Συστήματα Πολυμέσων Εργασία

Απλοποιημένος codec MP3

Α. Ντελόπουλος

2022-2023

1 Εισαγωγικές Παρατηρήσεις

Η παρακάτω εργασία αποτελεί *προαιρετικό* μέρος του μαθήματος Συστήματα Πολυμέσων και η εκτέλεσή της συνεισφέρει 1 έως 4 επιπλέον μονάδες στην τελική βαθμολογία.

Η εργασία θα πρέπει να εκτελεστεί σε ομάδες των δύο ατόμων.

Η εργασία αποτελείται από 6 ενότητες. Η υλοποίηση της εργασίας μπορεί κατ' επιλογή να περιλάβει την πρώτη ενότητα, την πρώτη και τη δεύτερη, κ.ο.κ. συνεισφέροντας τις αντίστοιχες μονάδες για κάθε ενότητα. Δεν μπορεί όμως να εκτελεστεί μία ενότητα χωρίς να έχει ορθά εκτελεστεί η προηγούμενή της.

Κάθε ενότητα της εργασίας αφορά στην υλοποίηση μίας βασικής λειτουργίας ενός κωδικοποιητή/αποκωδικοποιητή MPEG-1 Layer III.

2 Απλουστευμένη Κωδικοποίηση mp3

Η προτεινόμενη εργασία στοχεύει στην υλοποίηση ενός απλουστευμένου κωδικοποιητή/αποκωδικοποιητή ήχου κατά το πρότυπο MPEG-1 Layer III που περιλαμβάνει τα ακόλουθα στάδια:

1. Χωρισμό του σήματος σε υποζώνες συχνότητας (frequency subbands).
2. Εφαρμογή DCT.
3. Εφαρμογή του ψυχοακουστικού μοντέλου.
4. Κβαντισμό (και αποκβαντισμό) των συντελεστών DCT.
5. Κωδικοποίηση (και αποκωδικοποίηση) μήκους διαδρομής (Run Length Encoding)
6. Κωδικοποίηση (και αποκωδικοποίηση) Huffman.

Μία συνοπτική περιγραφή της διαδικασίας φαίνεται στο παρακάτω σχήμα.

Στη συνέχεια παρουσιάζεται αναλυτικά η λειτουργία κάθε μιας από τις παραπάνω βαθμίδες.

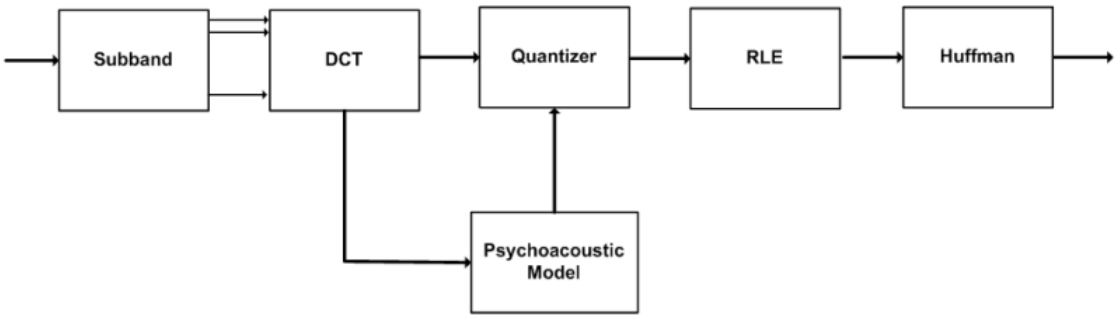

Figure 1: Η προτεινόμενη μεθοδολογία

2.1 Διαχωρισμός του σήματος σε subbands

Το πρώτο βήμα συνίσταται στην υποδιαίρεση του φάσματος του σήματος εισόδου σε 32 ίσου εύρους τμήματα. Χρησιμοποιείται μία συστοιχία απο $M = 32$ ψηφιακά φίλτρα με κρουστική απόκριση $h_i(n)$, $(i = 0, ..., M - 1)$ μήκους $L = 512$. Στο πεδίο της συχνότητας οι αντίστοιχες συναρτήσεις μεταφοράς παρουσιάζουν ζωνοπερατή συμπεριφορά με εύρος διέλευσης $B_d \approx \pi/M$ (rad) ή ισοδύναμα

$$B = B_d \frac{f_s}{2\pi} = \frac{f_s/2}{M} \approx 689Hz \tag{1}$$

για συχνότητα δειγματοληψίας $f_s = 44100\, samples/sec$.

Οι $M = 32$ διαφορετικές κρουστικές αποκρίσεις παράγονται με διαμόρφωση μίας πρότυπης κρουστικής απόκρισης $h(n)$, που αντιστοιχεί σε βαθυπερατό φίλτρο, σύμφωνα με τη σχέση:

$$h_i(n) = h(n) cos(\frac{(2i+1)\pi}{2 * M} n + \frac{(2i+1)\pi}{4}) \tag{2}$$

Οι συντελεστές της πρότυπης $h(n)$ προσδιορίζονται από το πρότυπο MPEG.

