

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάθημα Προπτυχιακών Σπουδών:

Συστήματα Πολυμέσων

Ακαδημαϊκό Έτος: 2022 - 2023

Εξάμηνο: 6ο

Εργασία

Ομάδα Εργασίας:

Θοδωρής Κοξάνογλου Π20094

Αποστόλης Σιαμπάνης Π20173

Δημήτρης Στυλιανού Π20004

Περιεχόμενα

Εκφώνηση Εργασίας.....	3
Θέμα 1.....	4
Υποερώτημα 1.....	4
Επεξήγηση κώδικα.....	4
Παράδειγμα Εκτέλεσης Προγράμματος.....	7
Terminal.....	7
Βίντεο.....	8
Υποερώτημα 2.....	10
Επεξήγηση κώδικα.....	10
Παράδειγμα Εκτέλεσης Προγράμματος.....	13
Terminal.....	13
Βίντεο.....	15
Θέμα 2.....	17
Επεξήγηση κώδικα.....	17
Παράδειγμα Εκτέλεσης Προγράμματος.....	18
Terminal.....	18
Βίντεο.....	19
Βιβλιογραφία - Δικτυογραφία.....	21

Εκφώνηση Εργασίας

- Ημερομηνία παράδοσης: Ημερομηνία εξέτασης μαθήματος, ώρα 23:59.
- Η παράδοση γίνεται μέσω της πλατφόρμας e-class.
- Η εργασία εκπονείται σε ομάδες **1-4** φοιτητών. Δεν χρειάζεται να δηλώσετε ομάδες.

Αποδεκτές γλώσσες προγραμματισμού είναι οι **Python, Matlab**. Παραδίδονται:

- a) η τεκμηρίωση της εργασίας σε ένα αρχείο pdf, στην πρώτη σελίδα της οποίας αναγράφονται τα ονοματεπώνυμα των φοιτητών και οι ΑΜ. Δεν θα βαθμολογηθούν εργασίες που δεν περιέχουν τεκμηρίωση ή που δεν αναφέρουν τα ονόματα των μελών της ομάδας στην τεκμηρίωση.
- b) τα αρχεία source code σε ένα συμπιεσμένο αρχείο με όνομα source2023.zip.
- c) οποιαδήποτε άλλα συνοδευτικά αρχεία η ομάδα κρίνει απαραίτητα σε ένα συμπιεσμένο αρχείο με το όνομα auxiliary2023.zip.

- Η εργασία αποτελείται από δύο θέματα.

Θέμα 1 (1.5 βαθμοί): Έστω ασυμπίεστο video της επιλογής σας, διάρκειας 5 s – 15 s. Υποθέστε ότι το Frame 1 είναι πάντα I frame και ότι τα επόμενα πλαίσια είναι P frames.

- 1) Κάθε πλαίσιο P προβλέπεται **χωρίς αντιστάθμιση κίνησης** από το προηγούμενο πλαίσιο. Υπολογίστε και απεικονίστε την ακολουθία εικόνων σφάλματος και κωδικοποιήστε την χωρίς απώλειες. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.
- 2) Υλοποιήστε **την τεχνική αντιστάθμισης κίνησης** για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε **macroblocks 64x64, ακτίνα αναζήτησης k=32** και τεχνική σύγκρισης macroblocks της επιλογής σας. Να επιταχυνθεί η διαδικασία υλοποιώντας ιεραρχική αναζήτηση. Υπολογίστε, αποθηκεύστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων σφαλμάτων. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

Θέμα 2 (1.5 βαθμοί): Σε βίντεο της επιλογής σας, διάρκειας 5s – 10s, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το αλγοριθμικά, αξιοποιώντας την τεχνική αντιστάθμισης κίνησης. Δηλαδή, δημιουργήστε και αποθηκεύστε ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που επιλέξατε. Υλοποιήστε και τεκμηριώστε το σχετικό σύστημα.

- Η αντιγραφή οδηγεί σε μηδενισμό όλων των εμπλεκόμενων εργασιών. Ομοίως, μηδενίζονται οι εργασίες που χρησιμοποιούν bots.

Καλή επιτυχία!

Θέμα 1

Έστω ασυμπίεστο video της επιλογής σας, διάρκειας **5s – 15s**. Υποθέστε ότι το Frame 1 είναι πάντα **I frame** και ότι τα επόμενα πλαίσια είναι **P frames**.

Υποερώτημα 1

Κάθε πλαίσιο P προβλέπεται **χωρίς αντιστάθμιση κίνησης** από το προηγούμενο πλαίσιο. Υπολογίστε και απεικονίστε την ακολουθία εικόνων σφάλματος και κωδικοποιήστε την χωρίς απώλειες. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

Επεξήγηση κώδικα

Στο αρχείο **thema_1.1.py** έχουμε δημιουργήσει δύο μεθόδους, μία μέθοδος για τον κωδικοποιητή και μία μέθοδος για τον αποκωδικοποιητή. Ακολουθούν τα βήματα της κάθε μιας μεθόδου:

Κωδικοποιητής

Ανάγνωση βίντεο

Ο κωδικοποιητής διαβάζει το αρχικό ασυμπίεστο βίντεο - `thema_1.avi`. Αφού φορτώσει το βίντεο στη μνήμη, αποθηκεύει κάθε πλαίσιο του βίντεο σε μία λίστα - **frames**, και τις ιδιότητες το βίντεο (πλάτος, ύψος, ρυθμός μετάδοσης πλαισίων) σε μία λίστα - **videoProperties**. Για την καλύτερη πρόβλεψη των πλαισίων του βίντεο από τον αποκωδικοποιητή, αποθηκεύει εκ νέου τα πλαίσια του βίντεο σε κλίμακα του γκρι.

Προεραϊκά αποθηκεύουμε το αρχικό βίντεο σε κλίμακα του γκρι στο φάκελο **auxiliary2023/OutputVideos** του project.

