

ΕΡΓΑΣΙΑ ΜΑΘΗΜΑΤΟΣ  
«ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ»

Αντώνιος Ελευθέριος Καρναβάς – Π18063

Ιούνιος 2021

## Περιεχόμενα

<b>Θέμα 1</b> .....	3
<b>Ερώτημα 1<sup>ο</sup></b> .....	3
Εκφώνηση.....	3
Δομή Αρχείων.....	3
Παραδοχές .....	3
Παρατηρήσεις .....	4
Ψευδοκώδικας .....	6
<b>Ερώτημα 2<sup>ο</sup></b> .....	8
Εκφώνηση.....	8
Δομή Αρχείων.....	8
Παραδοχές .....	8
Παρατηρήσεις .....	9
Ανάλυση συναρτήσεων .....	11
Ψευδοκώδικας .....	13
<b>Θέμα 2</b> .....	16
Εκφώνηση.....	16
Δομή Αρχείων.....	16
Κώδικας.....	16
Επισήμανση .....	17
Αποτελέσματα .....	17
<b>Θέμα 3</b> .....	20
Εκφώνηση.....	20
Δομή Αρχείων.....	20
Πληροφορίες .....	20
<b>Βιβλιογραφία</b> .....	21
Βιβλιογραφία.....	21
Βιβλιοθήκες Python.....	21



# Θέμα 1

## Ερώτημα 1<sup>ο</sup>

**Εκφώνηση:** Έστω 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. 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 και τον αποκωδικοποιητή.

### Δομή Αρχείων

Ο κώδικας που υλοποιεί το 1ο υποερώτημα του 1ου θέματος βρίσκεται στο αρχείο “1\_1.py” ( σχολιασμένος αναλυτικά στο αρχείο ). Η ακολουθία εικόνων σφάλματος εμφανίζεται με τη μορφή βίντεο στον φάκελο “OutputVideos” με το όνομα “1\_1lionerror.avi”. Ο κωδικοποιητής, αποθηκεύει σε δυαδική μορφή τη πληροφορία στο αρχείο “Encoded1\_1.bin”. Επίσης αποθηκεύονται και στο αρχείο “ Specs1\_1.bin ” κάποια βασικά χαρακτηριστικά του εκάστοτε βίντεο που χρειάζονται κατά την αποκωδικοποίηση. Ο αποκωδικοποιητής διαβάζει αυτά τα αρχεία και ανακατασκευάζει το αρχικό βίντεο και το αποθηκεύει στον φάκελο “OutputVideos” με το όνομα “ 1\_1lion.avi”. Ο κωδικοποιητής καθώς και ο αποκωδικοποιητής βρίσκονται στο ίδιο .py αρχείο και εκτελούνται διαδοχικά.

### Παραδοχές

Κατά την κωδικοποίηση, η ακολουθία εικόνων σφάλματος αποθηκεύονται στο αρχείο “Encoded1\_1.bin” και τα χαρακτηριστικά του βίντεο ( σύνολο frames, ύψος βίντεο, πλάτος βίντεο, frames ανά δευτερόλεπτο ) σε άλλο αρχείο με όνομα “ Specs1\_1.bin ”. Αυτό συμβαίνει διότι στη προσπάθειά μου η κωδικοποίηση να γίνει σε κάθε αριθμό κάθε εικονοστοιχείου ( μετατροπή δηλαδή από integer σε binary ) ήταν πολύ χρονοβόρα. Επίσης οι 2 πίνακες δεν είχαν ίδιες διαστάσεις έτσι ώστε να ομαδοποιηθούν σε έναν πίνακα. Επομένως χρησιμοποίησα την μέθοδο που προσφέρει η numpy με όνομα “tofile()” που αποθηκεύει ένα numpy.ndarray σε binary αρχείο ( .bin ). Θα μπορούσε να χρησιμοποιηθεί η μέθοδος numpy.save() που έχει τη δυνατότητα να αποθηκεύσει πολλαπλά numpy.ndarray σε ένα αρχείο αλλά η

