



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ

ΔΙΔΑΣΚΟΝ
ΑΓΓΕΛΟΣ ΠΙΚΡΑΚΗΣ

Π19196 ΚΟΛΟΒΟΣ ΚΥΡΙΑΚΟΣ

ΠΕΡΙΕΧΩΜΕΝΑ	1
ΕΚΦΩΝΗΣΕΙΣ ΕΡΓΑΣΙΑΣ.....	2
Αποτελέσματα εκτέλεσης προγράμματος:.....	4
Υποερώτημα II	5
Αποτελέσματα εκτέλεσης προγράμματος:.....	7
Θέμα 2:	8
Αποτελέσματα εκτέλεσης προγράμματος:.....	10

Εργασία μαθήματος «Συστήματα Πολυμέσων»

Θέμα 1:

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.

- i) 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.
- ii) 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:

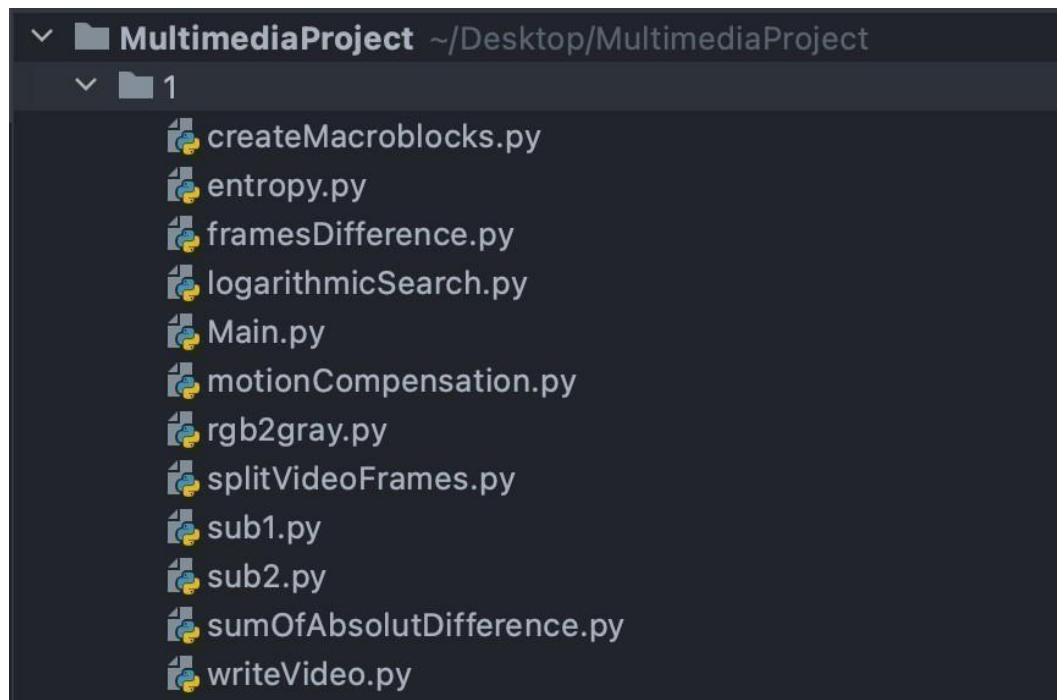
Σε βίντεο της επιλογής σας, διάρκειας 5s – 10s, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το αλγοριθμικά, αξιοποιώντας την τεχνική αντιστάθμισης κίνησης. Δηλαδή, δημιουργήστε και αποθηκεύστε ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που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

Υλοποιήσεις:

Θέμα 1:

Γλώσσα προγραμματισμού που χρησιμοποιήθηκε: **Python**

Για την επίλυση του θέματος δημιουργήθηκαν τα ακόλουθα αρχεία:



Το εκτελεστικό του προγράμματος είναι το αρχείο Main.py όπου καλεί την μέθοδο runSub1(), runSub2() δίνοντας μέσα το path του δείγματος (.mp4). Η runSub1(), runSub2() βρίσκονται μέσα στα sub1.py, sub2.py αντίστοιχα.

```
1 import sub1
2 import sub2
3
4 sample_path = "/home/kali/Desktop/uni/polimesa/auxiliary2022/sample.mp4"
5
6 sub1.runSub1(sample_path)
7
8 sub2.runSub2(sample_path)
9
```

Υποερώτημα Ι.