Στη συνέχεια ο κωδικοποιητής υπολογίζει την εντροπία όλων των πλαισίων του αρχικού βίντεο σε κλίμακα του γκρι και την αποθηκεύει στη μεταβλητή **H**.

Υπολογισμός ακολουθίας εικόνων σφάλματος

Το επόμενο βήμα που υλοποιεί ο κωδικοποιητής είναι ο υπολογισμός της ακολουθίας εικόνων σφάλματος, μεταξύ του τρέχοντος πλαισίου - **target frame** με το προηγούμενο πλαίσιο - **reference frame** για κάθε πλαίσιο του αρχικού βίντεο.

Εικόνα σφάλματος: (Πλαίσιο(*i*) - Πλαίσιο(*i*-1)), όπου *i* η θέση του τρέχοντος πλαισίου ($2 \leq i \leq n$) και *n* το πλήθος πλαισίων.

Υλοποίηση Κωδικοποίησης Huffman

Θέλουμε να αποθηκεύσουμε τα δεδομένα του βίντεο έχοντας συμπίεση χωρίς απώλειες. Για την επίτευξη του στόχου αυτού ο κωδικοποιητής χρησιμοποιεί την κωδικοποίηση Huffman.

Η διαδικασία ξεκινά υπολογίζοντας στατιστικά την πιθανότητα εμφάνισης κάθε συμβόλου.

Στην αρχή την κωδικοποίησης Huffman πρέπει ο κωδικοποιητής να μπορεί να επεξεργαστεί τα δεδομένα των πλαισίων του βίντεο. Για να γίνει εύκολη διαχείριση αυτή μετατρέπουμε τα πλαίσια σε bitarrays. Για κάθε bitarray μετράμε την συχνότητα της κλίμακας του γκρι για κάθε pixel και δημιουργούμε μία λίστα για όλα τα πλαίσια του βίντεο στην οποία αποθηκεύονται tuples της μορφής <απόχρωση γκρι, συχνότητα εμφάνισης>.

Σύμφωνα με αυτή την λίστα δημιουργούμε το δυαδικό δέντρο του Huffman. Τα φύλλα του δυαδικού δέντρου αντιστοιχίζονται στις πιθανότητες εμφάνισης των συμβόλων.

Σε δομή επανάληψης (μέχρι να υπάρχει μόνο ένας κόμβος στο δυαδικό δέντρο):

Συνδυάζονται κάθε φορά οι δύο μικρότερες πιθανότητες των κόμβων και δημιουργείται ένας νέος κόμβος (πατέρας), ο οποίος έχει πιθανότητα ίση με το άθροισμα των δύο πιθανοτήτων που συνδιάστηκαν (παιδιά). Οι κόμβοι που συγχωνεύτηκαν μπορεί να είναι φύλλα ή κάποιοι από τους νέους κόμβους που δημιουργήθηκαν κατά την έναρξη του αλγορίθμου.

Σε κάθε κλαδί του δέντρου ανατίθεται η ετικέτα 0 αν βρίσκεται αριστερά κάθε κόμβου ή 1 αν βρίσκεται δεξιά κάθε κόμβου, ξεκινώντας από τη ρίζα. Διατρέχονται όλοι οι ενδιαμέσοι κόμβοι.

Μετά το πέρας της επανάληψης υπολογίζουμε τον μοναδικό κωδικό Huffman για κάθε φύλλο του δέντρου που μόλις δημιουργήσαμε. Ο κωδικός για κάθε φύλλο αντιστοιχεί στη διαδρομή που σημειώνουμε ότι διασχίσαμε από την κορυφή του δέντρου μέχρι να φτάσουμε στο φύλλο αυτό. Για την καλύτερη διαχείριση του δυαδικού δέντρου από τον αποκωδικοποιητή, μετατρέψαμε το δυαδικό δέντρο σε λεξικό.

Εξαγωγή δεδομένων για το συμπιεσμένο χωρίς απώλειες βίντεο

Αφού δημιουργηθεί ο πίνακας Huffman με επιτυχία, αποθηκεύουμε τα κωδικοποιημένα πλαίσια του αρχικού ασπρόμαυρου βίντεο (σε bitarray) - τον πίνακα Huffman και τα δεδομένα του βίντεο (πλάτος, ύψος και ρυθμός μετάδοσης πλαισίων) - σε τρία διαφορετικά αρχεία pickle, ώστε να μπορεί να διαχειριστεί τα δεδομένα αυτά ο αποκωδικοποιητής.

Αποκωδικοποιητής

Ανάγνωση δεδομένων κωδικοποιημένου βίντεο

Διαβάζει τα αρχεία pickle που δημιούργησε ο κωδικοποιητής.

Υλοποίηση αποκωδικοποιητή Huffman

Δημιουργεί ένα ανεστραμμένο λεξικό - **reverseTable** - με τις αποχρώσεις του γκρι και τους μοναδικούς κωδικούς Huffman και μία λίστα - **decodedSeqErrorImages** - η οποία περιέχει τις εικόνες σφάλματος.

Σε μία δομή επανάληψης (που τελειώνει όταν έχουν διαβαστεί όλα τα bitarrays - κωδικοποιημένες εικόνες σφάλματος):

Δημιουργεί για κάθε εικόνα σφάλματος μία προσωρινή λίστα - **decodedErrorImage** - στην οποία αποθηκεύει τα αποκωδικοποιημένα pixel του πλαισίου. Στο τέλος κάθε επανάληψης αποθηκεύει την λίστα αυτή στην **decodedSeqErrorImages**.

Μετά το πέρας της τρέχουσας επανάληψης, κάθε πλαίσιο αναδιαμορφώνεται στις διαστάσεις που είχε στο αρχικό βίντεο.

Επαναφορά πλαισίων στη αρχική τους κατάσταση

Για την επαναφορά των πλαισίων στην αρχική τους κατάσταση, προσθέτουμε την κάθε εικόνα σφάλματος στο αντίστοιχο αποκωδικοποιημένο πλαίσιο.