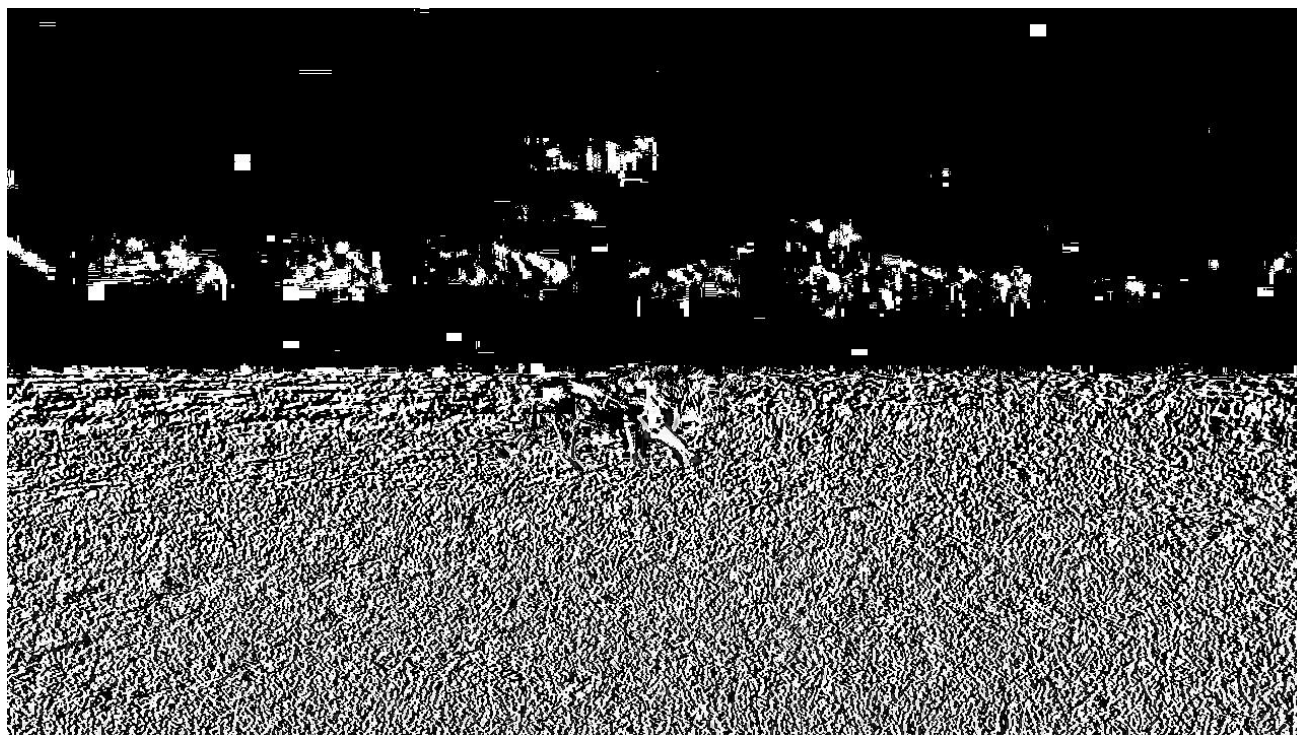
αποθήκευση γίνεται σε συγκεκριμένο format του numpy ( .npz ) και όχι σε μορφή binary.

### Παρατηρήσεις

Το πρώτο συμπέρασμα δημιουργείται αφού εκτελεστεί το πρώτο μέρος του κώδικα που συγκρίνω την εντροπία του αρχικού βίντεο μετετρεμμένο σε αποχρώσεις του γκρι με το σύνολο των εικόνων σφάλματος μαζί με το πρώτο frame αυτούσιο σε αποχρώσεις του γκρι. Τα αποτελέσματα φαίνονται παρακάτω:

```
The error frames sequence was saved in a video named '1_lion.avi' with the entropy of 2.8633392109958242.  
The original video named 'lion.mp4' had an entropy of 4.224262540345935.
```

Παρατηρώ πως η εντροπία του αρχικού ( 4.2242... ) είναι κατά πολύ μεγαλύτερη από αυτή της ακολουθίας εικόνων σφάλματος ( 2.8633... ) αφού στις εικόνες σφάλματος υπάρχει προφανώς λιγότερη πληροφορία. Για παράδειγμα μια εικόνα σφάλματος είναι η παρακάτω:



Βλέπουμε ότι η εικόνα είναι αρκετά μαύρη στον ουρανό αφού μεταξύ εικονοστοιχείων δεν υπάρχει αλλαγή στο χρώμα του ουρανού. Αυτό σημαίνει λίγη πληροφορία στο σύνολο της εικόνας.

Η δεύτερη βασική παρατήρηση είναι πως το αποκωδικοποιημένο βίντεο χάνει σε πληροφορία, ευκρίνεια και ποιότητα αφού ορισμένα σημεία όπως στα σύννεφα φαίνονται pixelated όπως βλέπουμε παρακάτω:

**Αρχικό βίντεο:**



**Αποκωδικοποιημένο βίντεο:**



## Ψευδοκώδικας

### **Χαρακτηριστικά βίντεο:**

Ανάγνωση βίντεο()

Αριθμός frames()

Μέγεθος βίντεο()

Δημιουργία αρχείου για αποθήκευση βίντεο εξόδου()

### **Διαδικασία Ζητούμενου:**

Προηγούμενο frame = Διάβασε πρώτο frame I ()

Μετατροπή frame σε greyscale(Προηγούμενο frame)

Πρόσθεσε στις εικόνες διαφορών (Προηγούμενο frame)

Όσο διαβάζονται frames{

Αν δεν υπάρχει frame { Σταμάτα() }

Εξεταζόμενο frame = Διάβασε επόμενο frame P ()

Μετατροπή frame σε greyscale(Εξεταζόμενο frame)

Διαφορά = Αφαίρεση frames ( Εξεταζόμενο frame - Προηγούμενο frame)

Πρόσθεσε την εικόνα στο καινούριο βίντεο (Διαφορά)

Προηγούμενο frame = Εξεταζόμενο frame

}

Εμφάνισε εντροπία βίντεο εικόνων διαφοράς()

Εμφάνισε εντροπία αρχικού greyscale βίντεο()

### **Κωδικοποίηση:**

Καινούριο βίντεο σε δυαδική μορφή()

Μετατροπή σε δυαδική μορφή(Λίστα με χαρακτηριστικά βίντεο())

Αποθήκευση αρχείων δυαδικής μορφής()

### **Αποκωδικοποίηση:**

Διάβασμα δυαδικού αρχείου με τα χαρακτηριστικά()

Διάβασμα δυαδικού αρχείου με τις εικόνες σφάλματος()

Δημιουργία αρχείου για αποθήκευση βίντεο εξόδου()

Όσο διαβάζονται frames{

Αν εξετάζουμε το πρώτο frame {

Πρόσθεσε την εικόνα στο καινούριο βίντεο (πρώτο frame)

Προηγούμενο frame = Πρώτο frame

}

Αλλιώς {

Εικόνα = Πρόσθεσε ( Προηγούμενο frame + Εξεταζόμενο frame)

Πρόσθεσε την εικόνα στο καινούριο βίντεο (εικόνα)

Προηγούμενο frame = Εικόνα

}



## Ερώτημα 2<sup>ο</sup>

Εκφώνηση: Υλοποιήστε την τεχνική αντιστάθμισης κίνησης για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε macroblocks 16x16, ακτίνα αναζήτησης  $k=16$  και τεχνική σύγκρισης macroblocks της επιλογής σας. Αν θέλετε, μπορείτε να επιταχύνετε τη διαδικασία υλοποιώντας λογαριθμική ή ιεραρχική αναζήτηση. Υπολογίστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων σφαλμάτων. Υλοποιήστε τον κωδικοποιητή και τον αποκωδικοποιητή.

