

# Codage de sources distribuées : comparaison de 2 systèmes approchant la borne de Wyner-Ziv

M. Jeanne<sup>1</sup>

R. Zhang<sup>2\*</sup>

B. Girod<sup>2</sup>

P. Siohan<sup>3</sup>

<sup>1</sup> France Télécom R&D, 4 rue du Clos Courtel, B.P. 59, 35512 Cesson-Sévigné Cedex, FRANCE

<sup>2</sup> Information System Laboratory, Stanford University, Stanford, CA 94305, USA

<sup>3</sup> IRISA Rennes, Campus universitaire de Beaulieu, 35042 Rennes Cedex, FRANCE

marion.jeanne@rd.francetelecom.com, Rui.Zhang@am.sony.com,  
bgirod@stanford.edu, pierre.siohan@irisa.fr

## Résumé

Cet article porte sur l'étude du codage de sources distribuées, i.e. la compression de sources corrélées qui sont codées indépendamment mais décodées conjointement. Dans la littérature les propositions de mise en œuvre pour approcher la limite théorique de Wyner et Ziv utilisent des techniques de codage de canal. Ici nous proposons une solution réalisée avec une technique de codage source : le codage à longueur variable (CLV). Cette dernière approche consiste à appliquer la méthode de Al Jabri et Al-Issa [1] au contexte des sources distribuées. L'étude est complétée par une comparaison avec la méthode DISCUS de Pradhan et Ramchandran [2, 3]. Dans la plupart des cas, DISCUS donne de meilleurs résultats, mais l'utilisation du CLV permet une plus grande flexibilité au niveau du taux de compression.

## Mots clefs

Codage de source distribuées, codes à longueur variables.

## 1 Introduction

Considérons un système de communication ayant à transmettre deux signaux  $X$  et  $Y$  fortement corrélés. Supposons que ces signaux proviennent de deux sources qui ne peuvent pas communiquer à l'encodeur, mais qui sont décodées conjointement. Un exemple d'application immédiat est un réseau de capteurs qui envoie des informations très corrélées à une unité centrale [3]. Un second exemple pourrait être un codeur/décodeur vidéo qui requiert une faible complexité à l'encodeur mais dont le décodeur peut être plus complexe [4]. Ces systèmes utilisent ce que l'on appelle le codage de sources distribuées. Le théorème de Slepian-Wolf [5] montre que deux variables aléatoires, i.i.d, à valeurs discrètes,  $X$  et  $Y$ , issues de deux sources corrélées, peuvent être compressées en *codage séparé* à un débit de  $H(X, Y)$  si elles sont décodées

*conjointement*. Ce résultat est le même que si l'on veut compresser et décompresser  $X$  et  $Y$  conjointement à l'encodeur et au décodeur. Il montre aussi une possibilité de réduire le débit en utilisant la corrélation. Ce théorème a été étendu au cas des sources à valeurs continues Gaussiennes par Wyner et Ziv [6]. On notera bien qu'il s'agit au départ de résultats uniquement théoriques et asymptotiques. Des propositions de mise en œuvre commencent à approcher d'assez près la théorie pour des sources d'abord binaires puis Gaussiennes. On notera ici le travail précurseur de Pradhan et Ramchandran [2, 3]. Ils utilisent la matrice de parité  $H$  d'un code correcteur d'erreur pour compresser les sources distribuées. Garcia-Frias et Zhao [7], Aaron et Girod [8] ont indépendamment proposé de poinçonner la sortie d'une source après turbo-codage pour atteindre le taux de compression désiré. La corrélation entre les deux sources permet alors d'aider le turbo-décodeur. Mitran et Bajcsy [9] ont, eux aussi, développé une technique basée sur les turbo-codes, mais au lieu d'utiliser le poinçonnage ils ont directement conçu leur treillis pour la compression. Une autre approche possible est celle étudiée par Zhao et Effros [10], elles dérivent un code, basé sur le CLV, pour compresser deux sources sans mémoire et à alphabet fini. Toutes ces méthodes cherchent à minimiser le taux d'erreur binaire ou des symboles de quantification, au décodeur. Mais qu'en est-il de la distorsion globale du système ? Dans le cas de sources gaussiennes, nous introduisons une nouvelle méthode de codage des sources distribuées, basée sur une technique de codage de sources : les codes de Huffman ; nous la comparons ensuite à la méthode DISCUS [2] de Pradhan et Ramchandran. Cette nouvelle méthode peut être vue comme une application de la technique de Al Jabri et Al-Issa [1] à un modèle de sources distribuées. Dans cet article nous nous focaliserons sur le schéma asymétrique, i.e.  $X$  est compressée,  $Y$  est supposée parfaitement transmise et connue au décodeur (cf. figure 1).