Η έξοδος κάθε φίλτρου υποδειγματοληπτείται με ρυθμό $1 : M$. Συνεπώς για σήμα εισόδου $x(n)$ παράγονται M σήματα subband από τη σχέση:

$$y_i(n) = \sum_k h_i(k) x(Mn - k), i = 0, ..., M - 1 \tag{3}$$

Μία συλλογή από $N = 36$ δείγματα (μετά και από την υποδειγματοληψία) από κάθε μπάντα ($i = 0, ..., M - 1$) ονομάζεται frame. Συνεπώς κάθε frame περιέχει $M \times N = 32 \times 36 = 1152$ δείγματα. Τα υπόλοιπα βήματα κωδικοποίησης εκτελούνται σε μεμονωμένα διαδοχικά frames.

Στην πλευρά του αποκωδικοποιητή το σήμα $x(n)$ αναπράγεται μέσω παρεμβολής - φιλτραρίσματος - υπέρθεσης από τη σχέση:

$$\hat{x}(n) = \sum_{i=0}^{M} r_i(n), \tag{4}$$

όπου

$$w_i(n) = \begin{cases} r_i(n/M), & n = kM \\ 0, & \text{αλλιώς} \end{cases} \tag{5}$$

$$r_i(n) = \sum_k g_i(n-k)w_i(k) = \sum_j g_i(n-jM)r_i(j), \tag{6}$$

όπου τα M φίτρα σύνθεσης $g_i(n)$ επιλέγονται έτσι ώστε σε συνδυασμό με τα $h_i(n)$ να επιτρέπουν σχεδόν τέλεια ανακατασκευή: $\hat{x}(n) \approx x(n)$

2.2 Εφαρμογή DCT

Στην απλουστευμένη έκδοση του κωδικοποιητή που εξετάζουμε τα $N = 36$ δείγματα κάθε μπάντας ενός frame μετασχηματίζονται κατά DCT παράγοντας ισάριθμους συντελεστές $c(u)$, $u = 0, ..., N-1$.

Ο υπολογισμός του DCT διαμερίζει κάθε μπάντα σε 36 ισαπέχοντα κομμάτια. Επομένως ο συντελεστής υπ' αριθμόν u, της μπάντας i αντιστοιχεί στη συχνότητα του αναλογικού σήματος

$$f = iB + u\frac{B}{N}(Hz) \tag{7}$$

Ισοδύναμα, αν τοποθετήσουμε τους DCT συντελεστές της μίας μπάντας μετά την άλλη και τους αριθμήσουμε με ενιαίο τρόπο, δηλαδή

$$k = i \cdot N + u, \tag{8}$$

τότε ο συντελεστής αρίθμησης k αντιστοιχεί στη συχνότητα

$$f = k\frac{B}{N} = k\frac{f_s}{2MN}(Hz) \tag{9}$$

του αναλογικού σήματος.

Στον αποκωδικοποιητή χρησιμοποιείται ο αντίστροφος μετασχηματισμός DCT για την αναπαραγωγή των δειγμάτων κάθε μπάντας στο πεδίο του χρόνου.

2.3 Ψυχοακουστικό Μοντέλο

Για να υπολογίσουμε τον αριθμό των bits που θα χρησιμοποιηθούν στην κωδικοποίηση των δειγματων του σήματος, θα πρέπει να υπολογιστεί το κατώφλι ακουστότητας για κάθε συχνότητα. Το κατώφλι καθορίζει τη μέγιστη επιτρεπτή ενέργεια του θορύβου κβαντισμού για κάθε συντελεστή DCT, έτσι ώστε η προκύπτουσα παραμόρφωση να μην είναι ακουστή. Αυτό θα γίνει με εφαρμογή του ψυχοακουστικού μοντέλου, όπως εξηγείται στη συνέχεια:

1. Εντοπίζονται οι συχνότητες στις οποίες εμφανίζονται τόνοι. Προσδιορίζεται δηλαδή ένα σύνολο S_T που περιέχει αυτές τις συχνότητες (tonal components ή maskers)

2. Ο αριθμός των maskers μειώνεται ώστε να χρησιμοποιηθούν στα επόμενα βήματα οι πλέον ισχυροί από αυτούς.

3. Για κάθε τόνο (k) που περιέχεται στο S_T, υπολογίζεται η συνεισφορά του $T_M(i,k)$ στο συνολικό κατώφλι $T_g(i)$.

4. Υπολογίζεται το συνολικό κατώφλι $T_g(i)$ ως το άθροισμα των επιμέρους συνεισφορών και του κατωφλίου στη σιωπή $T_q(i)$.

Όλοι οι υπολογισμοί αφορούν κάθε συντελεστή $i = 0, ..., MN-1$ του DCT.
Πιο συγκεκριμένα, για κάθε βήμα:

1. Εντοπισμός και υπολογισμός της ισχύος των Tonal Components (maskers). Στο απλουστευμένο μοντέλο, ως εκτίμηση του φάσματος ισχύος του κάθε frame χρησιμοποιούνται οι αντίστοιχοι συντελεστές DCT. Οπότε το φάσμα ισχύος στη συχνότητα που αντιστοιχεί στον αριθμό k δίνεται από τη σχέση:

$$P(k) = 10log_{10}(|c(k)|^2) \tag{10}$$

όπου $c(k)$ ο συντελεστής DCT υπ' αριθμόν k.

