|  |
| --- |
| Photo displaying partial image of two pie charts on a canvas-textured page |
| Συστήματα Πολυμέσων  [Εμβόλιμη εξεταστική - 2024] |
| |  |  |  | | --- | --- | --- | | ΓΕΩΡΓΙΟΣ ΧΡΙΣΤΟΠΟΥΛΟΣ | ΕΞΑΜΗΝΟ 9ο | Π20206 | |

**ΠΕΡΙΕΧΟΜΕΝΑ**

[preparation………………………………………………………………………………………………………………….1](#prepetation)

[ΘΕΜΑ i………………………………………………………………………………………………………………………..1](#i)

[ΘΕΜΑ ii……………………………………………………………………………………………………………………….2](#ii)

[ΘΕΜΑ iii………………………………………………………………………………………………………………………5](#iii)

[ΘΕΜΑ iv………………………………………………………………………………………………………………………6](#iv)

[Για το auxiliary……………………………………………………………………………………………………………](#giatoauxiliary)..7

[**MyGithub**](https://github.com/Georgechrp/MultimediaSystemsProject)

**Preparation:**

* Εγκατάσταση **PyCharm**.
* Θα χρησιμοποιήσω τις βιβλιοθήκες
* **numPy**,
* **opencv-python**(για ανάγνωση/αποθήκευση αρχείων - απαιτείται εγκατάσταση) και
* **zlib:** για συμπίεση χωρίς απώλειες και αποσυμπίεση δεδομένων(για συμπίεση τόσο του I-Frame όσο και του Error Frame) .
* **Os**, για δημιουργεία φακέλων στο σύστημα
* Χρησιμοποιούμε ένα βίντεο με κατάληξη **.avi** με διάρκεια **10 seconds.**

Σημείωση: Στο αρχείο **coder.py** έχει υλοποιηθεί το ερώτημα **(i), (ii)** & **(iv).** Στην τεκμηρίωση του **coder.py** δεν εξηγούνται σημεία που έχουν να κάνουν με το **menu** ή με τον τρόπο **δημιουργίας αρχείων** που προστέθηκαν για καλύτερη χρήση και κατανόηση.

**i)**

* Σε πρώτη φάση, στο αρχείο **coder.py** στον ίδιο φάκελο βάζουμε και το **video** μας.
* ανοίγουμε το βίντεο και αποθηκεύουμε όλα τα καρέ στον πίνακα frames:

A computer screen shot of a program

Description automatically generated

Εάν εκτυπώσουμε την μεταβλητή **frame\_count** θα έχουμε **output**: Σύνολο καρέ: 300 (Η διάρκεια του video είναι 10 seconds 🡪 30 frames per second)

Εάν εκτυπώσουμε τον πίνακα **frames** μπορούμε να δούμε πληροφορίες όπως διαστάσεις και ένταση φωτεινότητας.

* Σε δεύτερη φάση, θα υπολογίσουμε την **εικόνα σφάλματος** για κάθε **P-Frame** ως εξής: εικόνα σφάλματος = P-frame – προηγούμενο frame.Ο τύπος αυτο ισχύει επειδη η εκφώνηση αναφέρει οτι δεν πρέπει να χρησιμοποιηθεί **αντιστάθμιση κίνησης**.
* Να αναφέρουμε εν τάχει πώς τα **I–frames** κωδικοποιούνται ανεξάρτητα, ως πλήρεις εικόνες, ενώ τα **P-frames** κωδικοποιούνται με βάση την πρόβλεψη από προηγούμενα frames. Αποθηκεύονται μόνο οι διαφορές σε σχέση με το προηγούμενο frame.
* Το **GOP** είναι **12**, αφού η εκφώνηση λέει *υποθέστε ότι ανά 12 πλαίσια το πρώτο είναι πάντα τύπου I και τα υπόλοιπα τύπου P.*
* Η **μετατροπή σε byte array** με την χρήση της .**tobytes**() μετατρέπει το καρέ από **numpy array** σε μια **ακολουθία** **byte**. Αυτό είναι απαραίτητο για τη συμπίεση.
* Για να αποφευχθεί **υπερχείλιση τιμών** κατά την **αφαίρεση**, τα δεδομένα των καρέ μετατρέπονται προσωρινά σε ακέραιους (**astype**(**int**)).