### Δομή Αρχείων

Ο κώδικας που υλοποιεί το 2ο υποερώτημα του 1ου θέματος βρίσκεται στο αρχείο "1\_2.py" ( σχολιασμένος αναλυτικά στο αρχείο ). Η ακολουθία εικόνων σφάλματος εμφανίζεται με τη μορφή βίντεο στον φάκελο "OutputVideos" με το όνομα "1\_2lionerror.avi". Ο κωδικοποιητής, αποθηκεύει σε δυαδική μορφή τη πληροφορία στο αρχείο "Encoded1\_2.bin". Επίσης αποθηκεύονται και στο αρχείο "Specs1\_2.bin" κάποια βασικά χαρακτηριστικά του εκάστοτε βίντεο που χρειάζονται κατά την αποκωδικοποίηση. Τέλος αποθηκεύονται στο αρχείο "Mov\_vectors1\_2.bin" οι θέσεις των macroblocks που θα πρέπει να αντικατασταθούν με τις θέσεις των macroblocks που θα αντικαταστήσουν δηλαδή τα διανύσματα κίνησης. Ο αποκωδικοποιητής διαβάζει αυτά τα αρχεία και ανακατασκευάζει το αρχικό βίντεο και το αποθηκεύει στον φάκελο "OutputVideos" με το όνομα "1\_2lion.avi". Ο κωδικοποιητής καθώς και ο αποκωδικοποιητής βρίσκονται στο ίδιο .py αρχείο και εκτελούνται διαδοχικά.

### Παραδοχές

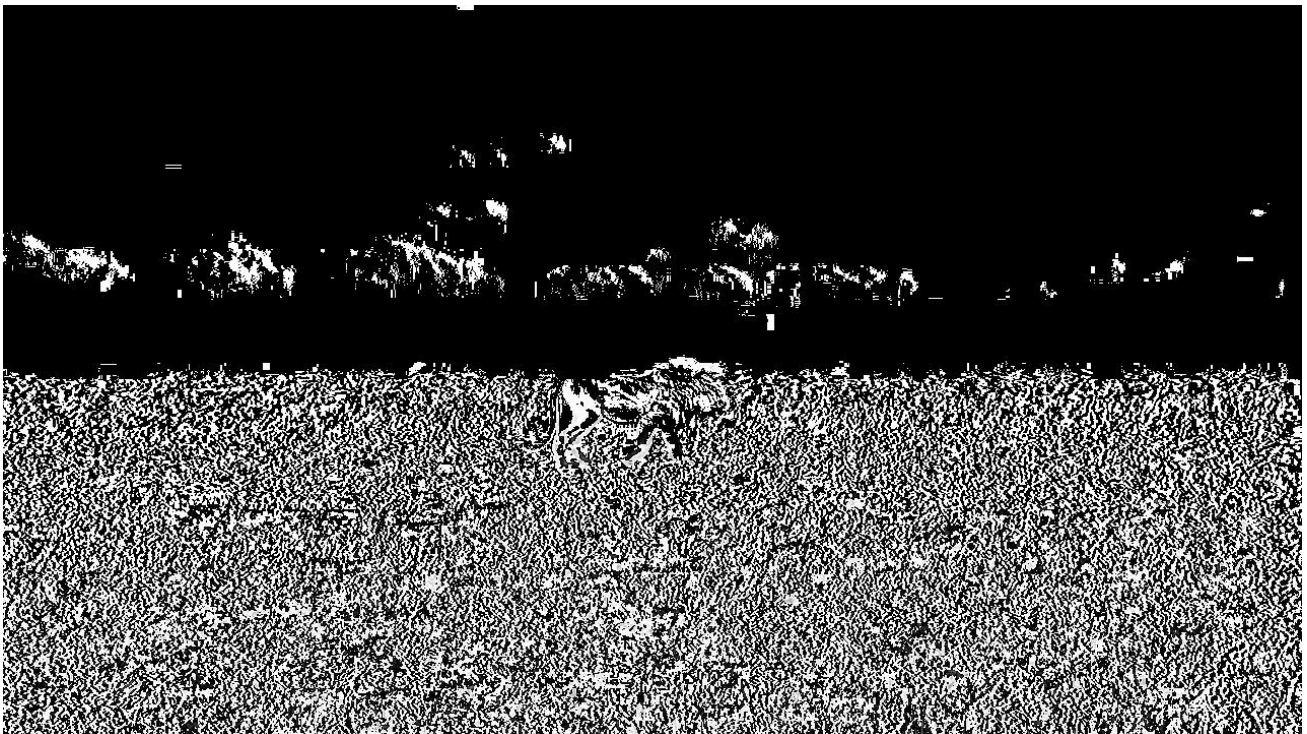
Οι παραδοχές είναι όμοιες με αυτές του 1<sup>ου</sup> υποερωτήματος με τη προσθήκη ότι πλέον υπάρχουν 3 δυαδικά αρχεία, ένα με τις εικόνες σφάλματος ("Encoded1\_2.bin"), ένα με τα χαρακτηριστικά του βίντεο ("Specs1\_2.bin" ) και πλέον ένα και με τα διανύσματα κίνησης ("Mov\_vectors1\_2.bin").