Με βάση την ισχύ $P(k)$ κάθε συντελεστή γίνεται ο υπολογισμός μιας αρχικής έκδοσης του συνόλου S_T από τη σχέση:

$$S_T = \left\{ k \;\middle|\; \begin{array}{c} P(k) > P(k \pm 1), \\ P(k) > P(k \pm \Delta_k) + 7dB \end{array} \right\} \tag{11}$$

όπου:

$$\Delta_k \in \begin{cases} 2 & 2 < k < 282 \quad (0.17 - 5.5)\, kHz \\ [2, 13] & 282 \leq k < 570 \quad (5.5 - 11)\, kHz \\ [2, 27] & 570 \leq k < 1152 \quad (11 - 22)\, kHz \end{cases} \tag{12}$$

Η ισχύς των maskers υπολογίζεται στη συνέχεια από τη σχέση:

$$P_M(k) = 10log_{10} \sum_{j=-1}^{1} 10^{0.1P(k+j)}(dB), \forall k \in S_T. \tag{13}$$

Η 13 μεριμνά ώστε σε κάθε τόνο να αποδοθεί η συνολική ισχύς του που, λόγω μειωμένης ευκρίνειας στο διακριτό φάσμα συχνοτήτων, μπορεί να έχει μοιραστεί σε μια γειτονιά $\{k-1, k, k+1\}$ του συντελεστή DCT υπ' αριρθμόν k.

2. Ελάττωση των maskers. Αρχικά, στο βήμα αυτό κρατάμε ως maskers μόνο αυτά που ικανοποιούν τη σχέση

$$P_M(k) \geq T_q(k) \tag{14}$$

με $T_q(k)$ το κατώφλι της μη ακουστότητας στη σιωπή για τη συχνότητα k. Όσα k δεν πληρούν το παραπάνω κριτήριο εξαιρούνται από το S_T.

Στη συνέχεια ελέγχεται η απόσταση μεταξύ των maskers. Η απόσταση εκφράζεται σε barks (αντι για Hz). Η συχνότητα z σε barks δίνεται από τη συχνότητα f σε Hz μέσω της σχέσης:

$$z(f) = 13arctan(0.00076f) + 3.5arctan((\frac{f}{7500})^2)(Bark) \tag{15}$$

Στην κλίμακα των barks η απόσταση μεταξύ δύο συχνοτήτων αντικατορπτρίζει καλύτερα τη συχνοτική διαφορά όπως την αντιλαμβάνεται η ανθρώπινη ακοή.

Αν η απόσταση μεταξύ δύο masker είναι μικρότερη από 0.5 bark ο μικρότερος από τους δύο εξαιρείται από το S_T.

3. Υπολογισμός των masking thresholds: Για κάθε masker $k\,(k \in S_T \subseteq [0, ..., MN-1])$ που έχει απομείνει στο S_T υπολογίζεται η συνεισφορά του στο κατώφλι ακουστότητας της συχνότητας i από τη σχέση:

$$T_M(i,k) = P_M(k) - 0.275z(f_k) + SF(i,k) - 6.025(dB) \tag{16}$$

όπου η συνάρτηση $SF(i,k)$ ονομάζεται spreading function και δίνεται από την έκφραση:

$$SF(i,k) = \begin{cases} 17\Delta_z - 0.4P_M(k) + 11, & -3 \leq \Delta_z < -1 \\ (0.4P_M(k) + 6)\Delta_z & -1 \leq \Delta_z < 0 \\ -17\Delta_z & 0 \leq \Delta_z < 1 \\ (0.15P_M(k) - 17)\,\Delta_z - 0.15P_M(k) & 1 \leq \Delta_z < 8 \end{cases} \quad (dB) \tag{17}$$

και $\Delta_z = z(f_i) - z(f_k)$, όπου f_i και f_k οι συχνότητες του αρχικού αναλογικού σήματος που αντιστοιχούν στους συντελεστές DCT υπ' αρθμόν i και k αντίστοιχ (βλ. Σχέση 9).

4. Τελικά υπολογίζεται για κάθε συντελεστή i το global masking threshold: Το συνολικό κατώφλι ακουστότητας για το συντελεστή (συχνότητα) i υπολογίζεται από την υπέρθεση των επιμέρους κατωφλίων ως:

$$T_g(i) = 10log_{10}(10^{0.1T_q(i))} + \sum_{k \in S_T} 10^{0.1T_M(i,k)})(dB) \tag{18}$$

2.4 Κβαντισμός/Αποκβαντισμός Συντελεστών DCT

Για την επίτευξη συμπίεσης, κβαντίζουμε τους συντελεστές DCT εφαρμόζοντας τα αποτελέσματα της προηγούμενης ανάλυσης. Η ακολουθούμενη διαδικασία είναι:

1. Ομαδοποιούμε τους συντελεστές σε critical bands.
2. Για κάθε critical band, οι συντελεστές DCT κανονικοποιούνται και κβαντίζονται με τον ίδιο ομοιόμορφο κβαντιστή.
3. Χρησιμοποιούμε το κριτήριο του ψυχοακουστικού μοντέλου για να επιλέξουμε τον αριθμό ($N = 2^b$) των συμβόλων του κβαντιστή με b bits.