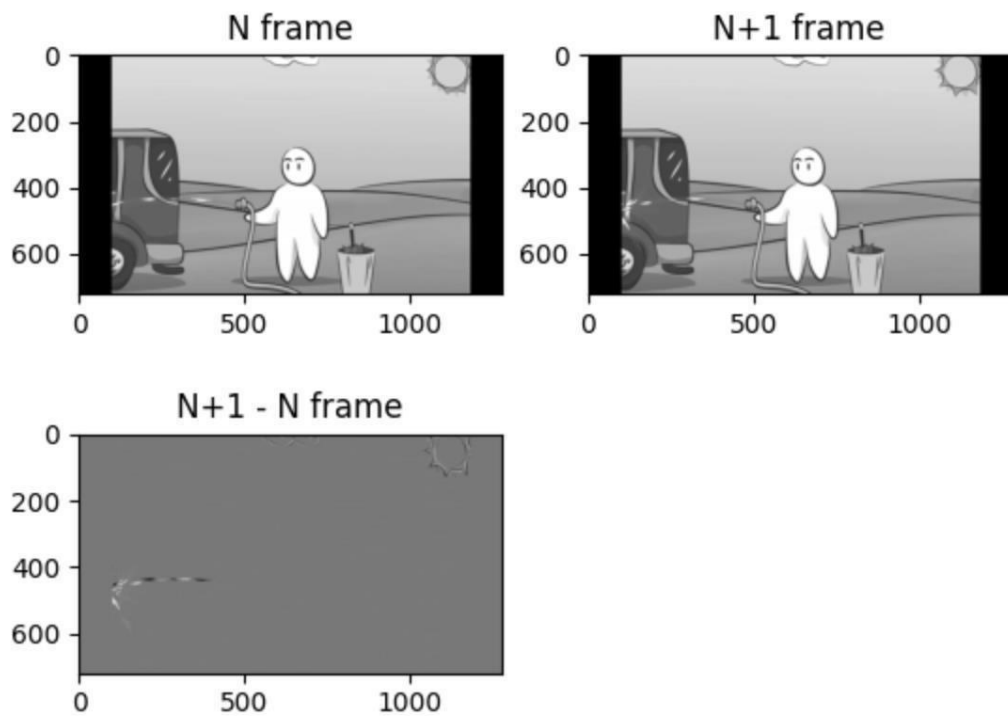
Για την υλοποίηση του ερωτήματος 1 χρησιμοποιήθηκαν τα αρχεία: **splitVideoFrames.py**, **rgb2gray.py**, **entropy.py**, **framesDifference.py**, **writeVideo.py**,

Επίσης, χρησιμοποιήθηκαν οι βιβλιοθήκες: **matplotlib**, **imageio**, **numpy**, **scipy**

- Αρχικά, καλείται η `splitVideoFrames` δίνοντας σαν όρισμα το `path` του βίντεο και επιστρέφει τα πλαίσια εικόνας σε μορφή συστοιχίας. Η `splitVideoFrames` είναι υπεύθυνη δηλαδή για την φόρτωση του δείγματος και την διαχώριση του σε εικονοστοιχεία.
- Στην συνέχεια, καλείται η `rgb2gray` δίνοντας την συστοιχία εικόνων και αυτό έχει ως αποτέλεσμα την μετατροπή των εικόνων σε μαυρόασπρες. Έξοδος από την μέθοδο είναι ένας πίνακας μαυρόασπρων εικόνων.
- Έπειτα καλείται η `framesDif()` που βρίσκεται μέσα στην `framesDifference.py` όπου υπολογίζει και επιστρέφει την διαφορά του $N+1$ frame μαζί με το N . Τα τοποθετεί με την σειρά σε μια λίστα όπου και την επιστρέφει.
- Στην συνέχεια χρησιμοποιούμε την βιβλιοθήκη `matplotlib` ως `plt` για να εμφανίσουμε ένα δείγμα με το αποτέλεσμα από των πιο πάνω διαδικασιών.
- Μετατρέπουμε και γράφουμε την συστοιχία εικόνων με το όνομα `'errorSeqSample.mp4'` και μορφή `.mp4`.
- Τέλος, καλώντας την `averageEntropy()` που βρίσκεται μέσα στην `entropy.py` μετράμε την εντροπία της κάθε εικόνας ξεχωριστά και τα τυπώνουμε στην κονσόλα σαν συνολικό μέσο όρο.