### Παρατηρήσεις

Το πρώτο συμπέρασμα δημιουργείται αφού εκτελεστεί το πρώτο μέρος του κώδικα που συγκρίνω την εντροπία του αρχικού βίντεο μετετρεμμένο σε αποχρώσεις του γκρι με το σύνολο των εικόνων σφάλματος μαζί με το πρώτο frame αυτούσιο σε αποχρώσεις του γκρι. Τα αποτελέσματα φαίνονται παρακάτω:

```
The error frames sequence was saved in a video named '1_2lionerror.avi' with the entropy of 2.8489902747687594.  
The original video named 'lion.mp4' had an entropy of 4.224215833981013.
```

Παρατηρώ πως η εντροπία του αρχικού ( 4.2242... ) είναι κατά πολύ μεγαλύτερη από αυτή της ακολουθίας εικόνων σφάλματος ( 2.8489... ) αφού στις εικόνες σφάλματος υπάρχει προφανώς λιγότερη πληροφορία. Για παράδειγμα μια εικόνα σφάλματος είναι η παρακάτω:



Βλέπουμε ότι η εικόνα είναι αρκετά μαύρη στον ουρανό αφού μεταξύ εικονοστοιχείων δεν υπάρχει αλλαγή στο χρώμα του ουρανού. Αυτό σημαίνει λίγη πληροφορία στο σύνολο της εικόνας.

Σε σύγκριση με την εντροπία του πρώτου ερωτήματος \, παρατηρούμε ότι η εντροπία ακολουθίας εικόνων σφάλματος του πρώτου ερωτήματος είναι 2.8633... ενώ του δεύτερου όπου υλοποιείται η αντιστάθμιση κίνησης η εντροπία είναι λίγο μικρότερη στα 2.8489... . Αυτό σημαίνει ότι ο θόρυβος μεταξύ διαφορών εικόνας μειώνεται άρα η ανακατασκευή εικόνας που είναι υλοποιημένη λειτουργεί σε κάποιο βαθμό.

Η δεύτερη βασική παρατήρηση είναι πως το αποκωδικοποιημένο βίντεο χάνει σε πληροφορία, ευκρίνεια και ποιότητα αφού ορισμένα σημεία όπως στο γρασίδι που γειτονικά blocks μεταξύ τους είναι πολύ όμοια με αποτέλεσμα η ανακατασκευή να περιέχει αρκετά λάθη. Αντιθέτως στο κινούμενο λιοντάρι που τα γειτονικά του blocks είναι ο ουρανός ( διαφορά απόχρωσης του γκρι )δεν υπάρχει τόσο πολύ σφάλμα όπως βλέπουμε παρακάτω:

**Αρχικό βίντεο:**



**Ανακατασκευασμένο βίντεο:**



## Ανάλυση συναρτήσεων

**Συνάρτηση «divide\_to\_macroblocks»:** Η συνάρτηση αυτή δέχεται ως όρισμα την ακτίνα  $k$  που θέλουμε να έχουν τα macroblocks καθώς και το frame που θα σπάσει σε macroblocks. Επιστρέφει έναν πίνακα macroblocks. Σε μια διπλή επανάληψη προσθέτει στον πίνακα macroblocks που επιστρέφει ανά 4 τα στοιχεία κατά γραμμή και κατά στήλη.

**Συνάρτηση «create\_upper\_levels»:** Η συνάρτηση αυτή δέχεται ως όρισμα την εικόνα στόχο και την εικόνα αναφοράς επιστρέφει έναν πίνακα για κάθε μια από τις εικόνες αυτές έναν πίνακα. Αυτός ο πίνακας περιέχει τα 3 επίπεδα της ιεραρχικής αναζήτησης για κάθε εικόνα που θα χρησιμοποιηθούν αργότερα. Η συνάρτηση αυτή υλοποιεί την υποδειγματοληψία για να πάμε στο ανώτερο επίπεδο με την επόμενη συνάρτηση.

**Συνάρτηση «hierarchical\_division»:** Η συνάρτηση δέχεται ως όρισμα μία εικόνα και την υποδειγματοληπτεί κατά 2, δηλαδή αποθηκεύει τις τιμές των pixel μιας εικόνας ανά 2. Για παράδειγμα μια εικόνα μεγέθους 720x1280 θα γίνει 360x640 με το τίμημα βέβαια την ευκρίνεια και την ποιότητα.

**Συνάρτηση «motion\_exist»:** Η συνάρτηση δέχεται ως όρισμα 2 macroblocks και επιστρέφει True αν η διαφορά των macroblock είναι μεγαλύτερη της τάξης του 10% ενώ False αν τα 2 macroblock είναι πάνω από 90% ίδιες. Το παραπάνω το πετυχαίνει αφαιρώντας τα 2 macroblock και μετρώντας μετά τα μηδενικά που προκύπτουν. Δηλαδή για 2 macroblocks 4x4 προκύπτουν από την αφαίρεση τους 16 αριθμοί. Μετρώ το ποσοστό των μηδενικών στο 16 και με αυτό λαμβάνω απόφαση.