A computer screen shot of a black screen

Description automatically generated

* Στο αρχείο **auxiliary** μπορούμε να δούμε όλα **error frames** (από τα **P-frames** εννοείται).
* Σε τρίτη φάση, θα φτιάξουμε και την λειτουργία της **αποκωδικοποίησης**.
* Τα **I-frames** αποθηκεύονται όπως είναι, ενώ τα **P-frames** ανακατασκευάζονται από το προηγούμενο αποκωδικοποιημένο καρέ και την αντίστοιχη **εικόνα σφάλματος**.
* Η κωδικοποιημένη μορφή του καρέ (στη λίστα **encoded\_frames**) αποσυμπιέζεται με τη μέθοδο **zlib.decompress**().Η μέθοδος αυτή επαναφέρει τα συμπιεσμένα byte στην αρχική τους μορφή.
* Τα **αποσυμπιεσμένα** δεδομένα είναι σε μορφή ακολουθίας **byte**, οπότε χρειάζεται να μετατραπούν ξανά σε πίνακα εικόνας (numpy array).Αυτό γίνεται με τη χρήση της np.**frombuffer**(), όπου καθορίζουμε τον τύπο δεδομένων (dtype=np.uint8) και το σχήμα (**reshape(frames[0].shape**)).
* Όπως πριν, τα αποσυμπιεσμένα δεδομένα μετατρέπονται σε πίνακα (**numpy array**) με τη **np.frombuffer**(). Εδώ ο τύπος δεδομένων είναι **np.int16**, καθώς το **Error Frame** περιέχει θετικές και αρνητικές διαφορές.

A computer screen shot of a black screen

Description automatically generated

* Αναλυτικά παραδείγματα εκτέλεσης στο **auxiliary2024.zip**

**ii)** Υλοποιήσαμε την τεχνική εξαντλητικής αντιστάθμισης κίνησης για τη βελτίωση της συμπίεσης βίντεο. Η προσέγγιση αυτή βασίζεται στον διαχωρισμό των καρέ σε macroblocks και τη χρήση διανυσμάτων κίνησης για την πρόβλεψη των P-frames.

Διαδικασία:

* Διαχωρισμός Καρέ σε Macroblocks:
  + Κάθε καρέ διαχωρίστηκε σε macroblocks μεγέθους 16x16 pixels.
  + Ο διαχωρισμός αυτός επιτρέπει την επεξεργασία μικρότερων περιοχών του καρέ.

A computer screen shot of a code

Description automatically generated

* Ορισμός Περιοχής Αναζήτησης:
  + Για κάθε macroblock του τρέχοντος καρέ (P-frame), ορίστηκε μια περιοχή αναζήτησης στο προηγούμενο καρέ (I-frame ή P-frame) με ακτίνα 8 pixels γύρω από τη θέση του macroblock.

A black background with white text

Description automatically generated

* Υπολογισμός Διανυσμάτων Κίνησης (Motion Vectors):
  + Για κάθε macroblock, εξετάσαμε όλα τα υποψήφια macroblocks εντός της περιοχής αναζήτησης.
  + Χρησιμοποιήσαμε ως μέτρο σύγκρισης τη Μέση Απόλυτη Διαφορά (MAD), η οποία υπολογίζεται ως το άθροισμα των απόλυτων διαφορών των pixel μεταξύ των δύο macroblocks.
  + Το macroblock με την ελάχιστη MAD επιλέγεται ως το "καλύτερο ταίριασμα". Αν δεν βρεθεί καλό ταίριασμα, χρησιμοποιείται μηδενικό διάνυσμα κίνησης.

A computer screen shot of a black screen

Description automatically generated

* Υπολογισμός Εικόνας Σφάλματος:
  + Η εικόνα σφάλματος (Error Frame) προκύπτει από τη διαφορά μεταξύ του τρέχοντος macroblock και του αντίστοιχου ταιριάσματος: 
  + Τα Error Frames αποθηκεύονται για κάθε P-frame.
* Αποθήκευση Συμπιεσμένων Δεδομένων:
  + Τα διανύσματα κίνησης και τα Error Frames συμπιέστηκαν χρησιμοποιώντας τη βιβλιοθήκη zlib και αποθηκεύτηκαν σε ξεχωριστό φάκελο (encoded\_frames\_with\_motion).