Αποτελέσματα εκτέλεσης προγράμματος:

Αποτελέσματα ryplot:



Αποτελέσματα κονσόλας:

```
Project 1_2
Average Entropy: 5.077328419914948
(kali@kali)-[~/uni/polimesa/sc
```

Αποτέλεσμα εγγραφής βίντεο:

errorSeqSample.mp4

Υποερώτημα II.

Για την υλοποίηση του ερωτήματος 2 χρησιμοποιήθηκαν τα αρχεία: **splitVideoFrames.py**, **rgb2gray.py**, **entropy.py**, **framesDifference.py**, **writeVideo.py**, **motionCompensation.py**, **sumOfAbsolutDifference.py**, **logarithmicSearch.py**, **createMacroblocks.py**.

Επίσης, χρησιμοποιήθηκαν οι βιβλιοθήκες:

matplotlib,
imageio,
numpy, **scipy**,
tqdm.

- Όπως και στο υποερώτημα I διαβάζουμε και χωρίζουμε το βίντεο με την βοήθεια της `splitVideoFrames()` και το μετατρέπουμε σε μαυρόασπρο όπως και πιο πάνω με την χρήση της `rgb2gray()`
 - Στην συνέχεια μέσα σε ένα `for` για κάθε προηγούμενο και επόμενο πλαίσιο μέσα στο σύνολο του συμπιεσμένου κανονικού δείγματος και του δείγματος εξαιρουμένου του πρώτου πλαισίου, καλούνται οι εξής μέθοδοι:
 - `motionCompensation()`
Με την βοήθεια της `createMacroblocks()` και της `logarithmicSearch()` δημιουργεί και επιστρέφει την αντιστάθμιση κίνησης δηλαδή τα `predicted block` και τα `vectors` κίνησης.
 - `createMacroblocks()`
Είναι υπεύθυνη για την δημιουργία των `macroblocks` μεγέθους 32 και για κάθε πλαίσιο εικόνας ξεχωριστά
 - `logarithmicSearch()`
Είναι υπεύθυνη για την λογαριθμική αναζήτηση του `macroblock` στόχου χρησιμοποιώντας ένα `frame` ως `reference`. Η μέθοδος αυτή λειτουργεί με την μέθοδο της αναδρομής. Με την βοήθεια της `createNeighbor()` δημιουργεί τα γειτονικά `macroblocks` του παρόντος.
- Στη συνέχεια καλείται η `calculateSAD()` όπου δέχεται σαν ορίσματα το `reference frame` και τα γειτονικά `macroblocks` και υπολογίζει το άθροισμα της απόλυτης διαφοράς. ο Έπειτα μέσα στο `loop` καλείται η `imageReconstructFromBlocks()` που δέχεται σαν όρισμα τα `predicted`

blocks που έχει επιστρέψει η `motionCompensation()` με σκοπό να κατασκευαστεί εικόνα - frame από τα blocks.

- Κατά την έξοδο από το loop καλείται η `framesDif()` δίνοντας μέσα τον πίνακα με όλα τα frames που προβλέψαμε και υπολογίζεται η διαφορά τους (όπως πιο πάνω $N+1 - N$).
- Χρησιμοποιώντας όπως και στο υποερώτημα I την `pyplot` απεικονίζουμε ένα οποιοδήποτε frame N το frame N+1 και την διαφορά τους.
- Γράφουμε τα δέγματα `compensatedVideoFramesSample` και `compensatedFrameDifferenceSample` χρησιμοποιώντας την `write` με το όνομα `'compensatedSample.mp4'` και `'compensatedErrorSample.mp4'` αντίστοιχα και μορφή `.mp4`.
- Τέλος όπως και στο I μετράμε την εντροπία της κάθε εικόνας ξεχωριστά και τα τυπώνουμε στην κονσόλα.