**Συνάρτηση «move\_to\_low\_levels»:** Η συνάρτηση αυτή υλοποιεί την αντίθετη λειτουργία από την «create\_upper\_levels» αφού μετακινούμαστε ξανά στο επίπεδο 1 από το επίπεδο 3. Τα macroblocks ανά επίπεδο έχουν μέγεθος:

Επίπεδο 1 -> 16x16

Επίπεδο 2 -> 8x8

Επίπεδο 3 -> 4x4

Η συνάρτηση δέχεται ως παράμετρο έναν πίνακα που παράγεται ανά macroblock από την συνάρτηση «motion\_exist» και τους 2 πίνακες ιεράρχησης που παράγονται από την συνάρτηση «create\_upper\_levels». Για κάθε macroblock που έχουμε

παρατηρήσει κίνηση στο επίπεδο 3 με την συνάρτηση **«motion\_exist»** ελέγχουμε αν ισχύει το ίδιο και για το παρακάτω επίπεδο (επίπεδο 2) που έχει περισσότερη ευκρίνεια αφού είναι κατά 2 λιγότερο υποδειγματοληπτημένο. Το ίδιο συμβαίνει και από το επίπεδο 2 προς το επίπεδο 1. Η συνάρτηση επιστρέφει τα 2 frames (στόχου και αναφοράς) χωρισμένα σε macroblocks  $k=16$  διαστάσεων και έναν πίνακα με τον αριθμό index κάθε macroblock που έχει παρατηρηθεί κίνηση. Ο μέγιστος αριθμός index είναι 3600 διότι  $720/16=45$ ,  $1280/16=80$  και  $45*80=3600$  macroblocks. Αυτό όμως δεν είναι δυνατό στο παράδειγμά μας γιατί για παράδειγμα ο ουρανός δεν μετακινείται.

**Συνάρτηση «calculate\_sad»:** Η συνάρτηση δέχεται ως είσοδο 2 macroblocks και υπολογίζει την μετρική sad και την επιστρέφει. Εδώ επιλέχθηκε αυτή η μετρική καθώς χρειάζεται τους λιγότερους υπολογισμούς σε σχέση για παράδειγμα με την mean square difference ή την mad. Επειδή η διαδικασία είναι αρκετά χρονοβόρα έκανα την επιλογή της ταχύτητας παρά του λίγο καλύτερου αποτελέσματος αναζήτησης.

**Συνάρτηση «sad»:** Δέχεται ως όρισμα 2 εικόνες που είναι σε μορφή macroblocks καθώς και τα indices των macroblocks που παρατηρούνται κίνηση τα οποία παράγονται από την συνάρτηση **«move\_to\_low\_levels»** (1 index macroblock την φορά). Η συνάρτηση ύστερα ελέγχει όλα τα γειτονικά macroblocks (πάνω, κάτω, δεξιά, αριστερά, κάτω διαγώνια δεξιά, κάτω διαγώνια αριστερά, πάνω διαγώνια δεξιά, πάνω διαγώνια αριστερά και το ίδιο που ελέγχουμε) και υπολογίζουμε την μετρική sad του κάθε γειτονικού macroblock με την συνάρτηση **«calculate\_sad»**. Ύστερα διαλέγουμε το macroblock με το μικρότερο sad score και αποθηκεύουμε το index του αντίστοιχου macroblocks στη μεταβλητή block που επιστρέφουμε.

**Συνάρτηση «rebuild\_image»:** Η συνάρτηση δέχεται ως όρισμα τις διαστάσεις του βίντεο (άρα και των frames) και έναν πίνακα από macroblocks και επιστρέφει την εικόνα ανακατασκευασμένη (κολλημένη). Στην ουσία με ένα διπλό loop ενώνει τα macroblocks κατά στήλη για να δημιουργήσει γραμμές και μετά κατά γραμμές για να δημιουργήσει ενωμένη τη φωτογραφία.

**Συνάρτηση «entropy\_score»:** Υπολογίζει την εντροπία με το να μετράει τον αριθμό των μοναδικών τιμών στον πίνακα-όρισμα που το δίνουμε.

## Ψευδοκώδικας

### **Χαρακτηριστικά βίντεο:**

Ανάγνωση βίντεο()

Αριθμός frames()

Μέγεθος βίντεο()

Δημιουργία αρχείου για αποθήκευση βίντεο εξόδου()

### **Διαδικασία Ζητούμενου:**

Προηγούμενο frame=Διάβασε πρώτο frame()

Μετατροπή frame σε greyscale(Προηγούμενο frame)

Πρόσθεσε στις εικόνες διαφορών (Προηγούμενο frame)

Όσο διαβάζονται frames{

Αν δεν υπάρχει frame { Σταμάτα() }

Εξεταζόμενο frame = Διάβασε επόμενο frame ()

Πίνακας επιπέδων ιεραρχικής αναζήτησης Προηγούμενου frame, Πίνακας επιπέδων ιεραρχικής αναζήτησης Εξεταζόμενου frame = **create\_upper\_levels** (Προηγούμενο frame, Εξεταζόμενο frame)

Προηγούμενο frame σε macroblocks= **divide\_to\_macroblocks**(4, Πίνακας επιπέδων ιεραρχικής αναζήτησης Προηγούμενου frame[επίπεδο 3])

Εξεταζόμενο frame σε macroblocks= **divide\_to\_macroblocks**(4, Πίνακας επιπέδων ιεραρχικής αναζήτησης Εξεταζόμενου frame[επίπεδο 3])

Αρχικοποίησε πίνακα macroblocks με κίνηση()