\*R. Zhang est maintenant avec Sony Electronics Inc., 3300 Zanker Road, San Jose, CA 95134, USA

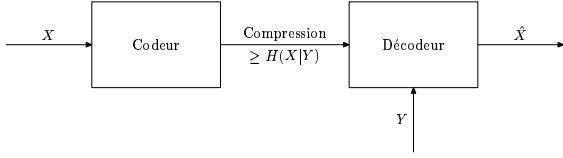


Figure 1 – Compression d'une source distribuée.

Cet article est organisé comme suit. Dans la section 2 nous décrivons l'application de la méthode de Al Jabri et Al-Issa à la conception d'un codeur de sources distribuées. Ce codeur est ensuite comparé en section 3 à celui proposé dans DISCUS [2]. Enfin, nous concluons.

## 2 Description du système

Dans un but de comparaison nous adoptons le scénario de Pradhan et Ramchandran [2].  $X$  et  $Z$  sont deux variables aléatoires Gaussiennes, i.i.d. et de moyenne nulle, la variance de  $X$  est l'unité et la variance de  $Z$  est  $\sigma_Z^2$ .  $X$  et  $Z$  sont décorréliées, on note alors  $Y = X + Z$ . On suppose que  $Y$  est transmise parfaitement et on veut transmettre  $X$ .  $X$  est d'abord quantifiée, on obtient alors  $X_q$  avec  $x_q \in \{1, \dots, M\}$ . Les minuscules représentent des réalisations de la variable aléatoire correspondante. Puis  $X_q$  est encodée pour donner  $X_b$ . Dans [2], la compression est réalisée avec le syndrome d'un code en bloc linéaire, ici nous utilisons une table de CLV. Cette table est préalablement définie en utilisant la méthode de Al Jabri et Al-Issa [1]. Elle tient compte de la densité de probabilité conjointe  $P(x_q, y_q)$ , avec  $Y_q$  la version quantifiée de  $Y$ .  $X_b$  est ensuite transmis sur un canal sans bruit. Au décodage,  $\hat{X}_q$ , l'estimée de  $X_q$  est trouvée grâce à  $X_b$ ,  $Y_q$  et à la table de CLV. Enfin  $\hat{X}$ , l'estimée de  $X$  est déterminée en utilisant toutes les informations disponibles. La figure 2 décrit le schéma de transmission.

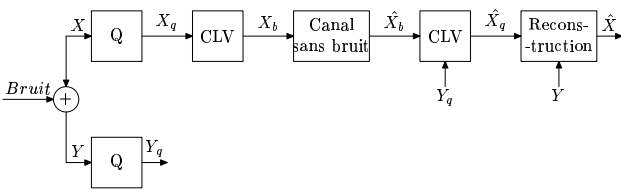


Figure 2 – Transmission d'une source distribuée.

### 2.1 Construction du CLV

Pour construire le CLV on utilise la méthode de Al Jabri et Al-Issa [1]. Cette méthode est la suivante. On veut compresser deux sources à valeurs discrètes d'alphabets finis  $A$  et  $B$ . Le codage de Huffman de  $A$  utilise la probabilité marginale  $P(a)$  et la compression de  $B$  s'effectue par un code de Huffman construit pour  $\phi(B)$  plutôt que pour  $B$ . L'application  $\phi : \{0, \dots, |B|-1\} \rightarrow \{0, \dots, S-1\}$  avec  $S < |B|$  est choisie de telle sorte que  $\phi(b_1) \neq \phi(b_2)$  s'il existe  $a^*$  tel que  $P(a^*, b_1) > 0$  et  $P(a^*, b_2) > 0$ . De plus l'entropie de la variable aléatoire  $b \in \phi(B)$  doit être minimum. Nous

utilisons cette méthode avec  $A = Y_q$  et  $B = X_q$ . Le CLV est donc basé sur  $P(x_q, y_q)$ .

Dans le contexte de sources distribuées  $P(x_q, y_q)$  dépend complètement de la corrélation entre  $X$  et  $Y$ . Ainsi l'algorithme pour trouver le CLV est le suivant :

- Fixer deux séquences d'apprentissage  $X$  et  $Y$  d'une certaine corrélation.
- Déterminer  $P(x_q, y_q)$  grâce à cet échantillon de données.
- Fixer un seuil pour forcer des probabilités à zéro.
- Calculer la table de Huffman pour  $P(x_q, y_q)$  en utilisant la méthode de Al Jabri et Al-Issa décrite plus haut.
- Calculer le taux de compression.

