étudiant des aspects tels que la réduction de pré-écho et l'injection de bruit, pour masquer le bruit musical et le pré-écho à 16 et 24 kbit/s.

4.2 Résultats pour le codage à qualité constante

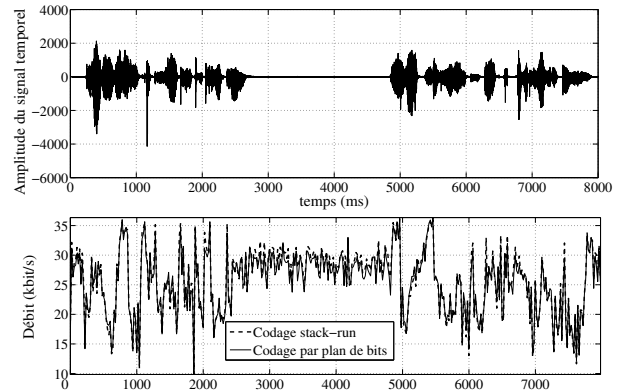


FIG. 4 – Codage à qualité constante (sans injection de bruit).

La figure 4 présente l'évolution du débit (en kbit/s) utilisé par une trame de 20 ms lorsque le codage effectué est pour une "qualité" (objective) constante. Cette figure a été obtenue avec un signal de parole (une voix femme) $x(n)$. Ces résultats montrent en terme de débit que le codage stack-run et le codage par plan de bits sont quasiment équivalents. La différence de débit entre ces deux codeurs n'est pas significative, ce qui valide l'intérêt du codage par plans de bits proposé qui est plus flexible et donne un train binaire scalable. A la figure 4 on observe que le silence entre les deux phrases de parole est codée à un débit proche de 30 kbit/s, ce qui indique que le critère de qualité (rapport signal sur bruit par trame) n'est peut-être pas le plus adapté d'un point de vue perceptuel.

4.3 Remarques sur la complexité

La technique proposée de codage par plan de bits n'exige aucun stockage de tables de codage statiques (en ROM), comme c'est le cas pour le codage stack-run [3] ou le codage par plan de bit basé modèle [11]. Elle a une complexité du calcul équivalente à celle du codage stack-run, ce qui la différencie du codage par plan de bits basé modèle [11] qui requiert une estimation de tables de probabilité par calcul des intégrales. La méthode proposée n'implique pas de tels calculs d'intégrales.

A noter que la nature scalable du codage par plan de bits peut être exploitée pour réduire la complexité du codeur en fixant un pas de quantification q et en utilisant un codage partiel (fractionnel) des plans de bits.

5 Conclusion

Dans cet article une nouvelle méthode de codage par plan de bits a été proposée. Celle-ci utilise un codage arithmétique en deux passes d'une séquence quinaire (+, -, 0, 1, *EoP*), où le symbole spécial "EoP" pour *End of Plane*, s'il est utilisé, indique la fin de chaque plan courant. L'application au codage prédictif par transformée des signaux de parole et audio échantillonnés à 16 kHz a été illustrée. La technique proposée ici a été comparée au codage stack-run [3] et au codage par plan de bit basé modèle [11]. Les résultats objectifs ont montré que la qualité audio de 16 à 40 kbit/s est très proche du codage non scalable (codage stack-run). Les résultats subjectifs de la qualité confirment ces conclusions à 32 et 40 kbit/s. Cependant, à 24 kbit/s la technique proposée ici donne une qualité inférieure au codage stack-run.

Des aspects complémentaires tels que l'injection de bruit ou la réduction de pré-écho doivent être étudiés pour obtenir une bonne qualité à bas débit (16-24 kbit/s). Cet article s'est en effet concentré principalement sur l'efficacité de codage entropique.

A noter que la méthode proposée pourrait être appliquée au codage d'image ou vidéo, même si certains aspects tels que l'introduction d'un symbole "End of Plane" sont optimisés pour le codage par transformée des signaux audio. De plus il est possible d'estimer la probabilité des symboles dans chaque plan pour rendre la technique proposée basée modèle et potentiellement améliorer les performances de codage, la complexité sera cependant plus élevée.

Références

- [1] S. Bruhn, V. Grancharov, B. Kleijn, J. Klejsa, M. Li, H. Pobloth, et S. Ragot. The FlexCode speech and audio coding approach. *8th ITG Conference Speech Communication*, Oct. 2008.
- [2] A.D. Subramaniam et B.D. Rao. PDF optimized parametric vector quantization of speech line spectral frequencies. *IEEE Trans. Speech and Audio Proc.*, 11(2) :130–142, Mar. 2003.
- [3] M. Oger, S. Ragot, et M. Antonini. Transform audio coding with arithmetic-coded scalar quantization and model-based bit allocation. Dans *Proc. ICASSP*, volume 4, pages 545–548, 2007.
- [4] B. Geiser, S. Ragot, et H. Taddei. Embedded Speech Coding : From G.711 to G.729.1. Dans *Advances in Digital Speech Transmission (R. Martin, U. Heute, C. Antweiler, eds.)*, chapitre 8, pages 201–248. Wiley, Jan. 2008.
- [5] S.H. Park et al. Multi-layer bit-sliced bitrate scalable audio coding. Dans *AES 103rd Convention*, Aug. 1997.
- [6] R. Yu et al. MPEG-4 scalable to lossless audio coding. Dans *AES 117th convention*, 2004.
- [7] C. Dunn. Efficient audio coding with fine-grain scalability. Dans *AES 111th convention*, Sept. 2001.
- [8] J. Li. Embedded audio coding (EAC) with implicit auditory masking. *ACM Multimedia 2002*, Dec. 2002.
- [9] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Trans. on Image Proc.*, 9 :1158–1170, July 2000.
- [10] H. Radha, M.V.D. Schaar, et Y. Chen. The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP. *IEEE Trans. Multimedia*, 3, Mar. 2001.
- [11] T.M.N. Hoang, M. Oger, S. Ragot, et M. Antonini. Embedded transform coding of audio signals by model-based bit plane coding. Dans *Proc. ICASSP*, pages 4013–4016, Apr 2008.
- [12] K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, 2005.
- [13] M.J. Tsai, J.D. Villasenor, et F. Chen. Stack-run image coding. *IEEE Trans. Circuits Syst. Video Techno.*, 6 :519–521, October 1996.
- [14] I.H. Witten, R.M. Neal, et J.G. Cleary. Arithmetic coding for data compression. *ACM Communications*, June 1987.
- [15] A.N. Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Tran. Inform. Theory*, IT2 :102–108, 1956.
- [16] ITU-T Rec P.862.2. *Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs*. Nov 2005.
- [17] ITU-R Rec. BS.1534. *Method for the subjective assessment of intermediate quality levels of coding systems*. Jan. 2003.