Αναλυτικότερα:

1. Οι συχνότητες σε critical bands με βάση τον πίνακα του σχήματος 2. Παρατηρείστε ότι τα διαστήματα για κάθε critical band είναι έτσι επιλεγμένα ώστε το $z(f_{max}) - z(f_{min})$ να είναι περίπου το ίδιο για όλες τις μπάντες.
2. Για κάθε συντελεστή DCT $c(i)$ κωδικοιποιείται η παράσταση

$$\tilde{c}(i) = sign(c(i)) \frac{|c(i)|^{3/4}}{Sc(band)}, \tag{19}$$

όπου $Sc(band) = max(|c(i)|^{3/4})$ (με $i \in$ στη μπάντα band). Οι τιμές αυτές ονομάζονται scale factors και αποθηκεύονται μαζί με το κωδικοποιημένο frame. Λόγω της κανονικοποίησης οι συντελεστές $\tilde{c}(i)$ λαμβάνουν τιμές στο διάστημα $[-1, 1]$.

3. Ο αριθμός των σταθμών ($N = 2^b - 1$) του κβαντιστή υπολογίζεται ως εξής: Αρχικά υποθέτουμε ότι επιλέγεται ομοιόμορφος [1] κβαντιστής 1 bit. Για κάθε συντελεστή $\tilde{c}(i)$, που ανήκει στην critical band, υπολογίζεται η αποκβαντισμένη τιμή $\hat{\tilde{c}}(i)$ και αντίστοιχος αποκβαντισμένος συντελεστής

$$\hat{c}(i) = sign(\hat{\tilde{c}}(i))(\hat{\tilde{c}}(i) \times Sc(band))^{4/3} \tag{20}$$

και το σφάλμα κβαντισμού $e_b(i) = |c(i) - \hat{c}(i)|$. Στη συνέχεια υπολογίζεται **η ισχύς του θορύβου κβαντισμού** σε dB: $P_b(i) = 10log_{10}(e_b(i)^2)$ και αν για κάθε συντελεστή της ομάδας ισχύει ότι το $P_b(i) \leq T_g(i)$, τότε ο αριθμός των bits θεωρείται επαρκής. Αλλιώς αυξάνουμε το b κατά 1 και επαναλαμβάνουμε τη διαδικασία μέχρι να ικανοποιηθεί η συνθήκη.

Όσον αφορά στον κβαντιστή, αντί για 2^b χρησιμοποιούνται $2^b - 1$ ζώνες με την υπόθεση ότι οι ζώνες εκατέρωθεν του 0 ενοποιούνται σε μία. Επομένως, οι στάθμες απόφασης του κβαντιστή απέχουν γενικά κατά

$$w_b = \frac{1}{2^b - 1} \tag{21}$$

[1] Παρόλο που ο κβαντιστής είναι ομοιόμορφος για τα $\tilde{c}(i)$, δεν είναι και για τα $c(i)$.

Band No.	Center Freq. (Hz)	Bandwidth (Hz)
1	50	-100
2	150	100-200
3	250	200-300
4	350	300-400
5	450	400-510
6	570	510-630
7	700	630-770
8	840	770-920
9	1000	920-1080
11	1370	1270-1480
12	1600	1480-1720
13	1850	1720-2000
14	2150	2000-2320
15	2500	2320-2700
16	2900	2700-3150
17	3400	3150-3700
18	4000	3700-4400
19	4800	4400-5300
20	5800	5300-6400
21	7000	6400-7700
22	8500	7700-9500
23	10,500	9500-12000
24	13,500	12000-15500
25	19,500	15500-

Figure 2: Εύρος των critical bands

εκτός από αυτές που είναι εκατέρωθεν του 0 που απέχουν κατά $2w_b$. Είναι δηλαδή:

$$d = [-1,\ -(2^{b-1}-1)w_b,\ -(2^{b-1}-2)w_b,\ \ldots,\ -w_b, \tag{22}$$

$$+w_b,\ \ldots,\ +(2^{b-1}-2)w_b,\ -(2^{b-1}-1)w_b,\ +1] \tag{23}$$

Στον αποκβαντιστή χρησιμοποιούνται στάθμες που βρίσκονται στο μέσον της αντιστοίχης ζώνης απόφασης.

Για κάθε critical band φυλάσσεται το scale factor και αριθμός των bits που χρησιμοποιήθηκαν, ενώ στην επόμενη βαθμίδα παραδίδονται τα σύμβολα κβαντισμών.

2.5 Κωδικοποίηση / Αποκωδικοποίηση Μήκους Διαδρομής

Μετά τον κβαντισμό, τα παραγόμενα σύμβολα αναμένεται να περιέχουν συχνές και σχετικά μακριές ακολουθίες συμβόλων που αντιστοιχούν στη ζώνη του 0. Κάθε (μή μηδενικό σύμβολο) μαζί με τα τυχόν μηδενικά σύμβολα που ακολουθούν ομαδοποιούνται σε μήκη διαδρομής που κωδικοποιούνται ως νέα συνοπτικά σύμβολα κατά τα γνωστά.