Il faut bien noter que le taux de compression peut être adapté en changeant soit le seuil, soit la valeur de la corrélation. Une fois la table fixée, elle est utilisée dans la transmission quelle que soit la corrélation.

Le tableau 1 donne un exemple des probabilités obtenues avec une paire de séquences d'apprentissage de 100000 échantillons quantifiés avec un Lloyd-Max à 8 niveaux adapté aux statistiques de cette source. La mesure de la corrélation entre les signaux  $X$  et  $Y$  est appelé corrélation-SNR (CSNR) et est définie par la quantité  $CSNR = 10 \log(\frac{\sigma_X^2}{\sigma_Z^2})$ . Ici  $CSNR = 15,2$  dB et le seuil est fixé à 0. Le tableau 2, donne la table de Huffman correspondant aux statistiques précédentes. On peut alors calculer la compression moyenne pour  $X_q$ ,  $L_{moyen} = 1,65$  bit/symbole, son entropie  $H(X_q) = 2,41$  bit/symbole et l'entropie de la table  $H(X_q|Y_q) = 0,94$  bit/symbole.

$Y_q \backslash X_q$	1	2	3	4	5	6	7	8
8	0	0	0	0	0	0	0,0029	0,0071
7	0	0	0	0	0	0,0128	0,0378	0,0015
6	0	0	0	0	0,0281	0,1213	0,0082	0
5	0	0	0	0,0340	0,2224	0,0228	0	0
4	0	0	0,0229	0,2232	0,0346	0	0	0
3	0	0,0083	0,1218	0,0279	0	0	0	0
2	0,0014	0,0389	0,0125	0	0	0	0	0
1	0,0067	0,0028	0	0	0	0	0	0

Tableau 1 – Statistiques obtenues avec un  $CSNR = 15,2$  dB et 100000 échantillons.

Groupe de symboles	Probabilités	Mot de code
2, 5, 8	0,3437	0
1, 4, 7	0,3421	10
3, 6	0,3141	11

Tableau 2 – Groupes formés avec l'exemple du tableau 1.

### 2.2 Estimation de $\hat{X}$ connaissant $\hat{X}_q$

Une fois  $\hat{x}_q$  déterminée, la valeur  $\hat{x}$  est calculée en utilisant toutes les informations possibles. Cette étape permet d'améliorer la distorsion globale du système mais n'intervient pas dans le calcul du taux d'erreur des symboles quantifiés. On utilise en fait la fonction de reconstruction de [2]. On note  $I_q$  l'intervalle de quantification dont le centre est  $\hat{x}_q$ . On veut minimiser l'espérance de la distor-

sion  $\rho(x, \hat{x})$ . L'estimée optimale est alors donnée par :

$$\hat{x} = \arg \min_{a \in \mathcal{R}} E[\rho(x, a) | X \in I_q, Y = y]. \quad (1)$$

En fait on cherche la meilleure valeur réelle pour  $\hat{x}$  dans l'intervalle  $I_q$  sachant la valeur exacte de  $Y$ .

### 3 Comparaison des deux méthodes

Le but de nos comparaisons est de montrer l'évolution de la distorsion globale du système (SNR), ainsi que le taux d'erreur des symboles quantifiés (TE) en fonction du  $CSNR$  pour les deux schémas de compression, i.e. DISCUS et CLV. Deux quantificateurs de Lloyd-Max ont été utilisés pour la mise en œuvre expérimentale.

#### 3.1 Quantificateur de Lloyd-Max à 8 niveaux

Par une paramétrisation appropriée, la méthode de la section 2 nous permet de générer aussi les solutions à 1 et 2 bit/symbole qui permettent la comparaison avec la méthode DISCUS. Le code pour un taux de compression de 2 bit/symbole est donné tableau 3, il a été défini avec des séquences d'apprentissage de  $CSNR = 13$  dB et de seuil fixé à 0. Le CLV pour un taux de compression de 1

Groupe de symboles	Probabilités	Mot de code
1, 4, 7	0,3431	00
2, 6	0,2082	10
3, 8	0,2850	11
5	0,1636	01

Tableau 3 – Groupes formés pour 2 bit/symbole.

bit/symbole est donné tableau 4. Il a été défini avec des séquences d'apprentissage corrélées avec un  $CSNR = 15,2$  dB et le seuil fixé à 0,0128. Cette table est donc sous-optimale car son seuil est très élevé. Enfin le CLV pour le taux de compression de 1,65 bit/symbole a été donné précédemment au tableau 2. La figure 3 présente les per-

