

Μη απωλεστική συμπίεση

Στην μη απωλεστική μπορούμε να πετύχουμε συμπίεση της τάξης 50% έως 60% χωρίς όμως να υπάρχει απώλεια ποιότητας (Beurden, 2013). Στα αρχικά στάδια της μη απωλεστικής συμπίεσης χρησιμοποιήθηκαν κοινοί αλγόριθμοι συμπίεσης (όπως ο rpkzip) αλλά σύντομα εγκαταλείφθηκαν γιατί δεν εξασφάλιζαν μεγάλα ποσοστά συμπίεσης (Hans & Schafer, 1999). Για αυτό το λόγο αναπτύχθηκαν ειδικοί αλγόριθμοι για την μη απωλεστική συμπίεση μουσικής. Οι αλγόριθμοι αυτοί όμως παρουσιάζουν πολύ μεγαλύτερη υπολογιστική πολυπλοκότητα μιας που βασίζονται κυρίως σε τεχνικές συνέλιξης (1-1 Convolution) και τεχνικές γραμμικής πρόβλεψης (linear prediction) (Wikipedia, Audio Compression, 2013). Στην παρακάτω εικόνα μπορούμε να δούμε τη βασική λειτουργία των αλγορίθμων αυτών.



Εικόνα 1 Βασικές λειτουργίες αλγορίθμων μη απωλεστικής συμπίεσης (Hans & Schafer, 1999)

Το πρώτο βήμα της συμπίεσης είναι η διαμόρφωση (framing). Στο στάδιο αυτό το ηχητικό σήμα χωρίζεται σε πλαίσια ίσης χρονικής διάρκειας, κάτι που διευκολύνει την περαιτέρω επεξεργασία του. Η επιλογή της χρονικής διάρκειας αυτής είναι κρίσιμη διότι αν είναι πολύ μικρή θα υπάρχει σημαντική αύξηση του μεγέθους του αρχείου λόγω του μεγάλου πλήθους των κεφαλίδων (headers) των πλαισίων. Στις κεφαλίδες αυτές αναγράφονται πληροφορίες που αφορούν στην μετέπειτα συμπίεση των πλαισίων αυτών. Αναλόγως την εφαρμογή που χρησιμοποιούμε αυτές οι κεφαλίδες μπορεί να περιέχουν πολυμεσικά δεδομένα ή πληροφορίες συγχρονισμού (Hans & Schafer, 1999).

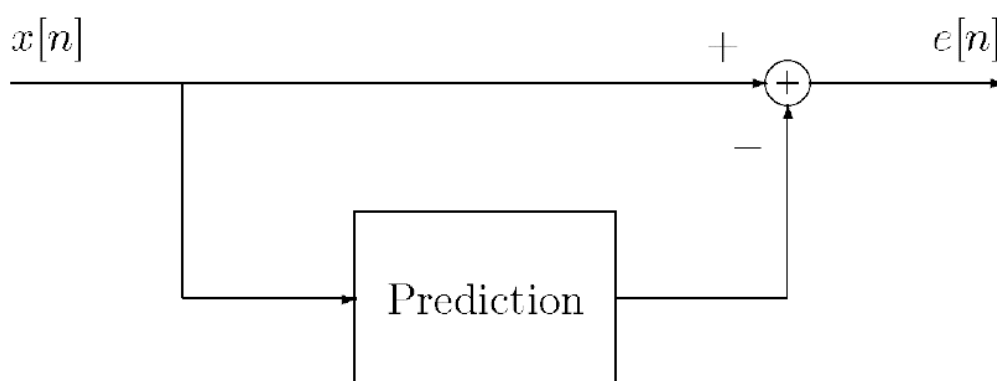
Στο δεύτερο στάδιο έχουμε την ενδοκαναλική αποσυσχέτιση (Intra-Channel Decorrelation). Ο βασικός στόχος μας είναι να απομακρύνουμε κάθε μορφής πλεονασμό αποσυσχετίζοντας τα δύο (ή περισσότερα) κανάλια ήχου μεταξύ τους. Αρχικά έχουμε τη μετατροπή ψηφιακού σήματος του κάθε πλαισίου των 2 στερεοφωνικών καναλιών (L,R) σε δύο σύνθετα σήματα X (mid) και Y(side), όπου:

$$X = (L + R) / 2 \quad \text{και} \quad Y = (L - R)$$

Αυτό γίνεται διότι συνήθως υπάρχει συσχέτιση μεταξύ των σημάτων των δύο καναλιών, και αυτό στην συνέχεια μπορεί να αξιοποιηθεί με διάφορους τρόπους, με πιο δημοφιλή την κωδικοποίηση mid/side. Σε αυτή την κωδικοποίηση η χωρική εντόπιση (spatial localization) του στερεοφωνικού σήματος προσδιορίζεται από τις διαφορές στην ένταση του ήχου ενός κύματος καθώς φτάνει σε φάση σε ένα ζεύγος μικροφώνων. Στηρίζεται πλήρως στα κατευθυντικά χαρακτηριστικά (πολικά μοτίβα) του ζεύγους μικροφώνου για να παραχθεί αυτό το αποτέλεσμα, δεδομένου ότι υπάρχουν μόνο διαφορές έντασης και δεν υπάρχουν διαφορές φάσης μεταξύ των καναλιών για κάθε μεμονωμένη πηγή που φθάνουν σε ένα συμπίπτον ζεύγος. (Ronald & Wesley, 1982).

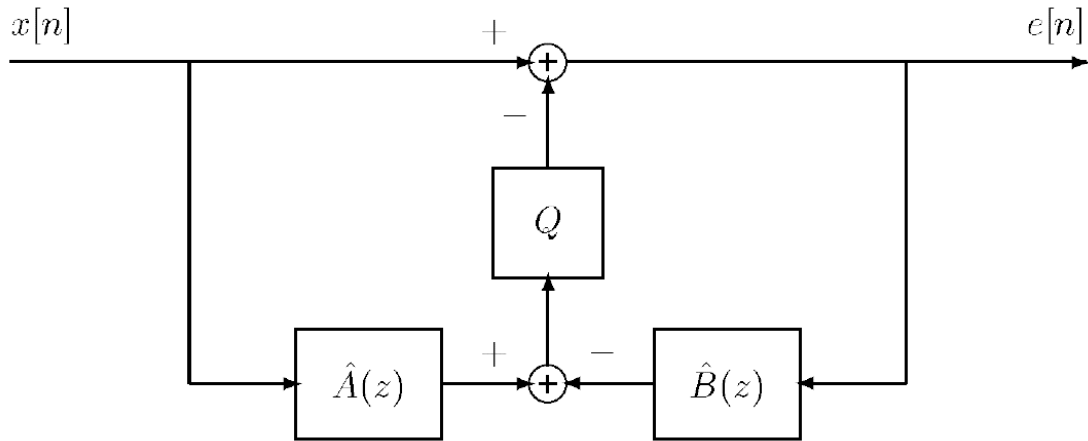
Στη συνέχεια τα X και Y επεξεργάζονται από έναν αλγόριθμο πρόβλεψης με σκοπό να απομακρυνθεί οποιοσδήποτε πλεονασμός. Βασικά, ο στόχος αυτού του σταδίου είναι να κάνει τους πίνακες που περιέχουν τις διαδοχικές τιμές των X και Y να περιέχουν τις μικρότερες δυνατές τιμές, ενώ το σήμα παραμένει ακόμη ασυμπίεστο. Αυτό το στάδιο είναι αυτό που ξεχωρίζει τον κάθε αλγόριθμο συμπίεσης από τους υπόλοιπους διότι υπάρχουν αμέτρητοι τρόποι να υλοποιηθεί ένας τέτοιος αλγόριθμος.

Το πιο διαδεδομένο μοντέλο πρόβλεψης είναι αυτό της γραμμικής πρόβλεψης. Στο μοντέλο αυτό προσπαθούμε να προβλέψουμε την τιμή ενός δείγματος $x[n]$, χρησιμοποιώντας τα προηγούμενα δείγματα $x[n-1]$, $x[n-2]$, κλπ.



Εικόνα 2 Μοντέλο πρόβλεψης (Hans & Schafer, 1999)

Αν η πρόβλεψη είναι πετυχημένη τότε το λάθος της πρόβλεψης $e[n]$, θα είναι μηδέν οπότε δε θα καταλαμβάνει χώρο. Γενικά η τιμή του $e[n]$ θα είναι κατά μέσο όρο μικρότερη από την τιμή του $x[n]$, και έτσι θα απαιτούνται λιγότερα bits για την κωδικοποίησή του.