Εξαγωγή αποκωδικοποιημένου βίντεο

Έχοντας τα τελικά αποκωδικοποιημένα πλαίσια, ανακατασκευάζουμε το αρχικό βίντεο με τα μεταδεδομένα (ύψος, πλάτος και ρυθμός μετάδοσης πλαισίων) που αποθήκευσε ο κωδικοποιητής.

Τέλος, ο αποκωδικοποιητής υπολογίζει την εντροπία του αποκωδικοποιημένου βίντεο και την αποθηκεύει στη μεταβλητή **H**.

Παράδειγμα Εκτέλεσης Προγράμματος

Terminal

Κωδικοποίηση εικόνων σφάλματος

```
The video has 282 frames, a height of 480 pixels, a width of 720 pixels and a framerate of 29.97 frames per second.
Original grayscale video exported successfully!
Entropy of the original grayscale video is: 4.8808442821014015
Huffman tree created successfully!
Huffman table created successfully!
Encoding the error frames sequence: |*****-----| 24.47%
```

Αποκωδικοποίηση εικόνων σφάλματος

```
The video has 282 frames, a height of 480 pixels, a width of 720 pixels and a framerate of 29.97 frames per second.
Original grayscale video exported successfully!
Entropy of the original grayscale video is: 4.8808442821014015
Huffman tree created successfully!
Huffman table created successfully!
Encoded the error frames sequence!
Encoded video properties exported successfully!
Encoded video properties imported successfully!
Decoding the error frames sequence: |***-----| 8.51%
```

Επιτυχής εκτέλεση προγράμματος

```
The video has 282 frames, a height of 480 pixels, a width of 720 pixels and a framerate of 29.97 frames per second.
Original grayscale video exported successfully!
Entropy of the original grayscale video is: 4.8808442821014015
Huffman tree created successfully!
Huffman table created successfully!
Encoded the error frames sequence!
Encoded video properties exported successfully!
Encoded video properties imported successfully!
Decoded the error frames sequence!
The video has 282 frames, a height of 480 pixels, a width of 720 pixels and a framerate of 29.97 frames per second.
Decoded grayscale video exported successfully!
Entropy of the decoded grayscale video is: 4.8808442821014015
The decoded video is the same as the original video!
```

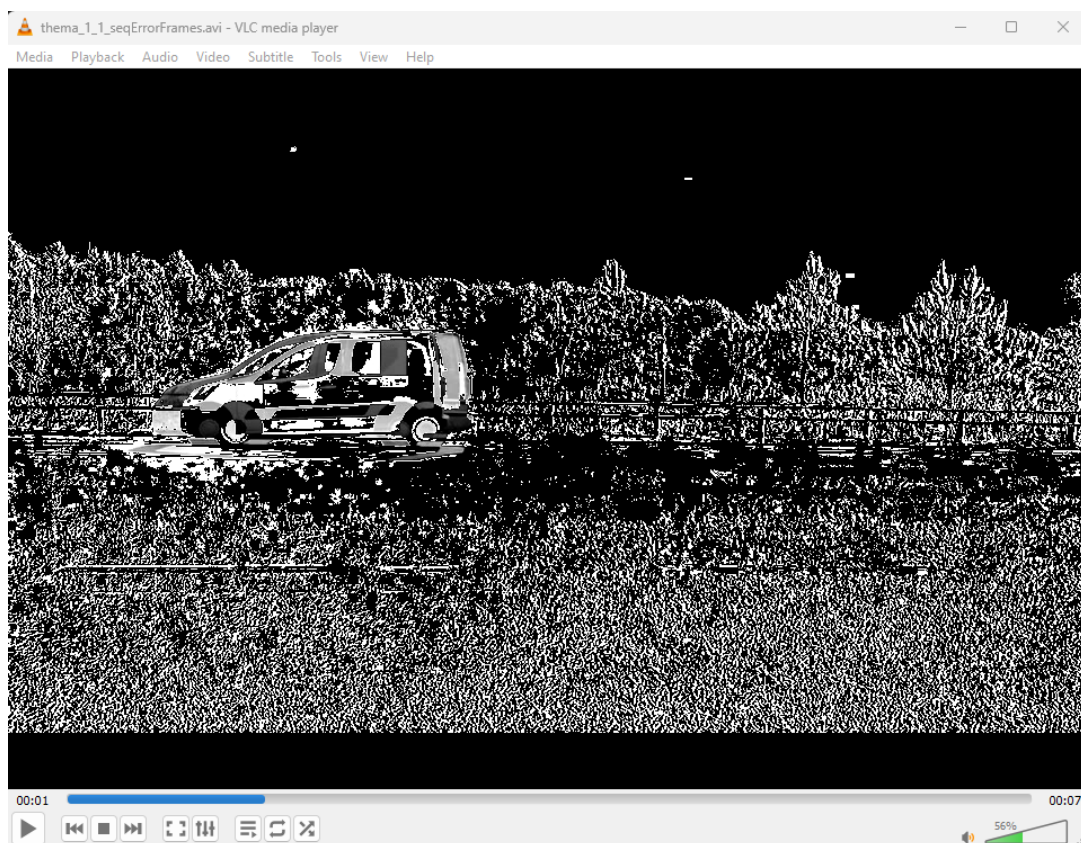
Παρατηρούμε ότι η εντροπία του αρχικού ασπρόμαυρου βίντεο είναι ίδια με την εντροπία του τελικού ασπρόμαυρου βίντεο. Καταλήγουμε στο συμπέρασμα ότι η αποκωδικοποίηση του βίντεο χωρίς απώλειες πραγματοποιήθηκε με επιτυχία.

