2016-2017

Υλοποίηση κωδικοποιητή & αποκωδικοποιητή AAC

Συστήματα Πολυμέσων και Εικονική Πραγματικότητα

Δημήτρης Νήρας 8057
Γεώργιος Σχινάς 7985

Εισαγωγή

Στην εργασία αυτή υλοποιήθηκε μία παραλλαγή του προτύπου AAC, σε περιβάλλον MatLab, για κωδικοποίηση και αποκωδικοποίηση ενός αρχείου wav. Η ανάπτυξη έγινε σε τρία ανεξάρτητα στάδια όπως προτάθηκε από την αναλυτική εκφώνηση και η παρουσίασή τους θα γίνει σε ξεχωριστές ενότητες στην παρούσα αναφορά.

Επίπεδο 1

Η πρώτη συνάρτηση που υλοποιήθηκε ήταν η

$$\text{frameType} = \textbf{SSC}(\text{frameT}, \text{nextFrameT}, \text{prevFrameType})$$

η οποία υλοποιεί την βαθμίδα Sequence Segmentation Control. Πρόκειται ουσιαστικά για την συνάρτηση που για δεδομένο frame προσδιορίζει τι είδους είναι. Για την υλοποίηση και θεωρητική τεκμηρίωσή της, στηριζόμαστε στην υπό-ενότητα 2.1 της εκφώνησης.

Έπειτα αναπτύχθηκε η

$$\text{frameF} = \textbf{filterbank}(\text{frameT}, \text{frameType}, \text{winType})$$

η οποία δέχεται ένα συγκεκριμένο frame και προσδιορίζει τους συντελεστές MDCT για αυτό. Ο αλγόριθμος που χρησιμοποιήθηκε για τον προσδιορισμό του μετασχηματισμού του MDCT πάρθηκε από το διαδίκτυο ενώ για τους δύο δυνατούς τύπους παραθύρων KBD και SIN αναπτύχθηκαν οι συναρτήσεις KBD και sinusoid, αντίστοιχα, οι οποίες επιστρέφουν το αντίστοιχο παράθυρο βαρών. Ιδιαίτερη προσοχή δόθηκε στην δημιουργία των παραθύρων ώστε να έχουν ακριβή ευθυγράμμιση με το frame που επεξεργαζόμασταν, καθώς μία μικρή απόκλιση οδηγούσε σε λανθασμένη κωδικοποίηση και απώλεια πληροφορίας που γινόταν αντιληπτή σε μεταγενέστερο στάδιο.

Ακριβώς αντίστοιχα και αντίστροφα αναπτύχθηκε η

$$\text{frameT} = \textbf{iFilterbank}(\text{frameF}, \text{frameType}, \text{winType})$$

η οποία αντιστρέφει την διαδικασία της προηγούμενης συνάρτησης.

Η συνάρτηση

$$\text{AACSeq1} = \textbf{AACoder1}(\text{fNameIn})$$

που αναπτύχθηκε αμέσως μετά, είναι αυτή η οποία δοσμένου του αρχείου ήχου, το διαχωρίζει κατάλληλα σε frames και για το καθένα από αυτά υλοποιεί τις παραπάνω συναρτήσεις με στόχο να επιστρέψει ένα και μόνο struct το οποίο περιέχει αρχείο ήχου, διαμελισμένο σε κομμάτια και μετασχηματισμένο.

Για την αντιστροφή αυτής της διαδικασίας αναπτύχθηκε η

$$\text{x} = \textbf{iAACoder1}(\text{AACSeq1}, \text{fNameOut})$$

η οποία αξιοποιώντας τις αντίστροφες συναρτήσεις που αναφέρθηκαν παραπάνω, επιστρέφει το αρχικό κομμάτι ήχου.

Τέλος, για να αξιολογήσουμε όλες αυτές τις μετατροπές και μετασχηματισμούς, κατασκευάστηκε η

$$\text{SNR} = \textbf{demoAAC1}(\text{fNameIn}, \text{fNameOut})$$

η οποία υπολογίζει την σηματοθορυβική σχέση του αρχικού και τελικού ήχου, και αποδεικνύει παράλληλα την επιτυχή εκτέλεση του πρώτου σκέλους της εργασίας.

```
Level 1
=======
Coding: time ellapsed is 0.476375 seconds
Decoding: time ellapsed is 0.326009 seconds
Channel 1 SNR: 301.619869 dB
Channel 2 SNR: 301.730702 dB
```

Fig.1 Using KBD window

```
Level 1
=======
Coding: time ellapsed is 0.223494 seconds
Decoding: time ellapsed is 0.195902 seconds
Channel 1 SNR: 307.836755 dB
Channel 2 SNR: 307.908893 dB
```