2.6 Κωδικοποίηση Huffman

Τα σύμβολα που αναπαριστούν τα μήκη διαδρομής της προηγούμενης ενότητας κωδικοποιούνται κατά Huffman.

3 Παραδοτέα

Τα παραδοτέα αποτελέσματα της εργασίας σας είναι τριών ειδών: (α) Τα προγράμματα προσομοίωσης όπως αυτά περιγράφονται στις επόμενες ενότητες, (β) Τα προγράμματα επίδειξης που θα καλούν με κατάλληλο και πειστικό τρόπο τις προαναφερθείσες συναρτήσεις, καθώς και αντίστοιχα αρχεία ήχου (wav) που περιέχουν προϋπολογισμένο παράδειγμα κωδικοποίησης-αποκωδικοποίησης. Για τον έλεγχο του κωδικοποιητή-αποκωδικοποιητή κάθε επιπέδου που θα κατασκευαστεί, θα χρησιμοποιηθεί ενδεικτικό αρχείο ήχου (MYFILE.wav), το οποίο έχει δειγματοληπτηθεί στα 44.1KHz. (γ) Μία έκθεση (report) των επιλογών που έγιναν, και των παραδοχών που χρησιμοποιήθηκαν στην υλοποίηση. Στην ίδια έκθεση θα πρέπει να περιγράφεται ο τρόπος κλήσης των προγραμμάτων επίδειξης.

3.1 Subband filtering (1 μονάδα)

Για να παράγετε τις κρουστικές αποκρίσεις των σχεδιαζόμενων φίλτρων, χρησιμοποιήστε τη συνάρτηση $H = make\_mp3\_analysisfb(h, M)$ όπου h η κρουστική απόκριση του πρότυπου βαθυπερατού φίλτρου και M ο αριθμός των ζωνών στις οποίες θα γίνει ο διαχωρισμός. Η έξοδος H είναι πίνακας διάστασης $L \times M$, όπου L το μήκος της κρουστικής απόκρισης του φίλτρου, ο οποίος ως στήλες έχει τις κρουστικές αποκρίσεις των φίλτρων $h_i(n),\ i = 1, \ldots, M$ που έχουν υπολογιστεί από τη Σχέση (2).

Με αντίστοιχο τρόπο η συνάρτηση $G = make\_mp3\_synthesisfb(h, M)$ παράγει τα φίλτρα σύνθεσης $g_i(n)$.

Για διευκόλυνσή σας η διαδικασία ανάλυσης και σύνθεσης subband έχει υλοποιηθεί σε block μορφή και σας παρέχεται έτοιμη μέσω των συναρτήσεων $frame\_sub\_analysis(\,)$ και $frame\_sub\_synthesis(\,)$.

Ζητούμενα:

1. Να υπολογίσετε τους πίνακες H και G που περιέχουν τα $M = 32$ φίλτρα ανάλυσης και σύνθεσης χρησιμοποιώντας την πρότυπη κρουστική απόκριση $h(n)$ οι συντελεστές της οποίας περιέχονται στο αρχείο h.npy.

2. Να σχεδιάσετε (σε κοινό διάγραμμα) το μέτρο των συναρτήσεων μεταφοράς των φίλτρων $h_i(n)$ ως προς τη συχνότητα $f\ (Hz)$ σε μονάδες dB ($10 \log_{10}\left(|H(f)|^2\right)$).

3. Να σχεδιάσετε (σε κοινό διάγραμμα) το μέτρο των συναρτήσεων μεταφοράς των φίλτρων $h_i(n)$ ως προς τη συχνότητα $z\ (barks)$ σε μονάδες dB ($10 \log_{10}\left(|H(z)|^2\right)$).

4. Να υλοποιήσετε τη συνάρτηση $xhat, Ytot = codec0(wavin,\ h,\ M,\ N)$ που μέσω ενός πλήθους επαναλήψεων αναλύει σε μπάντες και στη συνέχεια ανασυνθέτει ένα σήμα ως ακολούθως:

   (a) διαβάζει σε κάθε επανάληψη MN δείγματα ενός ηχητικού σήματος $x(n)$, από το αρχείο εισόδου $wavin$ τύπου wav.

   (b) υπολογίζει ένα frame Y διάστασης $N \times M$ μέσω ανάλυσης σε μπάντες με χρήση των φίλτρων ανάλυσης που αντιστοιχούν στην πρότυπη κρουστική απόκριση h.

   (c) επεξεργάζεται το frame με τον αλγόριθμο $Yc = donothing(Y)$ τον οποίο θα βρείτε έτοιμο στο αρχείο nothing.py.

   (d) συσσωρεύει τα κωδικοποιημένα frames Yc στο πίνακα $Ytot$ διάστασης $\ldots \times M$.