Βίντεο

Στιγμιότυπο από το αρχικό βίντεο σε κλίμακα του γκρι



Στιγμιότυπο από πλαίσιο της ακολουθίας εικόνων σφάλματος



Στιγμιότυπο από το τελικό βίντεο που εξήγαγε το πρόγραμμα χωρίς απώλειες



Υποερώτημα 2

Υλοποιήστε την **τεχνική αντιστάθμισης κίνησης** για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε **macroblocks 64x64**, **ακτίνα αναζήτησης $k=32$** και τεχνική σύγκρισης macroblocks της επιλογής σας. Να επιταχυνθεί η διαδικασία υλοποιώντας ιεραρχική αναζήτηση. Υπολογίστε, αποθηκεύστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων σφαλμάτων. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

Επεξήγηση κώδικα

Στο αρχείο **thema_1.2.py** έχουμε δημιουργήσει δύο μεθόδους, μία μέθοδος για τον κωδικοποιητή και μία μέθοδος για τον αποκωδικοποιητή. Ακολουθούν τα βήματα της κάθε μιας μεθόδου:

Κωδικοποιητής

Ανάγνωση βίντεο

Ο κωδικοποιητής διαβάζει το αρχικό ασυμπίεστο βίντεο - **thema_1.avi**. Αφού φορτώσει το βίντεο στη μνήμη, αποθηκεύει κάθε πλαίσιο του βίντεο σε μία λίστα - **frames**, και τις ιδιότητες το βίντεο (πλάτος, ύψος, ρυθμός μετάδοσης πλαισίων) σε μία λίστα - **videoProperties**. Για την καλύτερη πρόβλεψη των πλαισίων του βίντεο από τον αποκωδικοποιητή, αποθηκεύει εκ νέου τα πλαίσια του βίντεο σε κλίμακα του γκρι.

Προεραϊκά αποθηκεύουμε το αρχικό βίντεο σε κλίμακα του γκρι στο φάκελο **auxiliary2023/OutputVideos** του project.

Στη συνέχεια ο κωδικοποιητής υπολογίζει την εντροπία όλων των πλαισίων του αρχικού βίντεο σε κλίμακα του γκρι και την αποθηκεύει στη μεταβλητή **H**.

Υπολογισμός Διανυσμάτων Κίνησης

Με την μέθοδο της **ιεραρχικής αναζήτησης** υπολογίζουμε το διάνυσμα κίνησης για όλα τα πλαίσια.

Ιεραρχική αναζήτηση

Σε κάθε πλαίσιο εφαρμόζονται δύο επίπεδα υποδειγματοληψίας ανά 2, με αυτό τρόπο επιτυγχάνουμε την δημιουργία τριών επιπέδων. Κατά την εκτέλεση του υψηλότερου επιπέδου εκτελείται μια πλήρης αναζήτηση, η οποία ακολουθείτε από διορθωτικές αναζητήσεις στα χαμηλότερα επίπεδα. Έτσι, μπορούμε να έχουμε μεγαλύτερη ακρίβεια κατά την εύρεση των διανυσμάτων κίνησης. Τα διανύσματα κίνησης εξηγούν την μετακίνηση ενός **macroblock** στο **πλαίσιο στόχος - target frame** - από το πλαίσιο αναφοράς, όπου καθίσταται ένα αντικείμενο τμηματικής πρόβλεψης βάση του προηγούμενου πλαισίου, δηλαδή του **πλαισίου αναφοράς - reference frame**. Το διάνυσμα κίνησης που υπολογίζεται, βελτιώνεται σε κάθε επίπεδο βάσει της μετρικής **SAD (Sum of Absolute Differences)**, δηλαδή της απόλυτης διαφοράς μεταξύ της υποψήφιας περιοχής και του τμήματος στόχου.

Παραδοχή: Στη παρούσα υλοποίηση της ιεραρχικής αναζήτησης, τα κατώτερα επίπεδα εκτελούν αναζήτηση ακριβώς στα γειτονικά macroblocks, διότι από την εκφώνηση γνωρίζουμε ότι η ακτίνα αναζήτησης (k) του κατώτατου επιπέδου είναι ίση με 32 εικονοστοιχεία.

Υπολογισμός προβλεπόμενων P πλαισίων

Για τον υπολογισμό των προβλεπόμενων P πλαισίων, ορίζουμε ως **πλαίσιο αναφοράς** το αντίστοιχο πλαίσιο από την προαναφερθείσα λίστα **frames** και ως **πλαίσιο στόχος**, ένα πλαίσιο αρχικοποιημένο με τη τιμή 0 για κάθε εικονοστοιχείο - pixel. Στη συνέχεια, βάσει των διανυσμάτων κίνησης, αντικαθιστούμε τα macroblocks του πλαισίου στόχος με αυτά του πλαισίου αναφοράς, αντίστοιχα.

Υπολογισμός ακολουθίας εικόνων σφάλματος

Όταν δημιουργηθούν τα νέα προβλεπόμενα P πλαίσια, σειρά έχει η δημιουργία της ακολουθίας εικόνων σφάλματος. Η τεχνική για την δημιουργία αυτής της ακολουθίας είναι διαφορετική από αυτή του υποερωτήματος 1. Για να αποκτήσουμε τα πλαίσια σφάλματος, αρκεί ο κωδικοποιητής να αφαιρέσει τα νέα ανακατασκευασμένα P πλαίσια από τα αρχικά P πλαίσια του βίντεο.

Κωδικοποίηση Huffman

Χρησιμοποιούμε την κωδικοποίηση Huffman για τα διανύσματα κίνησης που υπολογίσαμε από την ιεραρχική αναζήτηση καθώς επίσης και για τις εικόνες ακολουθίας σφάλματος. Η υλοποίηση του κωδικοποιητή είναι ίδια με αυτή που περιγράφηκε στο θέμα 1.1.

Εξαγωγή δεδομένων για το συμπιεσμένο με απώλειες βίντεο

Αφού δημιουργηθεί ο πίνακας Huffman με επιτυχία, αποθηκεύουμε τα κωδικοποιημένα πλαίσια του αρχικού ασπρόμαυρου βίντεο (σε bitarray) - τον πίνακα Huffman και τα δεδομένα του βίντεο (πλάτος, ύψος και ρυθμός μετάδοσης πλαισίων) - σε τρία διαφορετικά αρχεία pickle, ώστε να μπορέσει να διαχειριστεί τα δεδομένα αυτά ο αποκωδικοποιητής.