Fig.2 Using SIN window

Επίπεδο 2

Το δεύτερο στάδιο ανάπτυξης έρχεται να προσθέσει στο πρώτο το Temporal Noise Shaping. η κεντρική ιδέα του Temporal Noise Shaping είναι η εφαρμογή ενός μοντέλου γραμμικής πρόβλεψης στους συντελεστές MDCT ώστε το σφάλμα κβαντισμού να προσαρμοστεί στο σχήμα του σήματος στο πεδίο του χρόνου. Στηριζόμενοι λοιπόν, στην υπό-ενότητα 2.3 της εκφώνησης, υλοποιήθηκε η συνάρτηση

$$[\text{frameFout}, \text{TNScoeffs}] = \textbf{TNS}(\text{frameFin}, \text{frameType})$$

η συνάρτηση αυτή, πριν αναζητήσει τους κατάλληλους συντελεστές πρόβλεψης, κανονικοποιεί τους συντελεστές MDCT ως προς την ενέργειά τους, και αφού προσδιορίσει τους συντελεστές αυτούς του κβαντίζει περνώντας τους από κβαντιστή συγκεκριμένων προδιαγραφών (3bit,

συμμετρικό, με βήμα 0.1). Ενώ στο τέλος φιλτράρει το αρχικό frame με τους κβαντισμένους πλέον συντελεστές.

Αμέσως μετά αναπτύχθηκε και η αντίστροφη συνάρτηση

$$\text{frameFout} = \textbf{iTNS}(\text{frameFin}, \text{frameType}, \text{TNScoeffs})$$

ώστε να είναι δυνατή η αντιστροφή της βαθμίδας αυτής.

Συνεχίζοντας, η συνάρτηση

$$\text{AACSeq2} = \textbf{AACoder2}(\text{fNameIn})$$

είναι υπεύθυνη για την κωδικοποίηση του αρχείου ήχου, όπως αντίστοιχα και η **AACoder1** ήταν για την προηγούμενη βαθμίδα, λαμβάνοντας όμως αυτήν φορά υπόψιν την βαθμίδα **TNS**. Αξίζει να αναφέρουμε πως για την υλοποίηση της βαθμίδας αυτής αξιοποιήθηκαν και οι συναρτήσεις που αναπτύχθηκαν στο Επίπεδο 1.

Ενώ όπως είναι πλέον αναμενόμενο, αναγκαία ήταν και η ανάπτυξη της συνάρτησης

$$\text{x} = \textbf{iAACoder2}(\text{AACSeq2}, \text{fNameOut})$$

η οποία αντιστρέφει την λειτουργία της **AACoder2**.

Τέλος, όπως και προηγουμένως, για να αξιολογήσουμε την δουλειά μας κατασκευάσαμε την

$$\text{SNR} = \textbf{demoAAC2}(\text{fNameIn}, \text{fNameOut})$$

η οποία υπολογίζει την σηματοθορυβική σχέση του αρχικού και τελικού ήχου, και αποδεικνύει παράλληλα την επιτυχή εκτέλεση του δεύτερου σκέλους της εργασίας.

```
Level 2
=======
Coding: time ellapsed is 1.507928 seconds
Decoding: time ellapsed is 0.330891 seconds
Channel 1 SNR: 301.513728 dB
Channel 2 SNR: 301.610385 dB
```

Fig.3 Using KBD window

```
Level 2
=======
Coding: time ellapsed is 2.340932 seconds
Decoding: time ellapsed is 0.354923 seconds
Channel 1 SNR: 307.421254 dB
Channel 2 SNR: 307.499990 dB
```

Fig.4 Using KBD window

Επίπεδο 3

Στα προηγούμενα επίπεδα έρχεται το τρίτο και τελευταίο στάδιο ανάπτυξης να προσθέσει με τη σειρά του το ψυχοακουστικό μοντέλο (Psychoacoustic model) το οποίο είναι υπεύθυνο για τη μείωση των απαιτούμενων bit του κβαντιστή, υπολογίζοντας τα κατώφλια ακουστότητας για κάθε συχνότητα. Η υλοποίηση της συνάρτησης στηρίχθηκε στην υπο-ενότητα 2.4 της εκφώνησης, καθώς και στο αρχείο w2203tfa. Η συνάρτηση αυτή είναι η εξής:

$$\mathrm{SMR} = \mathbf{psycho}(\mathrm{frameT}, \mathrm{frameType}, \mathrm{frameTprev1}, \mathrm{frameTprev2})$$

η οποία όπως είπαμε υπολογίζει τα κατώφλια ακουστότητας για κάθε συχνότητα, για ένα κανάλι. Τα ορίσματα της είναι το frame το οποίο εξετάζεται, ο τύπος του συγκεκριμένου frame, καθώς και τα δύο προηγούμενα frames (256 συντελεστών για frames τύπου ESH και 2048 συντελεστών για τους υπόλοιπους τύπους), ενώ η συνάρτηση επιστρέφει έναν πίνακα με το Signal to Mask Ratio, για κάθε μπάντα (41x8 στοιχείων για frame τύπου ESH και 68x1 για όλα τα υπόλοιπα frames). Για τη δημιουργία των πινάκων της spreading function, δημιουργήθηκε η συνάρτηση **spreadingfun_tables()**, η οποία επιστρέφει δύο πίνακες, τους **spread_long** 68x68 στοιχείων και **spread_short** 41x41 στοιχείων, οι οποίοι περιέχουν τους συνδυασμούς της συνάρτησης για όλες τις μπάντες για long και short παράθυρα αντίστοιχα.