Εικόνα 3 Αναπαράσταση μαθηματικού μοντέλου της πρόβλεψης (Hans & Schafer, 1999)

Η μαθηματική έκφραση του μοντέλου πρόβλεψης που χρησιμοποιούν οι αλγόριθμοι μη απωλεστικής συμπίεσης αναπαριστάται στην εικόνα 5 (σελ.17) και η μαθηματική έκφραση είναι η ακόλουθη:

$$e[n] = x[n] - Q \left\{ \sum_{k=1}^M \hat{a}_k x[n-k] - \sum_{k=1}^N \hat{b}_k e[n-k] \right\},$$

Όπου το $Q\{\}$ αναπαριστά τον κβαντισμό στο ίδιο μήκος λέξης με το $x[n]$ και τα $\hat{A}(z)$ και $\hat{B}(z)$ πολυώνυμα z-μετασχηματισμού με κβαντισμένους συντελεστές και τύπους $\hat{A}(z) = \sum_{k=1}^M \hat{a}_k x[n-k]$ και $\hat{B}(z) = \sum_{k=1}^N \hat{b}_k e[n-k]$.

Όπως φαίνεται και από το διάγραμμα αν $\hat{B}(z) = 0$ τότε δεν έχουμε ανάδραση και το φίλτρο του λάθους της πρόβλεψης έχει πεπερασμένη κρουστική απόκριση (FIR predictor). Διαφορετικά αν $\hat{B}(z) \neq 0$ τότε το φίλτρο έχει άπειρη κρουστική απόκριση (IIR predictor).

Στους περισσότερους αλγόριθμους χρησιμοποιείται η λύση του FIR predictor και οι συντελεστές του φίλτρου πρόβλεψης $\hat{A}(z)$ επιλέγονται έτσι ώστε να ελαχιστοποιείται το μέσο τετράγωνο του λάθους της πρόβλεψης. Σε αυτή την περίπτωση οι συντελεστές του $\hat{A}(z)$ μπορούν να υπολογισθούν από την λύση απλών γραμμικών εξισώσεων. Οι ίδιοι ακριβώς συντελεστές μπορούν να χρησιμοποιηθούν για την ανακατασκευή του σήματος (Rabiner & Schafer, 1978). Οι συντελεστές αυτοί όμως είναι συνήθως κλασματικοί αριθμοί, οπότε για να κωδικοποιηθούν πρέπει να κβαντιστούν γιατί το φίλτρο του λάθους της πρόβλεψης πρέπει να υλοποιηθεί με

ακέραιους συντελεστές. Αν όμως η πρόβλεψη γίνει με ακέραιους συντελεστές είναι ευκολότερη η περαιτέρω επεξεργασία της κωδικοποίησης της πρόβλεψης και μάλιστα γίνεται με τον ίδιο τρόπο σε όλες τις υπολογιστικές πλατφόρμες (Hans & Schafer, 1999). Για αυτό το λόγο οι περισσότεροι αλγόριθμοι αποφεύγουν την χρήση του μέσου τετραγώνου του λάθους της πρόβλεψης

Η πρόβλεψη των συντελεστών μπορεί επίσης να γίνει με επιλογή από μία μικρή βιβλιοθήκη προκαθορισμένων συντελεστών. Έτσι αποφεύγεται ο υπολογισμός νέων συντελεστών και έτσι έχουμε εξοικονόμηση υπολογιστικών πόρων. Σε κάθε πλαίσιο έτσι κωδικοποιείται μόνο η θέση του συντελεστή στη βιβλιοθήκη και το λάθος της πρόβλεψης. Η μέθοδος αυτή ακολουθείται και στην περίπτωση της χρήσης IIR predictor διότι στην περίπτωση αυτή ο υπολογισμός του μέσου τετραγώνου του λάθους της πρόβλεψης είναι ιδιαίτερα πολύπλοκος (Hans & Schafer, 1999). Γενικά η λύση του IIR predictor μπορεί να επιφέρει καλύτερα αποτελέσματα χρησιμοποιώντας το ίδιο πλήθος συντελεστών με την χρήση του FIR predictor αλλά η υλοποίησή του αλγορίθμου είναι αρκετά πολύπλοκη οπότε και συνήθως αποφεύγεται (Craven, Law, & Stuart, 1997). Για χάρην απλότητας λοιπόν, οι περισσότεροι αλγόριθμοι που χρησιμοποιούνται σήμερα βασίζονται στον αλγόριθμο του Shorten ο οποίος υλοποιείται με τη χρήση IIR predictor με βιβλιοθήκη συντελεστών (Robinson, 1994). Ένα απλό παράδειγμα περιγραφής της λειτουργίας του αλγορίθμου αυτού χρησιμοποιώντας απλή γραμμική άλγεβρα:

Έστω PX και PY οι προβλέψεις για τα X και Y αντίστοιχα, X₋₁ η προηγούμενη τιμή του X και X₋₂ η τιμή του X δύο βήματα πίσω. Και έστω οι ακόλουθες σχέσεις πρόβλεψης:

$$PX = (2 * X_{-1}) - X_{-2}$$

$$PY = (2 * Y_{-1}) - Y_{-2}$$

Αν λοιπόν οι διαδοχικές τιμές στον πίνακα του X είναι X = [2, 8, 24, ω, ...] (όπου ω είναι η προς πρόβλεψη τιμή) τότε

$$PX = (2 * X_{-1}) - X_{-2} = (2 * 24) - 8 = 40$$

Ακολουθώς οι τιμές των PX και PY συγκρίνονται με τις πραγματικές και η διαφορά αυτή (ονομάζεται «λάθος της πρόβλεψης») χρησιμοποιείται στο επόμενο στάδιο της κωδικοποίησης.

