Μοσχόπουλος Βασίλειος 9105

Εργασία - Συστήματα Πολυμέσων

Υλοποίηση AAC Encoder-Decoder

Περίληψη

Στα πλαίσια της φετινής εργασίας του μαθήματος “Συστήματα Πολυμέσων” υλοποιήθηκε μια απλοποιημένη έκδοση του προτύπου ήχου ***AAC.*** Η εργασία, πιο συγκεκριμένα, αποτελείται από 3 μέρη. Στο πρώτο έγινε η υλοποίηση των modules **Filterbank** και **SSC** (Sequence Segementation Control) για την κωδικοποίηση, και των αντίστροφών τους για χρήση κατά την αποκωδικοποίηση. Στη συνέχεια, ζητούμενο ήταν η δημιουργία ενός module που εκτελεί **TNS** (Temporal Noise Shaping) στους συντελεστές που προκύπτουν απ’ το MDCT και τέλος χρειάστηκε να κατασκευαστεί μια βαθμίδα **κβαντιστή**, η οποία ρυθμίζεται εν μέρει από μια βαθμίδα που υλοποιεί το ψυχοακουστικό μοντέλο. Σημείωση: Στο τέλος παρουσιάζεται μια συνοπτική περιγραφή για κάθε συνάρτηση που βρίσκεται στα αρχεία του project.

Πρώτο Μέρος (Filterbanks, SSC)

Στα πλαίσια της υλοποίησης του προτύπου, έγινε χρήση επικαλυπτόμενων κατά 50% (ανά δυο) παραθύρων, όπως αναφέρεται στην εκφώνηση, ενώ στο αρχείο ήχου εισόδου γίνεται επιπλέον zero-padding στην αρχή και στο τέλος του σήματος. Αναλυτικότερα, γίνεται pad 1024 μηδενικών στην αρχή και στο τέλος του σήματος (για βελτιώση της εκτίμησης των filterbanks στα άκρα) και στη συνέχεια γίνεται επιπλέον zero-padding στο τέλος του σήματος (εφόσον χρειαστεί) μέχρις ότου το μήκος του σήματος γίνει πολλαπλάσιο 2048 δειγμάτων, ώστε να μπορεί να εφαρμοστεί η παραπάνω μέθοδος με τα frames.

Λόγω του γεγονότος ότι η αρχική υλοποίηση μου για το MDCT (for loop για τα σημεία της εξόδου) έκανε πολύ αργή τη χρονική διάρκεια της συνολικής διαδικασίας (καταναλώνοντας το μεγαλύτερο μέρος της κωδικοποίησης, αλλά και της αποκωδικοποίησης, μέσω του Inverse MDCT), αναζητήθηκε στο διαδίκτυο κάποια πιο γρήγορη υλοποίηση (δε χρησιμοποιήθηκε η built-in MDCT της Matlab γιατί ζητώντας ως όρισμα τον τύπο του παραθύρου, ουσιαστικά άφηνε αχρησιμοποίητη την κατασκευή κάθε τύπου παραθύρου που πραγματοποιείται στο αρχείο getWindow.m). Πιο συγκεκριμένα, με μια πρόχειρη αναζήτηση χρησιμοποιήθηκε αυτή η υλοποίηση, που, παραδόξως, οδήγησε και σε αύξηση του SNR στην ανακατασκευή του αρχείου ήχου (παράδοξο διότι ακολουθήθηκαν βήμα-βήμα πιστά οι τύποι για το MDCT-IMDCT που δόθηκαν στην εκφώνηση). Στα αρχεία mdct.m και imdct.m πάντως υπάρχουν και οι δύο εκδόσεις του μετασχηματισμού και μπορεί να γίνει χρήση και των 2 (naive – fast) κάνοντας comment out την αντίστοιχη συνάρτηση. Συμπερασματικά, βλέπουμε ότι επαληθεύεται ο μη-lossy χαρακτήρας της διαδικασίας αφού οι τιμές SNR που εμφανίζονται στο output σήμα (~250dB για το naive MDCT και ~300dB για τον fast) δείχνουν τη σχεδόν τέλεια ανακατασκευή του σήματος.