Αποκωδικοποιητής

Ανάγνωση δεδομένων κωδικοποιημένου βίντεο

Διαβάζει τα αρχεία pickle που δημιούργησε ο κωδικοποιητής.

Υλοποίηση αποκωδικοποιητή Huffman

Η υλοποίηση του αποκωδικοποιητή είναι ίδια με αυτή που περιγράφηκε στο θέμα 1.1. Χρησιμοποιούμε τον αποκωδικοποιητή Huffman, για την αποκωδικοποίηση των διανυσμάτων κίνησης και των εικόνων ακολουθίας σφάλματος.

Ανακατασκευή προβλεπόμενων P πλαισίων

Για την ανακατασκευή των προβλεπόμενων P πλαισίων παίρνει μέρος μια δομή επανάληψης (με πλήθος επαναλήψεων όσο και το πλήθος frames), όπου σε κάθε επανάληψη ορίζουμε ως **πλαίσιο αναφοράς** το τελευταίο προβλεπόμενο πλαίσιο και ως **πλαίσιο στόχος**, ένα πλαίσιο αρχικοποιημένο με τη τιμή 0 για κάθε εικονοστοιχείο - pixel. Στη συνέχεια, βάσει των

διανυσμάτων κίνησης, αντικαθιστούμε τα macroblocks του πλαισίου στόχος με αυτά του πλαισίου αναφοράς, αντίστοιχα.

Επαναφορά πλαισίων στη αρχική τους κατάσταση

Για την επαναφορά των πλαισίων στην αρχική τους κατάσταση, προσθέτουμε την κάθε εικόνα σφάλματος στο αντίστοιχο προβλεπόμενο πλαίσιο.

Εξαγωγή αποκωδικοποιημένου βίντεο

Έχοντας τα τελικά αποκωδικοποιημένα πλαίσια, ανακατασκευάζουμε το αρχικό βίντεο με τα μεταδεδομένα (ύψος, πλάτος και ρυθμός μετάδοσης πλαισίων) που αποθήκευσε ο κωδικοποιητής.

Τέλος, ο αποκωδικοποιητής υπολογίζει την εντροπία του αποκωδικοποιημένου βίντεο και την αποθηκεύει στη μεταβλητή **H**.

Παράδειγμα Εκτέλεσης Προγράμματος

Terminal

Υπολογισμός Διανυσμάτων Κίνησης

```
The video has 210 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.  
Original grayscale video exported successfully!  
Entropy of the original grayscale video is: 5.026577939934464  
Calculating the motion vectors: |*****-----| 10.95%
```

Κωδικοποίηση εικόνων σφάλματος

```
The video has 210 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.  
Original grayscale video exported successfully!  
Entropy of the original grayscale video is: 5.026577939934464  
Motion Vectors Calculations Completed!  
Motion Compensation Frames Created!  
Sequence Error Images Created!  
Huffman tree created for the vectors successfully!  
Huffman table created successfully!  
Motion vectors encoded successfully!  
Huffman tree created for the sequence error images successfully!  
Huffman table created for the sequence error images successfully!  
Encoding the error frames sequence: |*****-----| 24.29%
```

Αποκωδικοποίηση εικόνων σφάλματος

```
The video has 210 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.  
Original grayscale video exported successfully!  
Entropy of the original grayscale video is: 5.026577939934464  
Motion Vectors Calculations Completed!  
Motion Compensation Frames Created!  
Sequence Error Images Created!  
Huffman tree created for the vectors successfully!  
Huffman table created successfully!  
Motion vectors encoded successfully!  
Huffman tree created for the sequence error images successfully!  
Huffman table created for the sequence error images successfully!  
Encoded the error frames sequence!  
Sequence error images encoded successfully!  
Encoded motion vectors saved successfully!  
Encoded sequence error images saved successfully!  
Encoded video properties imported successfully!  
Decoding the error frames sequence: |*****-----| 39.52%
```

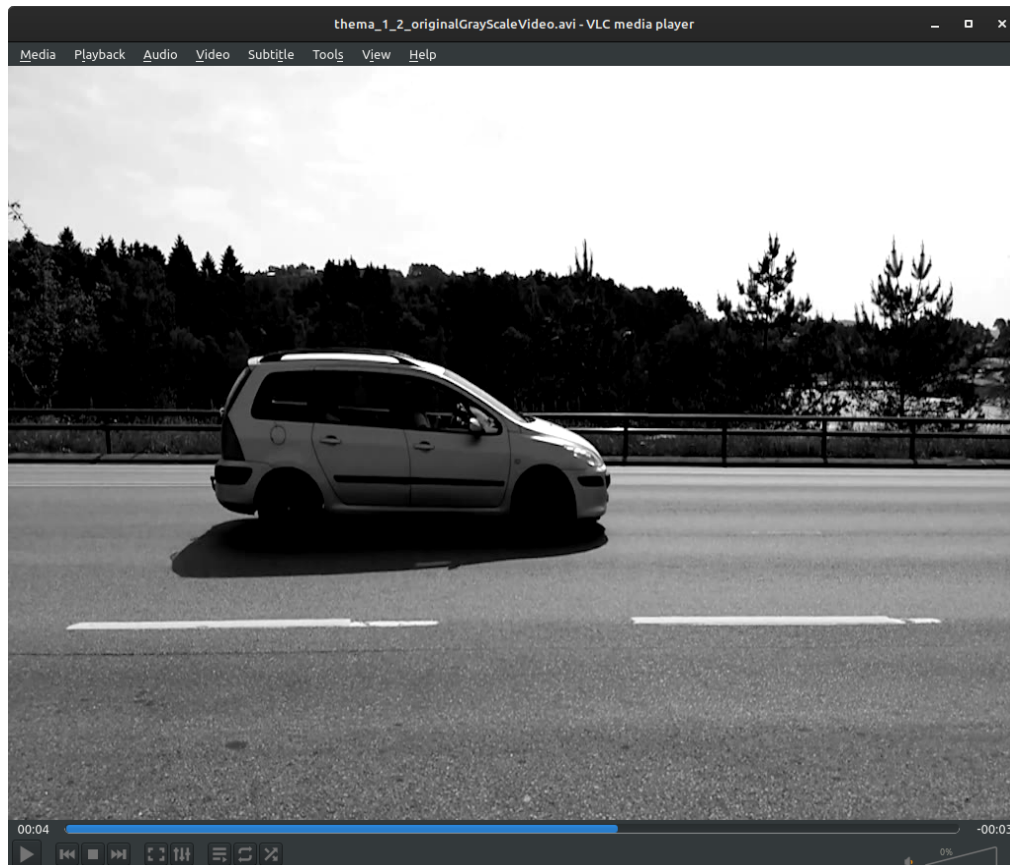
Επιτυχής εκτέλεση προγράμματος

```
The video has 210 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.
Original grayscale video exported successfully!
Entropy of the original grayscale video is: 5.026577939934464
  Motion Vectors Calculations Completed!
  Motion Compensation Frames Created!
  Sequence Error Images Created!
Huffman tree created for the vectors successfully!
Huffman table created successfully!
Motion vectors encoded successfully!
Huffman tree created for the sequence error images successfully!
Huffman table created for the sequence error images successfully!
  Encoded the error frames sequence!
Sequence error images encoded successfully!
Encoded motion vectors saved successfully!
Encoded sequence error images saved successfully!
Encoded video properties imported successfully!
  Decoded the error frames sequence!
Decoded grayscale video exported successfully!
Entropy of the decoded grayscale video is: 5.026577939934464
The decoded video is the same as the original video!

Process finished with exit code 0
```