Για κάθε macroblock του Προηγούμενου frame σε macroblocks{

Αν **motion\_exist**(macroblock Προηγούμενου frame, macroblock Εξεταζόμενου frame){

Πρόσθεσε macroblock στον πίνακα macroblocks με κίνηση()

}}

Προηγούμενο frame σε macroblocks, Εξεταζόμενο frame σε macroblocks, πίνακας macroblocks με κίνηση = **move\_to\_low\_levels**(πίνακας macroblocks με κίνηση, Πίνακας επιπέδων ιεραρχικής αναζήτησης Προηγούμενου frame, Πίνακας επιπέδων ιεραρχικής αναζήτησης Εξεταζόμενου frame)

Αρχικοποίησε πίνακα προβλεφθείσας θέσης macroblocks()

Για κάθε macroblock στον πίνακα macroblocks με κίνηση{

Υπολόγισε **sad**(Προηγούμενο frame σε macroblocks, Εξεταζόμενο frame σε macroblocks, πίνακας macroblocks με κίνηση)

Πρόσθεσε παραπάνω index macroblock στον πίνακα προβλεφθείσας θέσης macroblocks()

Κατασκεύασε προβλεφθέν frame αντικαθιστώντας τα αρχικά macroblock με τα index των προβλεφθέντων macroblocks

}

Διανύσματα κίνησης = Πρόσθεσε την αρχική θέση των macroblocks και τη προβλεφθείσα()

Πρόσθεσε στις εικόνες σφάλματος (Εξεταζόμενο frame σε macroblocks – Προβλεφθέν frame)

Πρόσθεσε την εικόνα στο καινούριο βίντεο (εικόνα σφάλματος)

Προηγούμενο frame = Εξεταζόμενο frame

}

### **Κωδικοποιητής:**

Κωδικοποίησε (Εικόνες σφάλματος, χαρακτηριστικά βίντεο, διανύσματα κίνησης)

### Αποκωδικοποιητής:

Αποκωδικοποίησε (Εικόνες σφάλματος, χαρακτηριστικά βίντεο, διανύσματα κίνησης)

Δημιουργία αρχείου για αποθήκευση βίντεο εξόδου()

Όσο διαβάζονται frames(εικόνες σφάλματος){

Αν εξετάζουμε το πρώτο frame {

Πρόσθεσε την εικόνα στο καινούριο βίντεο (πρώτο frame)

Εξεταζόμενο frame = Πρώτο frame

}

Αλλιώς {

Macro1= **divide\_to\_macroblocks**(Εξεταζόμενο frame)

Macro2= **divide\_to\_macroblocks**(Εξεταζόμενο frame)

Για κάθε index macroblock που υπάρχει κίνηση στο frame{

Ανακατασκεύασε στο macro2 την προβλεφθείσα εικόνα()

Ανακατασκευασμένο frame = **rebuild\_image**(ύψος βίντεο, πλάτος βίντεο, πρόσθεσε(προβλεφθέν, εικόνα σφάλματος)

Γράψε στο βίντεο εξόδου(Ανακατασκευασμένο frame)

Εξεταζόμενο frame = Ανακατασκευασμένο frame

}

}



## Θέμα 2

Εκφώνηση: Σε βίντεο της επιλογής σας, διάρκειας 5 s – 15 s, στο οποίο υπάρχει ήπια κίνηση αντικειμένου και κάμερας, επιλέξτε ένα αντικείμενο και εξαφανίστε το. Δημιουργήστε και αποθηκεύστε δηλαδή ένα νέο βίντεο στο οποίο δεν θα υπάρχει το αντικείμενο που επιλέξατε. Για τον σκοπό αυτόν, αξιοποιήστε την τεχνική αντιστάθμισης κίνησης.

### Δομή Αρχείων

Ο κώδικας που υλοποιεί το 2<sup>ο</sup> θέμα βρίσκεται στο αρχείο “2\_1.py” ( σχολιασμένος αναλυτικά στο αρχείο ). Στον κώδικα υλοποιείται η απομάκρυνση της φόρμουλα από το βίντεο “formula.mp4” που βρίσκεται στον φάκελο “OriginalVideos” και το αποτέλεσμα υπάρχει στον φάκελο “OutputVideos” με το όνομα “2\_1formula.avi”. Παρόλα αυτά μέσα στον κώδικα υπάρχουν σχολιασμένες οι γραμμές για να εισάγουμε κάποια βίντεο και να απομακρύνουμε από αυτά μια μπάλα και μια κουκουβάγια, τα αποτελέσματα των οποίων βρίσκονται στον φάκελο “OutputVideos” με τα ονόματα “2\_1ball.avi” και “2\_1owl.avi” αντίστοιχα.

### Κώδικας

Ο κώδικας και οι συναρτήσεις που χρησιμοποιήθηκαν ήταν ίδια με το 2<sup>ο</sup> ερώτημα του 1<sup>ου</sup> θέματος. Η μόνη διαφορά της διαδικασίας ήταν η εξής:

Αποθηκεύουμε το πρώτο frame (πριν μπει το αντικείμενο στο πεδίο όρασης μας ή πριν κουνηθεί) ως background και υλοποιούμε την αντιστάθμιση κίνησης με το background πάντα ως πλαίσιο αναφοράς και βρίσκουμε τα macroblocks που παρατηρήθηκαν κίνηση στο πλαίσιο στόχο. Έτσι αντικαθιστώντας τα macroblocks του εκάστοτε frame (που παρατηρήθηκε κίνηση) με αυτά του background που είναι πάντα στατικά χωρίς το αντικείμενο που κινείται, αναδημιουργείται το βίντεο χωρίς τα κινούμενα αντικείμενα που μπήκαν στο πλάνο.