Δεύτερο Μέρος (Temporal Noise Shaping)

Στο δεύτερο μέρος της εργασίας ζητήθηκε η υλοποίηση μια βαθμίδας TNS (Temporal Noise Shaping) η οποία βρίσκεται αμέσως μετά το Filterbank, απ' όπου λαμβάνει τους συντελεστές MDCT του σήματος και απαλείφει τις όποιες περιοδικότητες εμφανίζουν αυτοί. Πιο αναλυτικά, για κάθε frame, πρώτα γίνεται μια κανονικοποίηση των συντελεστών με βάση τη σχέση (3) (σελ. 9), χρησιμοποιώντας τις σχέσεις (3) και (4) και τους πίνακες B.2.1.9a.b (για long και short παράθυρα, αντίστοιχα) που δόθηκαν και περιγράφουν διάφορα στοιχεία των μπαντών και συνδέουν αυτές με τα *X(k)*. Αφού γίνει το smoothing των συντελεστών $S_w(k)$, υπολογίζονται οι συντελεστές α_i (i=1,2,3,4) με τη μέθοδο των ελαχίστων τετραγώνων (λύνεται το σύστημα **R*α=r** ως προς α) και κατόπιν γίνεται κβαντισμός των συντελεστών αυτών (*TNNcoeffs*) όπου φιλτράρονται γραμμικά με αυτούς οι αρχικοί συντελεστές *X(k)*. Ο κβαντισμός γίνεται με τον κβαντιστή του σχήματος. Στην πλευρά του αποκωδικοποιητή, από την άλλη, λαμβάνονται οι μετασχηματισμένοι συντελεστές και τα *TNScoeffs* και λαμβάνονται οι αρχικοί φιλτράροντας τους με το αντίστροφο φίλτρο σε σχέση με πριν (*1/H(z)*). Και σε αυτή τη βαθμίδα παρατηρείται σχεδόν πλήρης ανακατασκευή του αρχικού σήματος με τιμές SNR~300dB

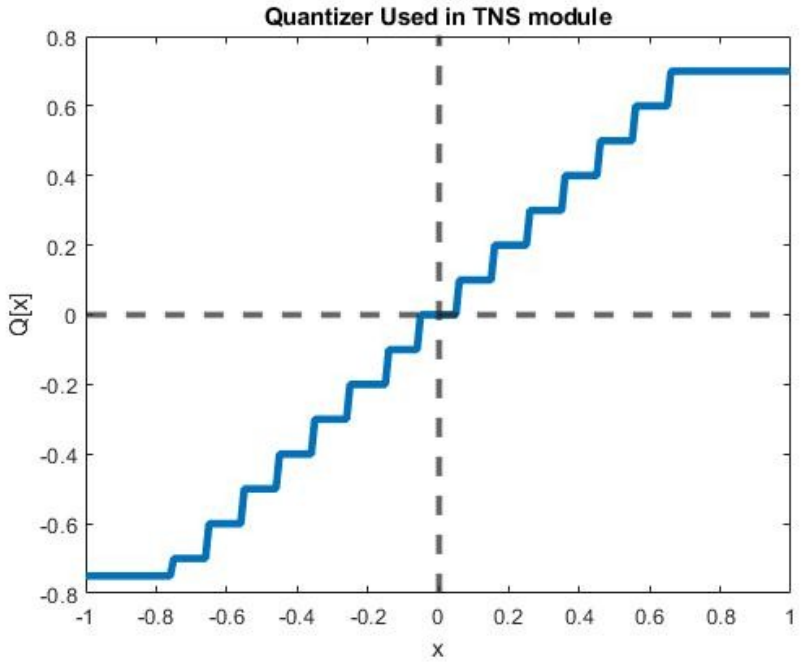

Figure 1: Κβαντιστής Συντελεστών TNS

Τρίτο Μέρος (Psychoacoustics,Quantization,Huffman)