Στη συνέχεια δημιουργήθηκε η συνάρτηση του κβαντιστή, ο οποίος χρησιμοποιώντας του συντελεστές της βαθμίδας TNS από το δεύτερο επίπεδο και τα κατώφλια ακουστότητας για κάθε μπάντα, παράγει σύμβολα τα οποία θα κωδικοποιηθούν στη συνέχεια από τη βαθμίδα Huffman. Η συνάρτηση αυτή υλοποιήθηκε ακολουθώντας πιστά τα βήματα που περιγράφονται στην υπο-ενότητα 2.6 της εκφώνησης και στο αρχείο του προτύπου. Η συνάρτηση αυτή είναι η εξής:

$$[\mathrm{S}, \mathrm{sfc}, \mathrm{G}] = \mathbf{AACquantizer}(\mathrm{frameF}, \mathrm{frameType}, \mathrm{SMR})$$

η οποία δέχεται ως ορίσματα το frame στο πεδίο της συχνότητας **frameF** υπό τη μορφή συντελεστών MDCT, τον τύπο του frame, **frameType** και τον πίνακα που περιέχει το Signal to Mask Ratio, **SMR**, ο οποίος υπολογίζεται από το ψυχοακουστικό μοντέλο. Η συνάρτηση επιστρέφει έναν πίνακα **S** 1024x1 στοιχείων με τα σύμβολα κβάντισης των συντελεστών MDCT του τρέχοντος frame, έναν πίνακα **sfc** (41x8 στοιχείων για frames τύπου ESH και 68x1 στοιχείων για όλα τα υπόλοιπα frames), ο οποίος περιέχει τις τιμές των συντελεστών scalefactor για κάθε scalefactor band και το Global Gain **G** του τρέχοντος frame (1x8 στοιχείων για frames τύπου ESH και 1 συντελεστή για όλα τα υπόλοιπα frames).

Τέλος αναπτύχθηκε και η συνάρτηση

$$\mathrm{frameF} = \mathbf{iAACquantizer}(\mathrm{S}, \mathrm{sfc}, \mathrm{G}, \mathrm{frameType})$$

η οποία με τη σειρά της αντιστρέφει την προηγούμενη συνάρτηση του κβαντιστή. Η συνάρτηση αυτή δέχεται ως ορίσματα τις εξόδους της συνάρτησης του κβαντιστή, καθώς και τον τύπο

του frame και επιστρέφει το frame στο πεδίο της συχνότητας υπό τη μορφή συντελεστών MDCT.

Στη συνέχεια του προτύπου χρησιμοποιούνται και οι συναρτήσεις encodeHuff και decodeHuff, οι οποίες είναι υπεύθυνες για την κωδικοποίηση και την αποκωδικοποίηση αντίστοιχα εντροπίας, χρησιμοποιώντας κώδικα Huffman. Οι συναρτήσεις αυτές δίνονται έτοιμες. Δυστυχώς στο τρίτο επίπεδο δεν υπήρξε ανάπτυξη του κωδικοποιητή και αποκωδικοποιητή, ώστε να λάβουμε αποτελέσματα για το σηματοθορυβικό λόγο, το bitrate και τη συμπίεση αντίστοιχα, λόγω έλλειψης χρόνου.

Συμπεράσματα

Όσον αφορά τα δύο πρώτα επίπεδα στα οποία έγινε πλήρης ανάπτυξη, μπορούμε να δούμε εμφανώς μέσω του σηματοθορυβικού λόγου, αλλά και ακουστικά ότι υπάρχει πλήρης ανακατασκευή του αρχικού σήματος έχοντας ελάχιστη απώλεια. Η πλήρης ανακατασκευή φαίνεται υπολογίζοντας επίσης και το μέσο τετραγωνικό σφάλμα ανάμεσα στο αρχικό σήμα και το ανακατασκευασμένο. Συγκεκριμένα για το πρώτο επίπεδο το σφάλμα είναι $2.4479*10^{-32}$, ενώ για το δεύτερο $2.512*10e^{-32}$, πολύ μικρά όπως παρατηρούμε. Επίσης ο χρόνος κωδικοποίησης και αποκωδικοποίησης του σήματος για το πρώτο και δεύτερο επίπεδο είναι στα ίδια επίπεδα με αυτά που δίνονται ως πρότυπα.