Οι καλοί αλγόριθμοι πρόβλεψης είναι προσαρμοστικοί, έτσι ώστε να προσαρμόζονται με το πόσο «προβλέψιμα» είναι τα δεδομένα την συγκεκριμένη στιγμή. Για παράδειγμα πάλι ας υποθέσουμε ότι έχουμε ένα παράγοντα πρόβλεψης “m” που παίρνει τιμές στο διάστημα 0 με 1024 (το 0 σημαίνει καμία πρόβλεψη και το 1024 σημαίνει πλήρης πρόβλεψη). Μετά από κάθε πρόβλεψη ο παράγοντας αυτός αλλάζει τιμή, είτε προς τα πάνω, είτε προς τα κάτω, αναλόγως με το αν η πρόβλεψη ήταν πετυχημένη ή όχι. Χρησιμοποιώντας τα νούμερα από το προηγούμενο παράδειγμα:

$$\text{Έστω } \omega = 45 \text{ και } m = 512$$

$$[\text{Final Value}] = \omega - (PX * m / 1024) = 45 - (40 * 512 / 1024) = 45 - 20 = 25$$

Μετά από αυτό τον υπολογισμό η τιμή του m θα ανεβεί διότι ένα μεγαλύτερο m θα έδινε καλύτερη πρόβλεψη.

Χρησιμοποιώντας διαφορετικές εξισώσεις πρόβλεψης και πολλαπλά περάσματα των δεδομένων από τον αλγόριθμο πρόβλεψη μπορεί να έχουμε σημαντικές διαφορές στο επίπεδο της συμπίεσης. Οι πιο διαδεδομένες εξισώσεις πρόβλεψης (για διάφορες τάξεις κωδικοποίησης) είναι οι ακόλουθες (Robinson, 1994):

$$P_0 = 0$$

$$P_1 = X_{-1}$$

$$P_2 = (2 * X_{-1}) - X_{-2}$$

$$P_3 = (3 * X_{-1}) - (3 * X_{-2}) + X_{-3}$$

Το επόμενο στάδιο είναι γίνεται η συμπίεση του σήματος (Entropy Coding). Ο στόχος της συμπίεσης είναι να γίνουν όλες τις τιμές του ψηφιοποιημένου ηχητικού σήματος όσο το δυνατόν μικρότερες, αφαιρώντας οποιαδήποτε συσχέτιση που μπορεί να υπάρχει μεταξύ τους. Προφανώς όσο πιο μικρές είναι οι τιμές τόσο λιγότερα bits απαιτούνται για την αποθήκευσή του σήματος. Για παράδειγμα, λέμε ότι θέλουμε να κωδικοποιήσει αυτό το φάσμα των αριθμών (μήκους 32 bit):

$$[10, 14, 15, 46] \text{ ή καλύτερα σε δυαδικό } [1010, 1110, 1111, 101110]$$

Επειδή όμως χρησιμοποιούμε λέξεις 32bit στο τελικό αποτέλεσμα θα χρειαζόμασταν 128bit. Αν όμως είχαν αποθηκευθεί μόνο οι αξίες χωρίς τα επιπλέον μηδενικά θα είχαμε το ακόλουθο αποτέλεσμα 101011101111101110 το οποίο απαιτεί λιγότερα bits για να αποθηκευθεί. Το πρόβλημα τώρα όμως είναι ότι δεν ξέρουμε πού ξεκινά και πού τελειώνει η κάθε ψηφιακή λέξη.

Για να μπορέσουμε να αποθηκεύσουμε τα δεδομένα σε αυτή την μορφή και να μπορέσουμε στη συνέχεια να τα αποκωδικοποιήσουμε, θα πρέπει να χρησιμοποιήσουμε την κωδικοποίηση Golomb (Golomb, 1966). Οι περισσότεροι αλγόριθμοι συμπίεσης χρησιμοποιούν παραλλαγές της κωδικοποίησης Golomb που βασίζονται στον Robert F. Rice (Rice, 1971).