Παρατηρούμε ότι η εντροπία του αρχικού ασπρόμαυρου βίντεο είναι ίδια με την εντροπία του τελικού ασπρόμαυρου βίντεο. Καταλήγουμε στο συμπέρασμα ότι η αποκωδικοποίηση του βίντεο χωρίς απώλειες με χρήση της τεχνικής αντιστάθμισης κίνησης πραγματοποιήθηκε με επιτυχία.

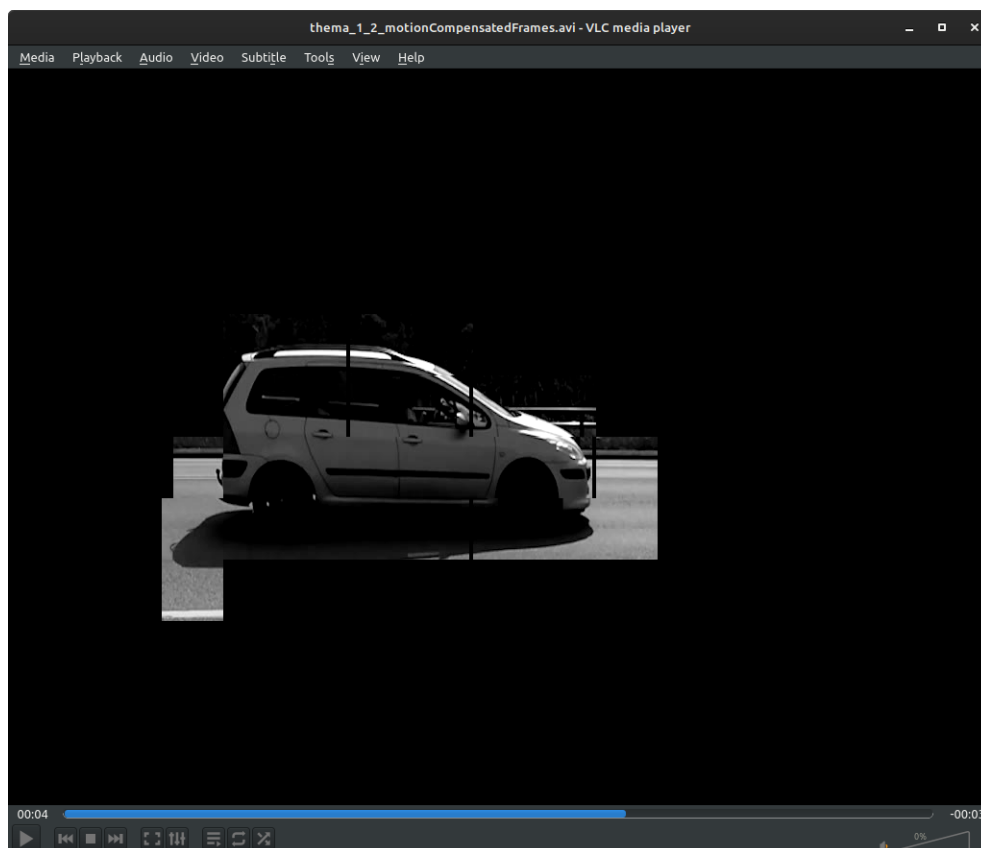
Στιγμιότυπο από το αρχικό βίντεο σε κλίμακα του γκρι