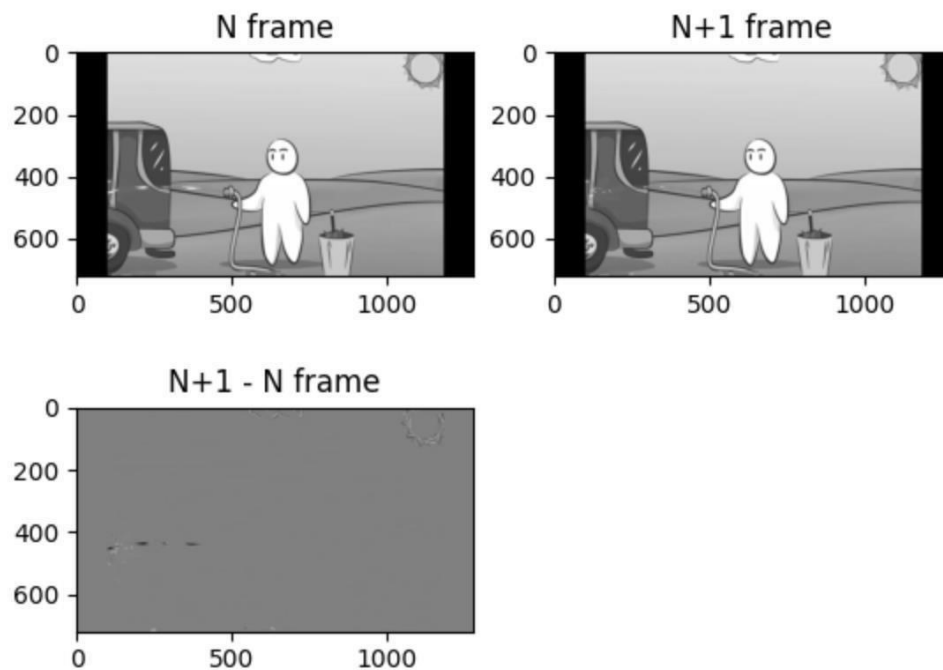
Αναμένοντας το loop να τελειώσει έχει την εξής απόκριση στην κονσόλα όπου προχωράει δυναμικά :

```
7%|█          | 7/99 [00:18<03:56, 2.58s/it]
```

Στην επόμενη σελίδα θα βρείτε αποτελέσματα εκτέλεσης προγράμματος.

Αποτελέσματα εκτέλεσης προγράμματος:

Αποτελέσματα ryrplot:



Αποτελέσματα κονσόλας:

```
Project 1_2  
Average Entropy: 5.999918418855336
```

Αποτέλεσμα εγγραφής βίντεο's:

```
compensatedErrorSample.mp4  
compensatedSample.mp4
```


Θέμα 2:

Για την υλοποίηση του ερωτήματος 2 χρησιμοποιήθηκαν τα αρχεία: **splitVideoFrames.py**, **rgb2gray.py**, **framesDifference.py**, **writeVideo.py**, **motionCompensation.py**, **sumOfAbsolutDifference.py**, **logarithmicSearch.py**, **createMacroblocks.py**.

Όπου η υλοποίηση τους είναι αρκετά παρόμοια (και κάποια ίδια) με την υλοποίηση του θέματος 1

Επίσης, χρησιμοποιήθηκαν οι βιβλιοθήκες: **matplotlib**, **imageio**, **numpy**, **scipy**, **tqdm**.

Το θέμα 2 ακολουθεί την ίδια ακριβώς διαδικασία με το δεύτερο ερώτημα του θέματος 1 με μερικές διαφοροποιήσεις στην διαδικασία αντιστάθμισης κινήσεις όπως: ○ Με την βοήθεια 2 loop και του reference πλαισίου ψάχνουμε να βρούμε την περιοχή όπου βρίσκεται ο ήλιος. Αυτό γίνεται με την βοήθεια εντοπισμού των μαζικών λευκών εικονοστοιχείων.