Groupe de symboles	Probabilités	Mot de code
2, 5, 8, 3	0,5009	0
1, 4, 7, 6	0,499	1

Tableau 4 – Groupes formés pour 1 bit/symbole.

formances de ces différentes méthodes. En haut le taux d'erreur des symboles quantifiés est représenté en fonction du  $CSNR$ . En bas la distorsion globale est donnée en fonction de ce même  $CSNR$ . Les courbes nommées "Borne de Wyner Ziv" représentent la distorsion minimale du système global avec codage séparé et décodage conjoint. Elles dépendent du débit et sont calculées d'après [6]. La courbe intitulée "Sans Erreur" correspond à la distorsion provenant uniquement du quantificateur, i.e. tous les symboles, quantifiés sur trois bits, sont parfaitement connus au décodeur. C'est donc la limite à atteindre pour les deux méthodes étudiées. D'après ces courbes, plusieurs remarques peuvent être formulées :

- La courbe résultat de la méthode avec le CLV pour un taux de 1 bit/symbole est très mauvaise quelle que soit la corrélation. Ceci découle directement du fait que cette table n'est jamais optimale comme il a été mentionné précédemment.
- Pour un taux de 2 bit/symbole, la méthode DISCUS donne de meilleurs résultats que la méthode utilisant le CLV.
- Les taux de compression 1,65 bit/symbole et 2 bit/symbole pour la méthode avec le CLV donnent des résultats très proches.

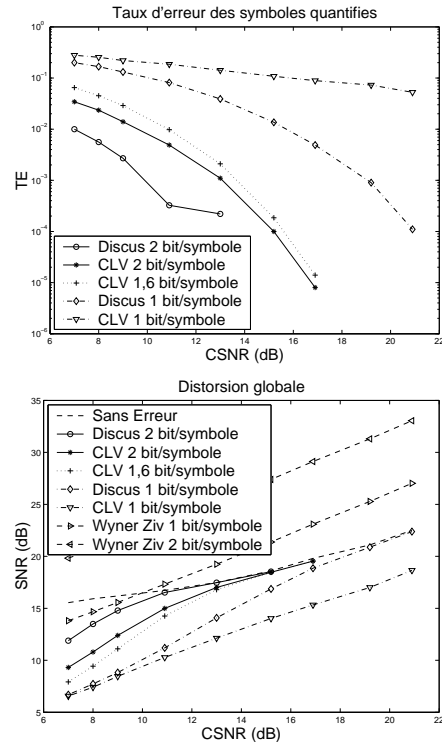


Figure 3 – Performance des différentes méthodes pour 8 niveaux de quantification.

#### 3.2 Quantificateur de Lloyd-Max à 16 niveaux

Pour la méthode utilisant le CLV, trois taux de compression ont été étudiés : 3 bit/symbole, 2,6 bit/symbole et 2 bit/symbole. La méthode DISCUS, quant à elle, a été étudiée à 2 bit/symbole et 3 bit/symbole. Le code pour un taux de compression de 3 bit/symbole est donné tableau 5, il a été défini avec des séquences d'apprentissage corrélées avec un  $CSNR = 9$  dB et un seuil fixé à 0. De la même façon, le code pour un taux de compression de 2,6 bit/symbole est donné tableau 6. La corrélation des deux séquences d'apprentissage est  $CSNR = 15,2$  dB et le seuil est fixé à 0. Enfin, pour 2 bit/symbole, le CLV est présenté tableau 7, il a été défini avec des séquences corrélées avec un  $CSNR = 19,2$  dB et un seuil fixé à 0. La figure 4 présente les performances de ces différentes méthodes. Comme pour le Lloyd-Max à 8 niveaux, les

Groupe de symboles	Probabilités	Mot de code
1, 8, 15	0,1465	000
2, 10	0,1305	001
9	0,1305	010
6, 14	0,1221	011
7, 16	0,1218	100
3, 11	0,1202	101
5, 13	0,1148	110
4, 12	0,1137	111

Tableau 5 – Groupes formés pour 3 bit/symbole.

Groupe de symboles	Probabilités	Mot de code
2, 6, 11, 15	0,2153	10
1, 7, 13, 16	0,1719	11
4, 10	0,1621	000
8, 14	0,1568	001
3, 9	0,1557	010
5, 12	0,1383	011

Tableau 6 – Groupes formés pour 2,6 bit/symbole.

