

# Συστήματα Πολυμέσων

## Εργασία 2022-2023

Κώστας Μαυρογιαννάκης , 9789

February 20, 2023

Καθώς υπάρχουν σχόλια πάνω στον κώδικα για το τι εντολές επέλεξα να χρησιμοποιήσω θα δώσω έμφαση σε αυτή την αναφορά μόνο στην λογική πίσω από τις συναρτήσεις

- Για τις γραφικές παραστάσεις κράτησα και θετικές και αρνητικές συχνότητες.
- Εφόσον ο buffer έχει  $(N - 1) * M + L$  δείγματα σε κάθε επανάληψη, προκαλείται μία καθυστέρηση 480 δειγμάτων στο παραγόμενο  $\hat{x}$ . Οπότε κατά τον υπολογισμό του SNR αρκεί να κάνουμε slide στο διάνυσμα ώστε οι πρώτες 480 τιμές να γίνουν ίσες με τις 480 τελευταίες και οι υπόλοιπες  $514944 - 480$  με τις τιμές από 0 έως  $514944 - 480$ .
- Όταν γράφουμε το παραγόμενο  $\hat{x}$  με την εντολή `write` της `scipy io wavfile`, παρότι το SNR μας δείχνει ότι το σήμα είναι κοντά στο αρχικό, δεν ακούγεται τίποτα (τουλάχιστον στο VLC που το άνοιγα). Αυτό διάβασα ότι γίνεται διότι δεν δίνεται το αναμενόμενο `type` και με μία διαίρεση με το μέγιστο το αρχείο γίνεται ακουστό.
- Το παραγόμενο `.wav` αρχείο ακούγεται λίγο πιο δυνατά από το αρχικό. Εξήγηση για αυτό δεν βρήκα. Αν πριν το γράφουμε με την εντολή `write` πολλαπλασιάσουμε με περίπου 0.6 τότε ακουστικά έχουμε το ίδιο με το αρχικό.
- Στην συνάρτηση `STredution` κατά το δεύτερο κομμάτι αφαίρεσης των `maskers` ελέγχω διαδοχικά τους γειτονικούς `maskers` και διαγράφω στο τέλος του `loop`. Ίσως θα ήταν πιο σωστό, η διαδικασία αυτή να επαναλαμβανόταν μέχρι να μην αφαιρεθεί κανένας `masker` σε μία επανάληψη, αλλά δεν δίνεται τέτοια υπόδειξη στην εκφώνηση.
- Όσον αφορά τα `nan` στοιχεία του  $T_q$ , κατά την κλήση της `psycho()` κράτησα τον δωσμένο πίνακα  $T_q$  οπότε και ο  $T_g$  έχει στις ίδιες θέσεις `nan`. Στον έλεγχο για τα `bit` του κβαντιστή βρήκα τις θέσεις που βρίσκονται τα `nan` στοιχεία και σύγκρινα τους πίνακες  $P_b, T_g$  στις θέσεις που δεν βρίσκονται `nan`.
- Κατά τον υπολογισμό των `run lengths`, θεώρησα και την περίπτωση όπου τα πρώτα σύμβολα είναι 0. Δηλαδή για είσοδο 00012102.. τα `run lengths` είναι

$$\begin{bmatrix} 0 & 2 \\ 1 & 0 \\ 2 & 0 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}$$

και όχι

$$\begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}$$

ώστε να μπορούμε να ανακατασκευάσουμε την αρχική ακολουθία συμβόλων χωρίς να χάνουμε πληροφορία.

- Για την υλοποίηση του Huffman δημιουργήθηκε μία κλάση για τα Nodes, μία συνάρτηση που κατασκευάζει το look up table για αντιστοίχιση συμβόλου σε ακολουθία από 0 και 1, ένα αντίστροφο look up table για αντιστοίχιση ακολουθίας από 0 και 1 σε σύμβολο και μία συνάρτηση όπου δίνοντας τα ζευγάρια συμβόλου πιθανότητας παράγει το δέντρο και επιστρέφει την ρίζα του.
- Η υλοποίηση μου του Huffman επιστρέφει το frame stream ως έναν πίνακα 1152x1 με objects όπου το κάθε object είναι ένα string που περιέχει 0 και 1. Πχ '11110101' ..... Έτσι για να μετρήσω το μέγεθος σε bytes υπέθεσα ότι ο κάθε χαρακτήρας είναι 1 bit και βρήκα το άθροισμα του κάθε στοιχείου για κάθε frame (1152x447). Το άθροισμα αυτό είναι 4226144 bits ή 528268 bytes , δηλαδή 500kb , περίπου το μισό από το αρχικό myfile.wav που είναι 1mb.
- Όσον αφορά την MP3coder , αφού βρήκα την ακολουθία Huffman ανέστρεψα τα βήματα με τις αντίστροφες συναρτήσεις και έδωσα το ανεστραμένο  $Y_c$  στον πίνακα  $Y_{tot}$ .

Τρέχοντας το πρόγραμμα εμφανίζεται το plot, γράφεται το αρχείο xhat.wav και εμφανίζεται το SNR στην κονσόλα. Ύστερα ξεκινάει η διαδικασία για την εύρεση της συμβολοσειράς του Huffman και η αναστροφή στο αρχικό  $Y_c$ . Όταν τελειώσει (θέλει πάνω από λεπτό περίπου) εμφανίζεται στην κονσόλα το SNR μετά την αναστροφή και το μέγεθος της συμβολοσειράς ολόκληρου του αρχείου σε bytes.