- Έπειτα αντικαθιστούμε την μάζα φωτεινών εικονοστοιχείων με τα εικονοστοιχεία όπου βρίσκονται γύρο από τον ήλιο.
- Τέλος η διαδικασία όπως και στο ερώτημα 2 θέματος 1 συνεχίζετε κανονικά αλλά με το καινούργιο references frame και το frame πρόβλεψης με σκοπό οι καινούργιες προβλέψεις να περιέχουν την αλλαγή.

● Γράφουμε το δέγμα όπου ο ήλιος απουσιάζει χρησιμοποιώντας την write με το όνομα 'compensatedRemoveObject.mp4' και μορφή .mp4.

Ένα στιγμιότυπο από το αρχικό δείγμα είναι :



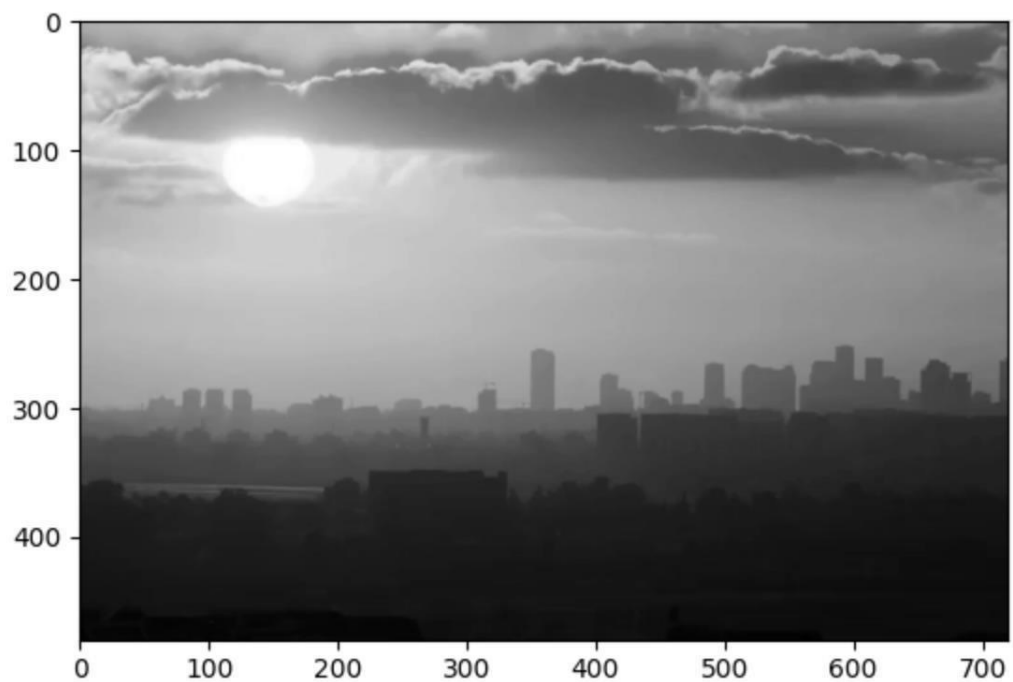
Τελειώνοντας την εκτέλεση έχει την εξής απόκριση στην κονσόλα όπου προχώραγε δυναμικά :

```
100%|██████████| 109/109 [01:57<00:00, 1.08s/it]
```

Στην επόμενη σελίδα θα βρείτε αποτελέσματα εκτέλεσης προγράμματος.

Αποτελέσματα εκτέλεσης προγράμματος:

Μετά την μετατροπή του δείγματος σε ασπρόμαυρο και πριν από την διαδικασία εξαφανίσεις του ήλιου:



Μετά την διαδικασία εξαφανίσεις του ήλιου:



Αποτέλεσμα εγγραφής βίντεο χωρίς τον ήλιο:



Recourses:

- I. Google.com
- II. Stackoverflow.com
- III. Youtube.com
- IV. Ebook - Parag Havaladar, Gerard Medioni - Multimedia Systems_
Algorithms, Standards, and Industry Practices -Course Technology (2009)