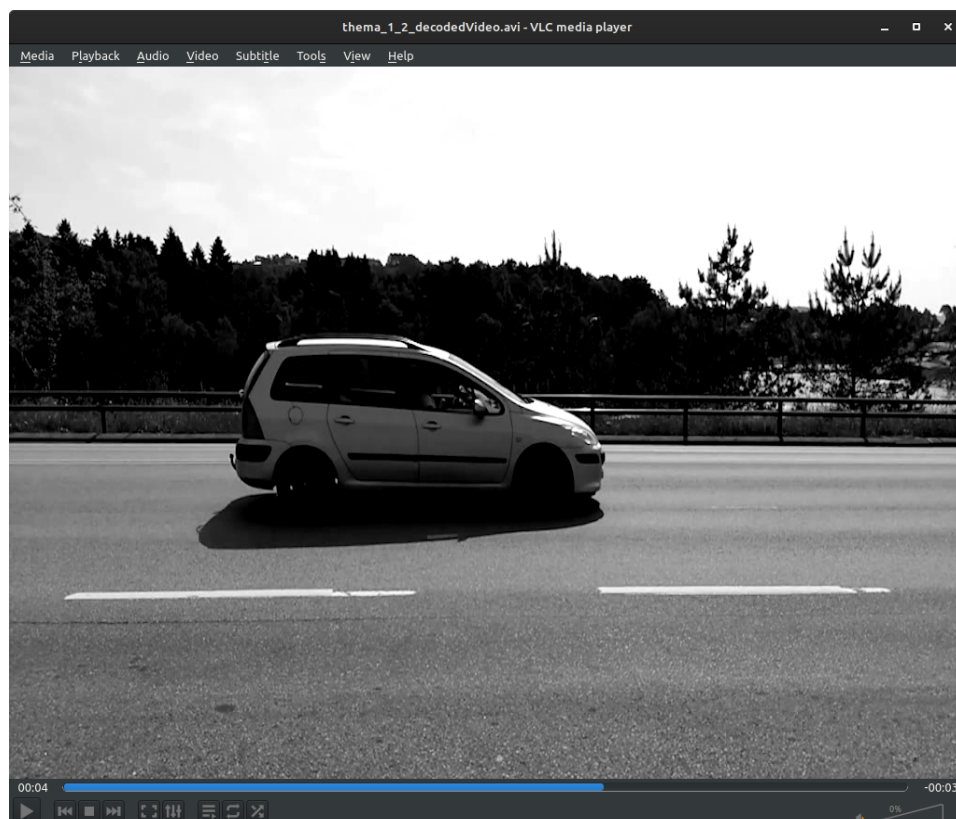
Στιγμιότυπο από πλαίσιο της ακολουθίας εικόνων σφάλματος



Στιγμιότυπο από πλαίσιο της ακολουθίας εικόνων διανυσμάτων κίνησης



Στιγμιότυπο από το τελικό βίντεο που εξήγαγε το πρόγραμμα χωρίς απώλειες



Θέμα 2

Σε βίντεο της επιλογής σας, διάρκειας **5s – 10s**, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το αλγοριθμικά, αξιοποιώντας την τεχνική αντιστάθμισης κίνησης. Δηλαδή, δημιουργήστε και αποθηκεύστε ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που επιλέξατε. Υλοποιήστε και τεκμηριώστε το σχετικό σύστημα.

Επεξήγηση κώδικα

Αρχικά, διαβάζουμε το αρχικό ασυμπίεστο βίντεο - `thema_2.avi`. Αφού φορτώσει το βίντεο στη μνήμη, αποθηκεύει κάθε πλαίσιο του βίντεο σε μία λίστα - **frames**, και τις ιδιότητες το βίντεο (πλάτος, ύψος, ρυθμός μετάδοσης πλαισίων) σε μία λίστα - **videoProperties**. Για την καλύτερη πρόβλεψη των πλαισίων του βίντεο από τον αποκωδικοποιητή, αποθηκεύει εκ νέου τα πλαίσια του βίντεο σε κλίμακα του γκρι.

Προεραϊκά αποθηκεύουμε το αρχικό βίντεο με τα “νέα” πλαίσια σε κλίμακα του γκρι στο φάκελο `auxiliary2023/OutputVideos` του project.

Στη συνέχεια, εκτελείται **ιεραρχική αναζήτηση** (ίδιο με θέμα 1.2) για τον υπολογισμό των διανυσμάτων κίνησης. Έπειτα, για την **εξαφάνιση του αντικειμένου** που βρίσκεται σε κίνηση, υπολογίζουμε τα προβλεπόμενα P πλαίσια. Πιο συγκεκριμένα, κατά τον υπολογισμό κάθε προβλεπόμενου πλαισίου που περιέχει το αντικείμενο (ή μέρος του) προς εξαφάνιση, αντικαθιστούμε τα εικονοστοιχεία του κινούμενου αντικειμένου με τα αντίστοιχα του πρώτου πλαισίου - **I frame** (το οποίο δεν περιέχει το αντικείμενο προς εξαφάνιση).

Εξαγωγή τελικού βίντεο

Έχοντας τα τελικά πλαίσια, ανακατασκευάζουμε το αρχικό βίντεο με τα μεταδεδομένα (ύψος, πλάτος και ρυθμός μετάδοσης πλαισίων).

Παράδειγμα Εκτέλεσης Προγράμματος

Terminal

Υπολογισμός Διανυσμάτων Κίνησης

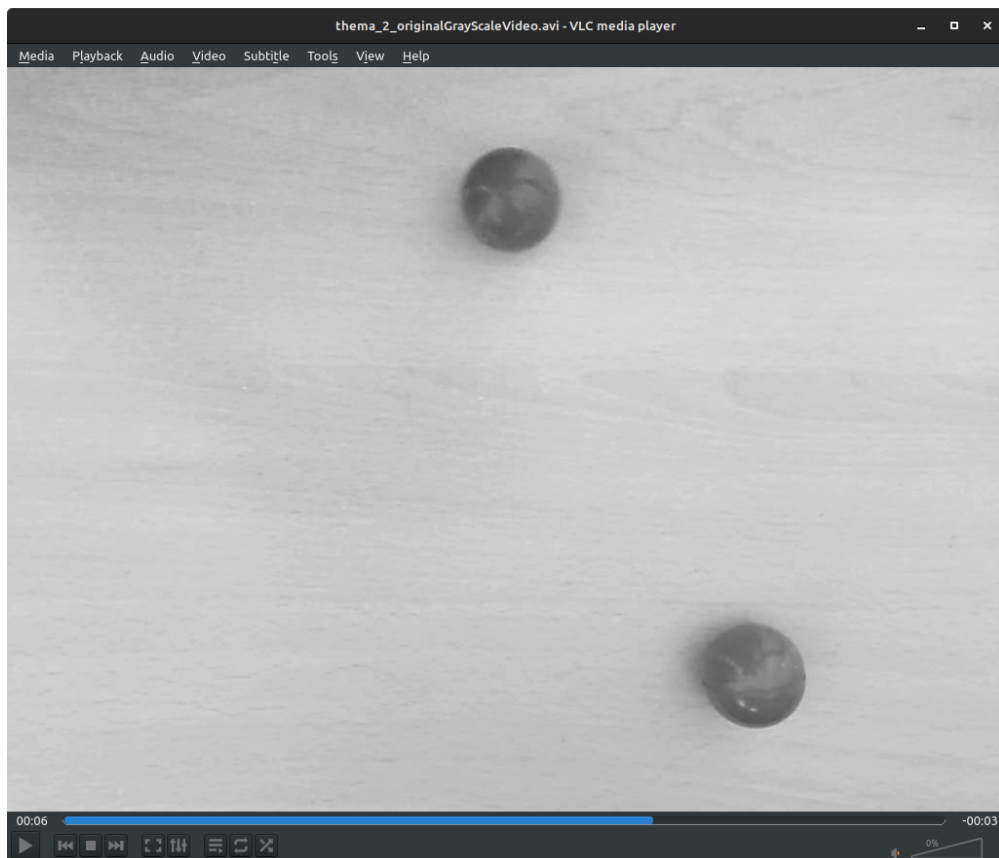
```
The video has 299 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.  
Original grayscale video exported successfully!  
Calculating the motion vectors: |*****-----| 23.41%
```