Στο τρίτο και τελευταίο μέρος της εργασίας είχε ως στόχο τη δημιουργία του κβαντιστή του προτύπου, το οποίο κβαντίζει τους συντελεστές που παράγει ο MDCT στο βαθμό που του επιτρέπεται από το ψυχοακουστικό μοντέλο. Κατόπιν, οι κβαντισμένες τιμές κωδικοποιούνται κατά Huffman.

Για τη συνάρτηση psycho() ακολουθήθηκαν τα βήματα των σελίδων 10,11,12,13, ενώ για τη βαθμίδα του κβαντισμού με αφετηρία το scalefactor estimation που γίνεται στη σχέση (14) και σταδιακή αύξηση κατά μιας μονάδας σε όσα scalefactors (κάθε frame ή subframe) επιτρέπεται κάθε φορά (πρέπει το σφάλμα που εισάγει ο κβαντισμός να είναι κάτω απ’ το κατώφλι ακουστότητας), λαμβάνονται τα τελικά scalefactors και οι κβαντισμένοι συντελεστές, ενώ στην αποκωδικοποίηση, αντίστροφα, γίνεται η ανάκτηση των εκτιμήσεων των συντελεστών αυτών με τη χρήση της σχέσης (13).

Κάτι που παρατηρήθηκε, επίσης, κατά τον υπολογισμό του ψυχοακουστικού μοντέλου είναι ότι ενώ αναφέρεται οτι οι τιμές tonality index (tb) πρέπει να είναι ανάμεσα σε 0 και 1, κάποιες φορές λάμβαναν αρνητικές τιμές. Έτσι σε αυτές τις περιπτώσεις, οι αρνητικές τιμές tonality index αποφασίστηκε να τεθούν ίσες με 0, κάτι που αύξησε κατά λίγο το SNR σε σχέση με πριν την παρέμβαση. Σχετικά με το bitrate και το compression που επιτυγχάνεται, γίνεται σύγκριση μεταξύ ενός AACSeq2 κι ενός AACSeq3 αντικειμένου (για κάθε τιμή οποιασδήποτε δομής θεωρήθηκε ότι είναι τύπου double καθώς αυτή την αναπαράσταση χρησιμοποίησε το Matlab). Παρόλα αυτά, παρατηρήθηκε ότι το αρχείο “AACodedName”.mat στο οποίο αποθηκεύεταιτο αντικείμενο AACSeq3 είναι ακόμα πιο συμπυκνωμένο σε μέγεθος bytes, συνεπώς μπορεί να θεωρηθεί ότι πετυχαίνεται ακόμα μεγαλύτερο compression από αυτό των αποτελεσμάτων που εμφανίζονται κατά την εκτέλεση.

Αποτελέσματα

Παρακάτω παρουσιάζονται κάποια αποτελέσματα από την εκτέλεση των εκδόσεων των κωδίκων για το sample αρχείο “LicorDeCalandraca.wav”. Μετρείται το SNR που επιστρέφεται από τη συνάρτηση demo (για την ακρίβεια επιστρέφεται SNR για κάθε ξεχωριστό κανάλι, αλλά εδώ εμφανίζεται μόνο για το αριστερό, το δεξι επιστρέφει παραπλήσιες τιμές), αλλά και ο χρόνος εκτέλεσης της κάθε έκδοσης, ενώ στο τέλος γίνεται μια παρουσίαση του χρόνου encoding που εμφανίζει μια λογική αύξηση λόγω του compression που γίνεται.