Groupe de symboles	Probabilités	Mot de code
2, 5, 8, 11, 14	0,3332	00
3, 6, 9, 12, 15	0,3325	01
1, 4, 10, 13, 16	0,2177	10
7	0,1167	11

Tableau 7 – Groupes formés pour 2 bit/symbole.

courbes “Borne de Wyner-Ziv” représentent les limites théoriques pour chaque débit et la courbe intitulée “Sans Erreur” correspond à une simulation où tous les symboles quantifiés transmis seraient retrouvés parfaitement au décodage, i.e. quantifiés sur quatre bits. Au vu des courbes on peut faire les remarques suivantes :

- Pour un taux de compression de 2 bit/symbole, les performances de la méthode DISCUS sont meilleures.
- Pour un taux de 3 bit/symbole, les résultats des deux méthodes sont très proches et pour une faible corrélation la méthode utilisant le CLV donne de meilleurs résultats.

## 4 Conclusion

Nous avons présenté dans cet article une application de la méthode présentée par Al Jabri et Al-Issa au contexte du codage des sources gaussiennes distribuées. Cette technique, basée sur l’utilisation du CLV, a été comparée à la technique DISCUS qui, elle, utilise les codes correcteur d’erreur. Des comparaisons ont été réalisées au niveau taux d’erreur des symboles quantifiés mais aussi au niveau de la distorsion globale. On montre que la méthode DISCUS donne, dans la plupart des cas, de meilleurs résultats que celle utilisant le CLV. Cependant cette dernière reste très compétitive. De plus la méthode utilisant le CLV est très intéressante au niveau de sa flexibilité dans le choix du taux de compression.

## Références

[1] A. Kh. Al Jabri et S. Al-Issa. Zero-error codes for correlated information sources. Dans *Proceedings of Cryptography*, pages 17–22, Cirencester, UK, Décembre 1997.

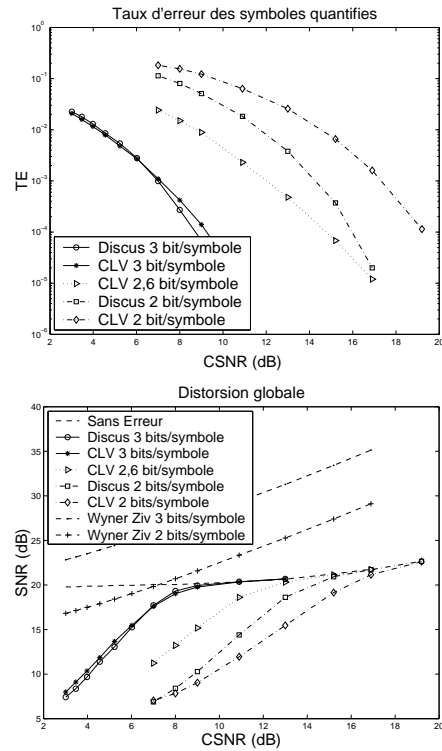


Figure 4 – Performance des différentes méthodes pour 16 niveaux de quantification.

[2] S. S. Pradhan et K. Ramchandran. Distributed source coding using syndromes (discus) : Design and construction. Dans *Proceedings of DCC*, pages 158–167, Snowbird, USA, Mars 1999.

[3] S. S. Pradhan et K. Ramchandran. Distributed source coding : symmetric rates and applications to sensor networks. Dans *Proceedings of DCC*, pages 363–372, Snowbird, Utah, USA, Mars 2000.

[4] A. Aaron et B. Girod. Wyner-Ziv coding of motion video. Dans *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, USA, Novembre 2002.

[5] D. Slepian et J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory*, 19 :471–480, Juillet 1973.

[6] A. D. Wyner et J. Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory*, 22 :1–10, Janvier 1976.

[7] J. Garcia-Frias et Y. Zhao. Compression of correlated binary sources using turbo codes. *IEEE Communication Letters*, 5(10) :417–419, Octobre 2001.

[8] A. Aaron et B. Girod. Compression with side information using turbo codes. Dans *DCC*, pages 252–261, Snowbird, USA, Janvier 2002.

[9] J. Bacjy et P. Mitran. Coding for the Slepian-Wolf problem with turbo codes. Dans *Global Communication Symposium*, pages 1400–1404, San Antonio, USA, Novembre 2001.

[10] Q. Zhao et M. Effros. Optimal code design for lossless and near lossless source coding in multiple access network. Dans *Data Compression Conference (DCC)*, pages 263–272, Snowbird, Utah, USA, Mars 2001.