A computer screen shot of a program code

Description automatically generated

Παρατηρήσεις:

* Τα διανύσματα κίνησης καταγράφουν τη μετατόπιση κάθε macroblock σε σχέση με το προηγούμενο καρέ, διευκολύνοντας την ανακατασκευή των P-frames.
* Η διαδικασία εξαντλητικής αναζήτησης είναι ακριβής αλλά υπολογιστικά εντατική, καθώς εξετάζονται όλες οι πιθανές μετατοπίσεις.
* Τα Error Frames περιέχουν μόνο τις διαφορές, μειώνοντας σημαντικά το μέγεθος των δεδομένων.

Συμπεράσματα:

* Η εξαντλητική αντιστάθμιση κίνησης προσφέρει καλύτερη συμπίεση σε σχέση με την απλή πρόβλεψη χωρίς αντιστάθμιση.
* Τα δεδομένα που προκύπτουν (διανύσματα κίνησης και Error Frames) παρέχουν τα απαραίτητα στοιχεία για την ακριβή αποκωδικοποίηση.

Όπως στο Ερώτημα i, οι παραγόμενες εικόνες σφάλματος και τα διανύσματα κίνησης αποθηκεύονται στο auxiliary2024.zip. Παραδείγματα περιλαμβάνονται για καλύτερη κατανόηση.

**iii)** Δεν έχει υλοποιηθεί το παρόν ερώτημα.

**iv)**

* Για τον υπολογισμό του **βαθμού συμπίεσης** για την περίπτωση **(i) :**
* Υπολογίζεται διαιρώντας το συνολικό μέγεθος των αρχικών δεδομένων (τα καρέ χωρίς συμπίεση) με το συνολικό μέγεθος των συμπιεσμένων δεδομένων.A computer screen shot of a program code

  Description automatically generated
* **Output**:   
  Βαθμός Συμπίεσης:

Μέγεθος αρχικών δεδομένων: 1779.79 MB

Μέγεθος συμπιεσμένων δεδομένων: 296.92 MB

Τελικός βαθμός συμπίεσης: 5.99

* Επομένως τα συμπιεσμένα δεδομένα είναι σχεδόν **6 φορές μικρότερα** από τα αρχικά. Με τη συμπίεση, εξοικονομήθηκε σημαντικός χώρος αποθήκευσης, καθώς το μέγεθος μειώθηκε από **1779.79 MB** σε μόλις **296.92 MB**.

**Για το auxiliary2024.zip[**Η μέθοδος **imwrite()** ανήκει στην κατηγορία αποθήκευσης αρχείων, αποθηκεύει frame σε μορφή αρχείου (π.χ., PNG, JPG)**]:**

* **Ερώτημα i**

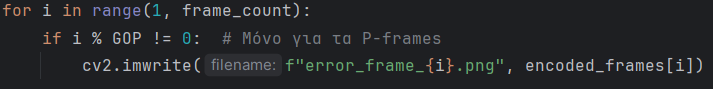
File : “ encoded\_frames” [Τα κωδικοποιημένα frames]

File : “ decoded\_frames” [Τα αποκωδικοποιημένα frames]

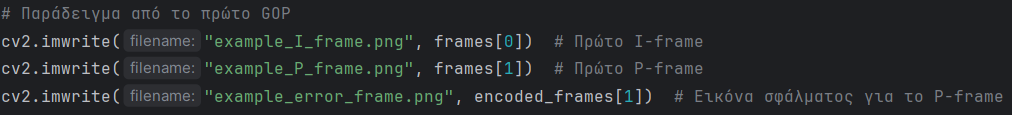
File : “ 300 original frames” [Ολα τα frames(θα χρησιμοποιηθούν στο ii ερώτημα)]



File : “ Error Frames(Only P-frames)” [Όλες οι **εικόνες σφάλματος**]



File : “Example - first GOP”[ Ένα παράδειγμα **I-frame** και το αντίστοιχο **P-frame**, καθώς και την εικόνα σφάλματος του **P-frame**]



* **Ερώτημα ii**

**ii)**