   (e) για κάθε frame αντιστρέφει τη διαδικασία με τον αλγόριθμο $Yh = idonothing(Yc)$ τον οποίο θα βρείτε έτοιμο στο αρχείο nothing.py

   (f) παράγει σταδιακά (MN σε κάθε επανάληψη) τα δείγματα ανασυντεθημένου σήματος, $xhat$, από τις μπάντες των επεξεργασμένων frames Yh με χρήση φίλτρων σύνθεσης που αντιστοιχούν στην πρότυπη κρουστική απόκριση h.

   Επισημάνσεις: (1) Η συνάρτηση $codec0$ θα αποτελέσει το κέλυφος του κωδικοποιητή MP3 που θα κατασκευάσετε στη συνέχεια. (2) Όλη η τέχνη στην κατασκευή του $codec0$ βρίσκεται στην κατασκευή και ορθή ενημέρωση ενός buffer δειγμάτων εισόδου διάστασης $(N-1)M + L$ και ενός buffer για τα δείγματα subband διάστασης $((N-1) + L/M) \times M$ όπου L το μήκος της κρουστικής απόκρισης των φίλτρων.

5. Να υλοποιήσετε τη συνάρτηση $Ytot = coder0(wavin, h, M, N)$ που υλοποιεί μόνο τα 4 πρώτα βήματα της $codec0()$.

6. Να υλοποιήσετε τη συνάρτηση $xhat = decoder0(Ytot, h, M, N)$ που υλοποιεί μόνο τα 2 τελευταία βήματα της $codec0()$.

7. Να κωδικοποιήσετε και αποκωδικοποιήσετε το σήμα του αρχείου MYFILE.wav και

   (a) να συγκρίνετε ακουστικά το αρχικό και το αποκωδικοποιημένο σήμα,

   (b) να υπολογίσετε το SNR του σφάλματος $x - xhat$. Προσοχή, λάβετε υπόψη την πιθανή καθυστέρηση που εισάγει η χρήση των buffers.

3.2 DCT (0 μονάδες)

Να κατασκευάσετε τη συνάρτηση $c = frameDCT(Y)$ που μετασχηματίζει κατά DCT ένα frame διαστάσεων $N \times M$ και τοποθετεί τους παραγόμενους συντελεστές στο διάνυσμα c διάστασης $NM \times 1$ σύμφωνα με τη Σχέση (8).

Να κατασκευάσετε τη συνάρτηση $Yh = iframeDCT(c)$ που αντιστρέφει την προηγούμενη.

3.3 Υπολογισμός του κατωφλίου ακουστότητας (1 μονάδα)

1. Να κατασκευάσετε συνάρτηση $P = DCTpower(c)$ που δέχεται σαν είσοδο ένα διάνυσμα συντελεστών DCT και παράγει στην έξοδο ένα ίδιων διαστάσεων διάνυσμα που περιέχει την ισχύ σε dB του κάθε συντελεστή σύμφωνα με τη Σχέση (10).

2. Να κατασκευάσετε τη συνάρτηση $D = Dksparse(Kmax)$ που σε εφαρμογή της Σχέσης (12) για κάθε υποψήφιο masker k υπολογίζει τη γειτονιά Δ_k. Η είσοδος $Kmax$ αντιστοιχεί στη μέγιστη διακριτη συχνότητα (εδώ $Kmax = MN - 1$) και η έξοδος D είναι αραιός (sparse) πίνακας διαστάσεων $Kmax \times Kmax$ με στοιχεία

$$D(k,j) = \begin{cases} 1 & \text{αν η συχνότητα } j \in \Delta_k \\ 0 & \text{αλλιώς} \end{cases} \tag{24}$$

3. Να κατασκευάσετε τη συνάρτηση $ST = STinit(c, D)$ που δέχεται σαν είσοδο τους συντελεστές DCT ενός frame (όπως αυτοί διατάσονται από την $frameDCT()$) και παράγει ένα διάνυσμα με τις θέσεις των tonal components σύμφωνα με τη Σχέση (11). Η δεύτερη μεταβλητή εισόδου, D, πρέπει να είναι ο αραιός πίνακας που παράγεται από την $Dksparse()$.

4. Να κατασκευάσετε τη συνάρτηση $PM = MaskPower(c, ST)$ που δέχεται σαν είσοδο (1) τους συντελεστές DCT ενός frame, όπως και η προηγούμενη, (2) τις θέσεις, ST, των tonal components και παράγει ένα διάνυσμα με την ισχύ των αντίστοιχων maskers σύμφωνα με τη Σχέση (13). από την $Dksparse()$.

5. Να κατασκευάσετε τη συνάρτηση $z = Hz2Barks(f)$ που δέχεται σαν είσοδο ένα διάνυσμα αναλογικών συχνοτήτων εκφρασμένο σε Hz και παράγει ένα ισομήκες διάνυσμα συχνοτήτων σε barks.