Η κωδικοποίηση Rice αποτελείται από τα ακόλουθα βήματα:

1. Κάνεις την καλύτερη πρόβλεψη για το πόσα bits θα έχει ο κάθε αριθμός που θα κωδικοποιήσεις και την τιμή αυτή την ονομάζεις k
2. Παίρνεις τα k bits που βρίσκονται στο δεξί άκρο του αριθμού και τα αποθηκεύεις (k least significant bits).
3. Από τον δυαδικό αριθμό απομακρύνεις αυτά τα k least significant bits και βρίσκεις την τιμή του αριθμού που απέμεινε. (αυτά η τιμή αντιπροσωπεύει τα ψηφία που είναι πέραν των k bits)
4. Χρησιμοποίησε αυτή την τιμή για την κωδικοποίηση του αριθμού. Η κωδικοποιημένη τιμή θα αποτελείται από τόσα μηδενικά όση η τιμή του αριθμού που υπολογίσαμε στο βήμα 3, μετά τον αριθμό 1 και τέλος τα k least significant bits.

Παράδειγμα:

Έστω ότι ο προς κωδικοποίηση αριθμός είναι ο $(578)_{10}$ και $k = 8$

Έχουμε αρχικά το πρόσημο (1 για θετικό, 0 για αρνητικό)

- Πρόσημο [1]
- $\lfloor \frac{n}{2^k} \rfloor$ μηδενικά: $\lfloor \frac{n}{2^k} \rfloor = \lfloor \frac{578}{256} \rfloor = 2$ άρα [00]
- 1 τερματισμού: [1]
- 8 least significant bits του 578 = [01000010]

Οπότε τελικά: [1][00][1][01000010] = 100101000010

Κατά την κωδικοποίηση το βέλτιστο k προσδιορίζεται με την ακόλουθη διαδικασία:
Πρώτα υπολογίζεται ο μέσος όρος (average) των προηγούμενων τιμών (από 16 μέχρι

128 τιμές) και στη συνέχεια υπολογίζεται το k για αυτό το μέσο όρο με από τον τύπο

$$k = \left\lfloor \frac{\log(\text{average})}{\log(2)} \right\rfloor.$$

Στις περισσότερες υλοποιήσεις για χάρη απλότητας η τιμή του k παραμένει σταθερή για όλο το πλαίσιο της ηχητικής πληροφορίας. Η τιμή του τότε υπολογίζεται από την σχέση:

$$k = \log_2(\ln(2)E(|e[n]|))$$

όπου $E()$ η μαθηματική συνάρτηση προσδοκώμενης τιμής (Robinson, 1994).

Πέραν της κωδικοποίησης Rice υπάρχει η κωδικοποίηση Huffman και η κωδικοποίηση run length coding οι οποίες χρησιμοποιούνται περισσότερο στις εφαρμογές συμπίεσης δεδομένων και λιγότερο στις εφαρμογές συμπίεσης ηχητικού σήματος (Hans & Schafer, 1999).

Βιβλιογραφία

1. Beurden, M. V. (2013, Ιανουάριος). *Lossless Audio Codec Comparison*. Retrieved Ιανουάριος 2013, from Δικτυακός Τόπος της Xiph.org: <https://xiph.org/flac/comparison.pdf>
2. Craven, P., Law, M., & Stuart, J. (1997). *Lossless Compression Using IIR Prediction Filters. 102nd AES Convention*. Munich.
3. Golomb, S. W. (1966). Run-length encodings. *IEEE Trans Info Theory* 12 , p. 399.
4. Hans, M., & Schafer, R. W. (1999). *Lossless Compression of Digital Audio*. Palo Alto: HP Laboratories.
5. Rabiner, L. R., & Schafer, R. W. (1978). *Digital processing of speech signals*. Prentice - Hall.
6. Rice, R. F. (1971, 12). Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data. *Communication Technology, IEEE Transactions on* , pp. 889-897.
7. Robinson, T. (1994). *SHORTEN: Simple lossless and near lossless waveform compression*. Cambridge: Cambridge University Engineering Department.
8. Ronald, S. D., & Wesley, D. L. (1982, October). M-S Stereo: A Powerful Technique for Working in Stereo. *JAES Volume 30 Issue 10* , pp. 707-718.

9. *Wikipedia, Audio Compression (Data)*. (2013, Ιούνιος). Ανάκτηση Ιούνιος 2013, από Wikipedia:
[http://en.wikipedia.org/wiki/Audio_compression_\(data\)#Audio](http://en.wikipedia.org/wiki/Audio_compression_(data)#Audio)