	Part 1	Part 2	Part 3 (Unclipped)	Part 3 (tb clip)
SNR	306.4 dB	306.3 dB	6.31 dB	6.77 dB

Table 1: Ποιότητα ανακατασκευασμένου σήματος (Fast MDCT, KBD window)

	Part 1	Part 2	Part 3 (Unclipped)	Part 3 (tb clip)
SNR	307.6 dB	307.63 dB	6.24 dB	6.65 dB

Table 2: Ποιότητα ανακατασκευασμένου σήματος (Fast MDCT, SIN window)

	Part 1	Part 2	Part 3
Encoding Times	0.34 sec	2.73 sec	81.3 sec

Table 3: Ενδεικτικοί χρόνοι encoding σε υπολογιστή με i7-4720HQ (Fast MDCT, KBD window)

Συνοπτική περιγραφή συναρτήσεων

- Όσες συναρτήσεις περιγράφονται στα ζητούμενα της εργασίας, επιτελούν ακριβώς αυτά που αναφέρονται στην εκφώνηση με ορίσματα, επίσης, αυτά που περιγράφονται στο pdf.
- padinput(x): Επιστρέφει την είσοδο, αφού έχει γίνει zero-padding στα άκρα του σήματος, με βάση τη μεθοδολογία που περιγράφηκε παραπάνω
- getWindow(frameType,winType): Επιστρέφει το παράθυρο που αντιστοιχεί στα ορίσματα που δίνονται (επιλογές frameType: “OLS”,”LSS”,”LPS”,”ESH” , επιλογές winType: “KBD”,”SIN”)
- getKaiser(N,alpha), lssKBD(), lpsKBD(), lssSIN(), lpsSIN(): βοηθητικές συναρτήσεις της getWindow() που καθέ μια από αυτές επιστρέφει το αντίστοιχο παράθυρο
- checkNextFrame(): Έχει ως είσοδο ένα μονοκαναλικό frame και επιστρέφει αν πρέπει να είναι Eight Short ή Only Long, με βάση τη μεθοδολογία που περιγράφεται στην εκφώνηση (φιλτράρισμα, έλεγχος attackValues)
- decideType(): Δεχόμενη ως είσοδο 2 frameTypes (των δύο καναλιών), αποφασίζει τι τύπο πρέπει να έχει το frame συνολικά, εφαρμόζοντας τη λογική του Πίνακα 1 (σελ. 5)
- ByteSize(): Συνάρτηση που βρίσκει το μέγεθος ενός αντικειμένου σε bytes (πηγή)
- quantize(): Είναι ο κβαντιστής που χρησιμοποιείται για την TNS βαθμίδα (σχήμα)
- quantV3(): Υλοποιεί τον κβαντισμό της σχέσης (12)
- iquantV3(): Υλοποιεί τον απο-κβαντισμό της σχέσης (13)

Σχόλιο για την εκτέλεση του κώδικα

Ο κώδικας σε κάθε υποφάκελο του αρχείου zip μπορεί να εκτελεστεί ανεξάρτητα από τους υπόλοιπους. Γι’ αυτό το λόγο επαναλαμβάνονται κάποια αρχεία, (όπως το filterbank.m), καθώς είναι απαραίτητα και για επόμενα στάδια από αυτό στο οποίο υλοποιήθηκαν. Η εναλλαγή μετάξυ της χρήσης “KBD” και “SIN” παραθύρων γίνεται στο πάνω μέρος του αρχείου AACoderX.m (X=1,2,3) . Για την εκτέλεση κάθε έκδοσης πρέπει να το Matlab να έχει ως current folder το φάκελο της έκδοσης (ή να το συμπεριλαμβάνει στο Path). Παραδείγματα κλήσεων που δοκιμάστηκαν είναι:

```
>> SNR = demoAAC1('LicorDeCalandraca.wav','LicorDeCalandraca_Decoded.wav');
>> SNR = demoAAC2('LicorDeCalandraca.wav','LicorDeCalandraca_Decoded.wav');
>> [SNR,bitrate,compression] =
demoAAC3('LicorDeCalandraca.wav','LicorDeCalandraca_Decoded.wav',’test.mat');
```