6. Να κατασκευάσετε τη συνάρτηση $STr, PMr = STreduction(ST, c, Tq)$ που δέχεται σαν είσοδο (α) το διάνυσμα ST με τις θέσεις κάποιων maskers (τυπικά αυτών που έχουν υπολογιστεί από την $STinit()$, (β) τους συντελεστές DCT ενός frame όπως προκύπτουν από $frameDCT()$, (γ) το διάνυσμα Tq με το όριο ακουστότητας στη σιωπή για κάθε ένα από τους MN συντελεστές DCT και παράγει στην έξοδο (a) ένα διάνυσμα με τις θέσεις των κυρίαρχων maskers και (b) ένα ισόμηκες διάνυσμα με την αντίστοιχη ίσχύ όπως περιγράφεται παραπάνω.

7. Να κατασκευαστεί η συνάρτηση $Sf = SpreadFunc(ST, PM, Kmax)$ που δέχεται σαν είσοδο (α) ένα διάνυσμα ST με τις θέσεις κάποιων maskers (τυπικά την έξοδο της $STreduction()$), (β) το διάνυσμα PM με την αντίστοιχη ισχύ τους, (γ) τη μέγιστη διακριτήσυχνότητα των συντελεστών

DCT ενός frame (τυπικά $Kmax = MN - 1$) και δίνει στην έξοδο τον πίνακα Sf διάστασης $(max + 1) \times length(ST)$ έτσι ώστε η j στήλη του να περιέχει τις τιμές του spreading function για το σύνολο των διακριτών συχνοτήτων $i = 0, \ldots, Kmax$.

8. Να κατασκευαστεί η συνάρτηση $Ti = Masking\_Thresholds(ST, PM, Kmax)$ που δέχεται τις ίδιες εισόδους με την $SpreadFunc()$ και δίνει στην έξοδο τον πίνακα $_i$ διάστασης $(max + 1) \times length(ST)$ έτσι ώστε η j στήλη του να περιέχει τις τιμές του κατωφλίου ακουστότητας για το σύνολο των διακριτών συχνοτήτων $i = 0, \ldots, Kmax$ εξαιτίας του masker υπ' αριθμόν j, δηλαδή του masker με διακριτή συχνότητα $S_T(j)$ και ισχύ $PM(j)$.

9. Να κατασκευαστεί η συνάρτηση $Tg = Global\_Masking\_Thresholds(Ti, Tq)$ που παράγει την τιμή του συνολικού κατωφλίου με βάση τα στοιχεία του πίνακα Ti (που παράγεται από την $Masking\_Thresholds()$) και του διανύσματος T_q με τις τιμές του κατωφλίου στη σιωπή.

10. Να κατασκευάσετε τη συνάρτηση $Tg = psycho(c, D)$ που παράγει τα συνολικά κατώφλια με χρήση των προηγούμενων συναρτήσεων. Οι είσοδοι είναι (α) το διάνυσμα c των συντελεστών DCT ενός frame και (β) ο αραιός πίνακας D που παράγεται από την $Dksparse()$. Οι τιμές του κατωφλίου στη σιωπή περιέχονται στο αρχείο Tq.py.**Προσοχή: Επειδή η διαδικασία που περιγράφεται αποτελεί σοβαρή απλούστευση του προτύπου είναι πιθανό τα απόλυτα αριθμητικά μεγέθη που δίνονται για το κατώφλι $T_q(i)$ να μην είναι σωστά. Συνεπώς αν χρειαστεί μετατοπίστε προς τα πάνω ή προς τα κάτω τη σχετική καμπύλη σε dB για να έχετε καλύτερα αποτελέσματα. Περιγράψτε στην αναφορά σας τις σχετικές τροποποιήσεις.**

3.4 Κβαντισμός/αποκβαντισμός (1 μονάδα)

1. Να κατασκευάσετε τη συνάρτηση $cb = critical\_bands(K)$ που δέχεται σαν είσοδο τις διαστάσεις ενός frame και παράγει ένα διάνυσμα cb, μήκους K, με τιμές $cb(k0) = band$ όταν ο συντελεστής DCT υπ' αριθμόν $k = k0 - 1$ ανήκει στην κρίσιμη μπάντα $band$

2. Να κατασκευάσετε τη συνάρτηση $cs, sc = DCT\_band\_scale(c)$ που δέχεται σαν είσοδο τους συντελεστές DCT ενός frame και παράγει τους κανονικοποιημένους συντελεστές $\tilde{c(i)}$ και τα scale factors, ένα για κάθε critical band.

3. Να κατασκευάσετε τη συνάρτηση $symb\_index = quantizer(x, b)$ που δέχεται σαν είσοδο ένα διάνυσμα x με πραγματικές τιμές στο διάστημα $[-1,\ 1]$ και τις κβαντίζει ομοιόρφα σε $2^b - 1$ ζώνες παραγοντας στην έξοδο το διάνυσμα συμβόλων $symb\_index$. Χρησιμοποιείστε τη σύμβαση ότι ο κβαντιστής παράγει ως σύμβολο τον ακέραιο αριθμό που αντιστοιχεί στον αύξοντα αριθμό της ζώνης που αντιστοιχεί το x. Π.χ. 0 για τη ζώνη $[-w_b,\ +w_b]$, -1 για τη ζώνη $[-2w_b,\ -w_b]$, $+1$ για τη ζώνη $[+w_b,\ +2w_b]$, κ.λπ.