Επιτυχής εκτέλεση προγράμματος

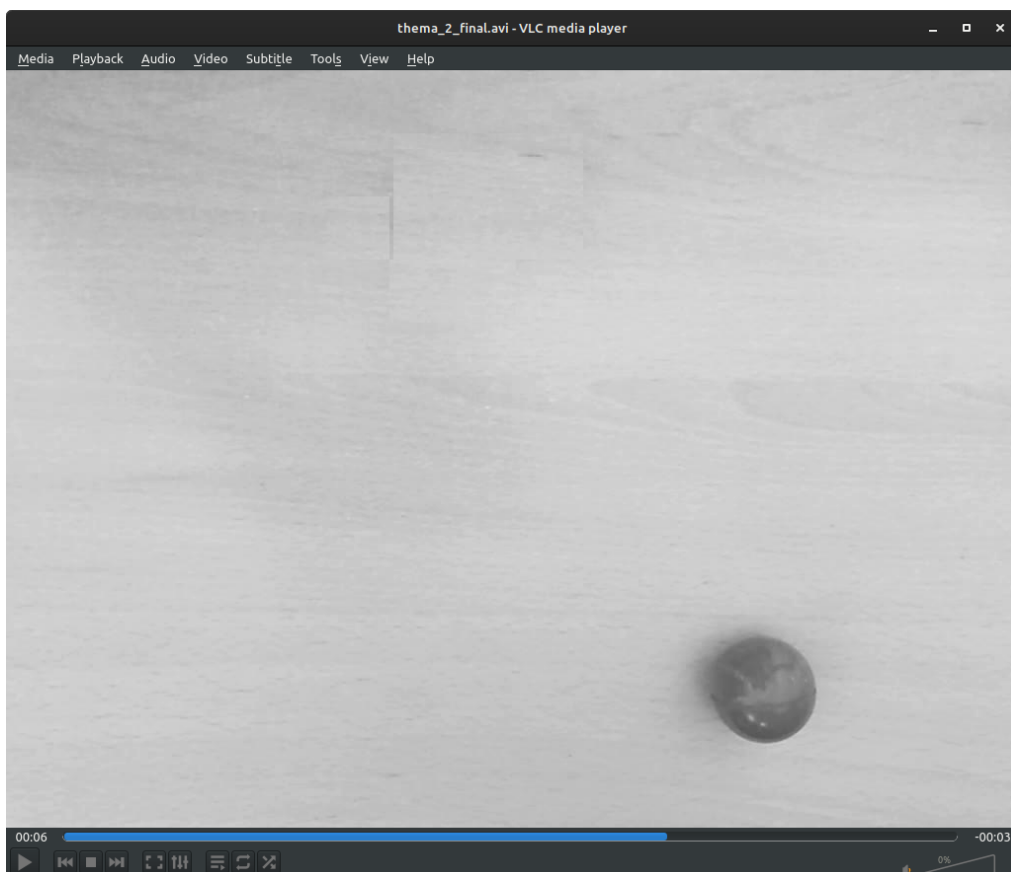
```
The video has 299 frames, a height of 768 pixels, a width of 1024 pixels and a framerate of 30.0 frames per second.  
Original grayscale video exported successfully!  
Motion Vectors Calculations Completed!  
Item disappeared successfully!
```

Βίντεο

Στιγμιότυπο από το αρχικό βίντεο



Στιγμιότυπο από το τελικό βίντεο (ίδιας χρονικής στιγμής)



Σημειώσεις

Τα δύο θέματα υλοποιήθηκαν την γλώσσα προγραμματισμού Python.

Επιπλέον χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες:

- **numpy**: για πράξεις μεταξύ πινάκων
- **colorama**: χρησιμοποιήθηκε για την προσθήκη χρωμάτων στο τερματικό
- **cv2**: για επεξεργασία των βίντεο
- **scipy**: χρησιμοποιήθηκε για τον υπολογισμό της Εντροπίας (H)
- **collections**: για την υλοποίηση του Huffman
- **heapq**: για την υλοποίηση του Huffman

Βιβλιογραφία - Δικτυογραφία

Parag H. & Gerard M. (2012), *Συστήματα Πολυμέσων: Αλγόριθμοι, Πρότυπα & Εφαρμογές*,
Λευκωσία: Broken Hill Publishers LTD

Θέμα 1

Car passing by in a Highway, ανάκτηση από <https://www.youtube.com/watch?v=K6xsEng2PhU>,
στις 30/07/2023

Θέμα 2

Το βίντεο του Θέματος 2 δημιουργήθηκε από εμάς (*thema_2.avi*).