4. Να κατασκευάσετε τη συνάρτηση $xh = dequantizer(symb\_index, b)$ που αντιστρέφει την προηγούμενη.

5. Χρησιμοποιώντας τον επαναληπτικό αλγόριθμο της Ενότητας (2.4) κατασκευάστε τη συνάρτηση $symb\_index, SF, B = all\_bands\_quantizer(c, Tg)$ που δέχεται σαν είσοδο τους συντελεστές DCT ενός frame και το συνολικό κατώφλι Tg που τους αντιστοιχεί και παράγει (α) ένα διάνυσμα με τα αντίστοιχα σύμβολα κβαντισμού (ακέραιους όπως αυτούς της $quantizer()$), (β) τους scale factors SF (που παράγονται από την $DCT\_band\_scale()$) και (γ) τον αριθμό των $bits$ που χρησιμοποίησε ο κβαντιστής σε κάθε critical band.

6. Να κατασκευάσετε τη συνάρτηση $xh = all\_bands\_dequantizer(symb\_index, B, SF)$ που αντιστρέφει την προηγούμενη.

3.5 Κωδικοποίηση μήκους διαδρομής (0.5 μονάδα)

Να κατασκευαστεί η συνάρτηση υπολογισμού των συμβόλων μήκους διαδρομής για τα διανύσματα των κβαντισμένων συντελεστών DCT. Τα σύμβολα μήκους διαδρομής θα είναι διάδες της μορφής $(quant\_symbol, \ followin$ Τα μήκη διαδρομής θα υπολογίζονται χωριστά για κάθε frame. Η συνάρτηση θα είναι της μορφής:

$$run\_symbols = RLE\,(symb\_index, K) \tag{25}$$

με εισόδους (α) το διάνυσμα, $symb\_index$, των σύμβολων κβαντισμού που αντιστοιχούν σε ένα frame όπως παράγοντται από την $all\_bands\_quantizer()$, (β) το πλήθος των συντελεστών DCT του frame (τυπικά $K = MN = 1152$). Ο πίνακας $run\_symbols$ στην έξοδο της συνάρτησης θα είναι διάστασης $R \times 2$ όπου R τα μήκη διαδρομής που εντοπίστηκαν στο συγκεκριμένο frame.

Να κατασκευαστεί η συνάρτηση

$$symb\_index = RLE\,(run\_symbols, K) \tag{26}$$

που αντιστρέφει την προηγούμενη.

3.6 Κωδικοποίηση Huffman (0.5 μονάδα)

1. Να κατασκευαστεί συνάρτηση κωδικοποίησης Huffman για τα μήκη διαδρομής της προηγούμενης ενότητας:

$$frame\_stream, frame\_symbol\_prob = huff\,(run\_symbols) \tag{27}$$

όπου η έξοδος $frame\_stream$ είναι μία συμβολοσειρά από 0 και 1 και ο πίνακας $frame\_symbol\_prob$ περιέχει δύο στήλες με τα σύμβολα μήκους διαδρομής που ανιχνεύτηκαν και μία τρίτη με τις αντίστοιχες εκτιμώμενες πιθανότητες εμφάνισής τους.

2. Να κατασκευαστεί η αντίστροφη συνάρτηση:

$$run\_symbols = \textbf{ihuff}\,(frame\_stream, frame\_symbol\_prob) \tag{28}$$

Να συναρμολογηθούν τα παραπάνω στις συναρτήσεις $MP3codec()$, $MP3cod()$ και $MP3decod()$ τροποποιώντας κατάλληλα τις $codec0()$, $coder0()$ και $decoder0()$ αντίστοιχα.

Επισήμανση: είναι πιθανόν το δημιουργούμενο bitstream που προκύπτει από την αλληλουχία των $frame\_stream$ να είναι υπερβολικά μεγάλο για τη μνήμη του υπολογιστή σας. Σ' αυτή την περίπτωση φροντίστε να γράφετε τα αποτελέσματα κάθε κλήσης της $huff()$ σε ένα αρχείο ascii το οποίο θα διαβάζετε στη συνέχεια κατά την αποκωδικοποίηση.

Τελικά,

1. να υπολογιστεί ο συνολικός βαθμός συμπίεσης.
2. να κωδικοποιήσετε και αποκωδικοποιήσετε το σήμα του αρχείου MYFILE.wav και
   (a) να συγκρίνετε ακουστικά το αρχικό και το αποκωδικοποιημένο σήμα,
   (b) να υπολογίσετε το SNR του σφάλματος $x - xhat$.

Για την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Η αναφορά θα πρέπει να επιδεικνύει την ορθή λειτουργία του κώδικά σας στο αρχείο ήχου που σας δίνεται.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλίσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για *οποιαδήποτε* είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για το σήμα που σας δίνεται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα, καθώς και να τηρούνται τα ακόλουθα:

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι `AEM1-AEM2.zip`, όπου `AEMi` είναι τα τέσσερα ψηφία του Α.Ε.Μ. των μελών της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα python και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα `report.pdf`.

- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε τα αρχεία ήχου που σας δίνονται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ “Thumbs.db”, “.DS_Store”, “.directory”.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “#”, “$”, “%” κλπ.