## Επισήμανση

Η επιλογή του αντικειμένου για απομάκρυνση αν το βίντεο περιέχει πολλά κινούμενα αντικείμενα γίνεται ως εξής:

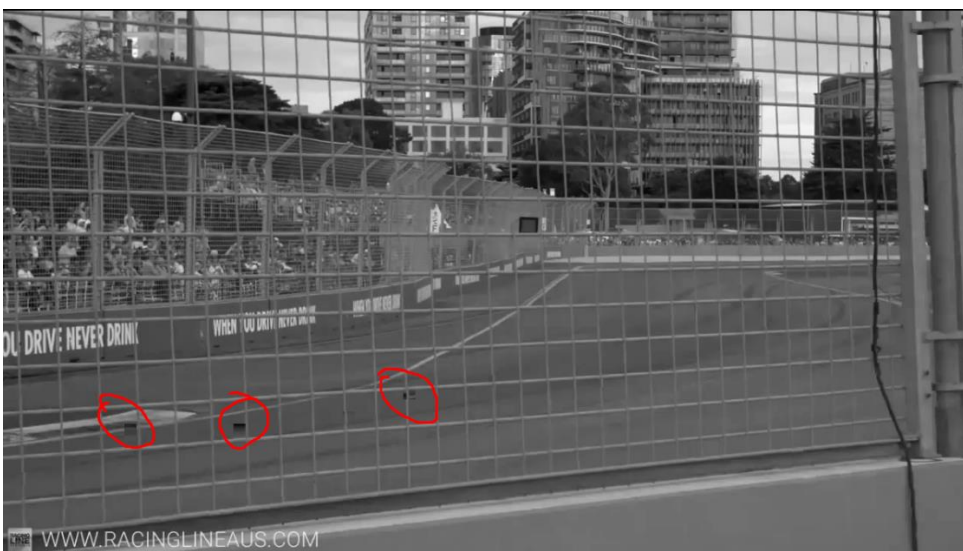
Θέτουμε ως αρχικό πλαίσιο το πρώτο frame του βίντεο αφού κάνουμε κάποιο edit ώστε να διαγράψουμε το αντικείμενο που θα θέλουμε και στη συνέχεια του βίντεο να μην υπάρχει.

## Αποτελέσματα

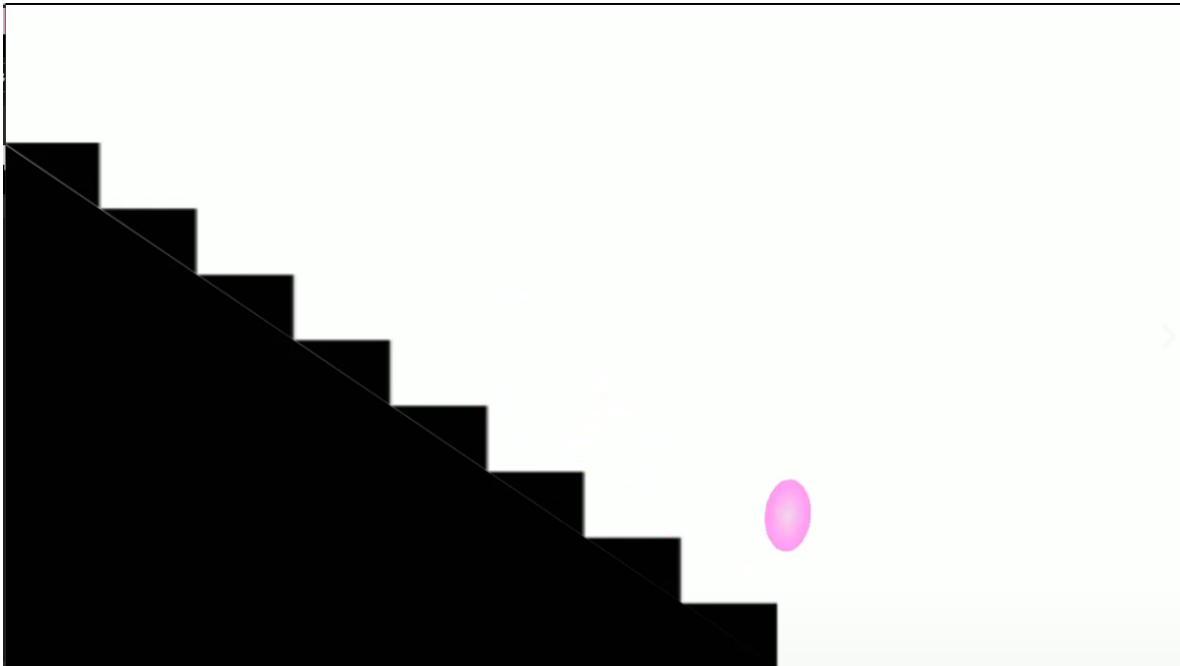
### Αρχικό βίντεο formula:



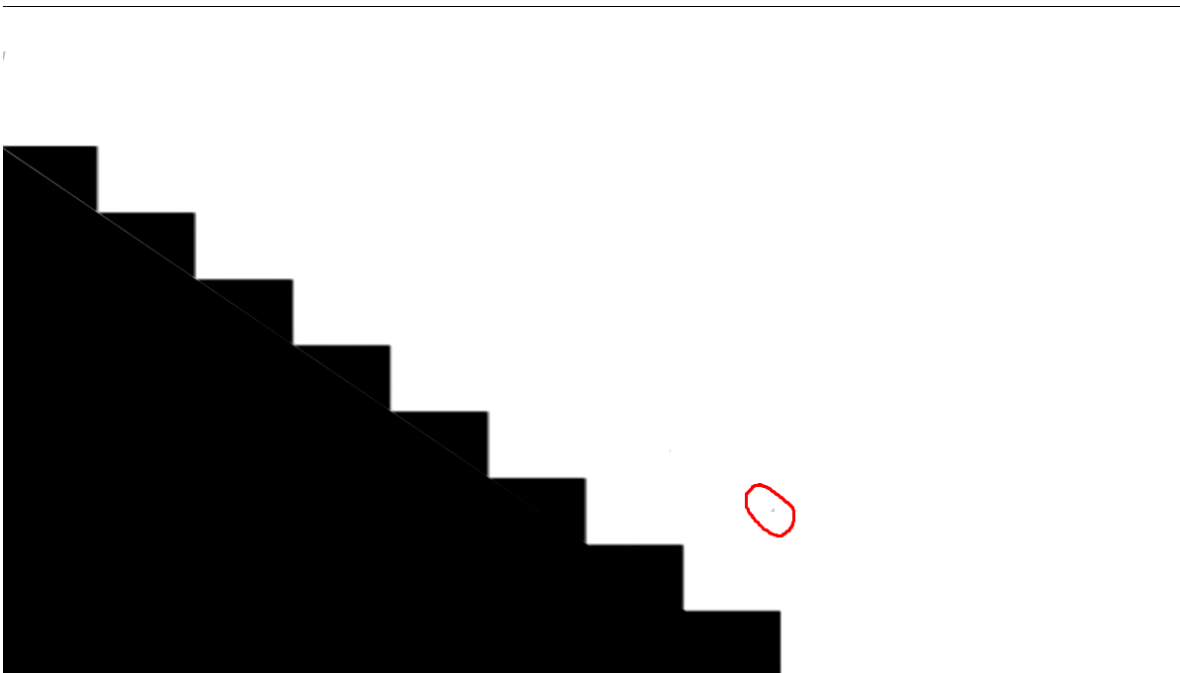
### Τελικό βίντεο formula:



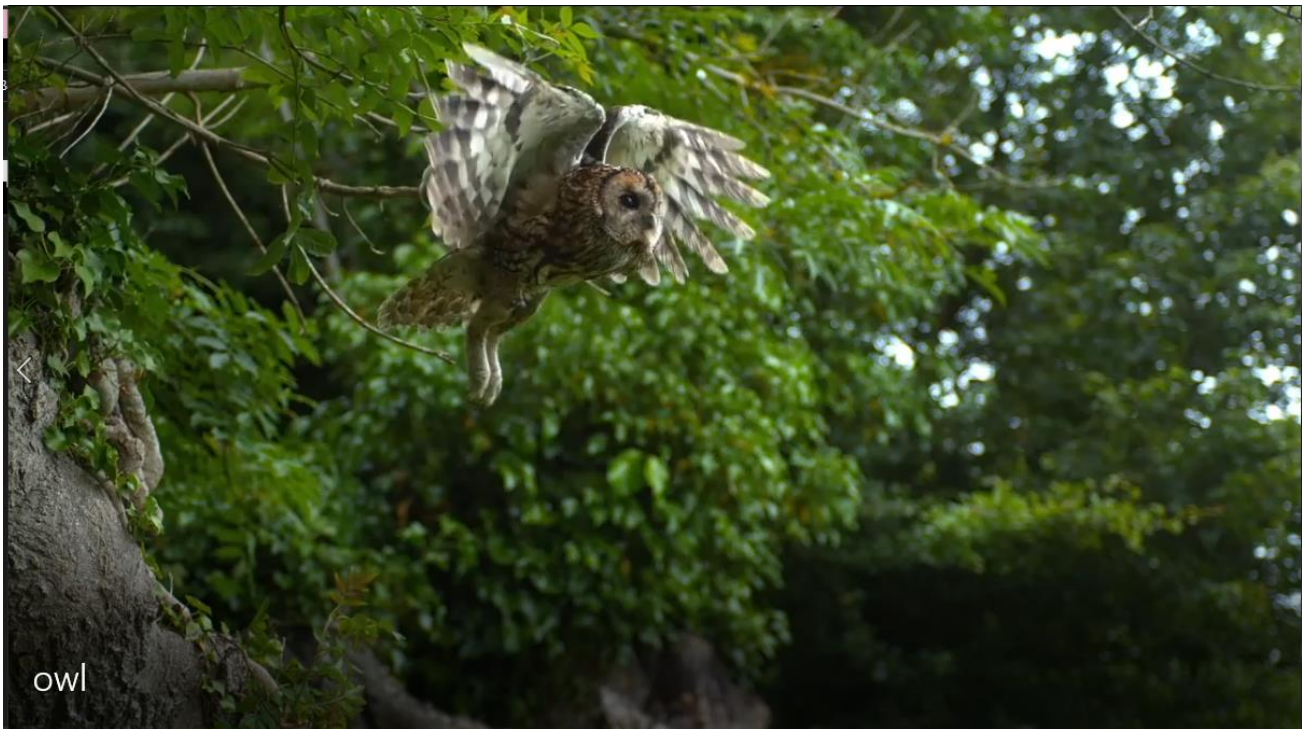
Αρχικό βίντεο μπάλας:



Τελικό βίντεο μπάλας:



**Αρχικό βίντεο κουκουβάγιας:**



**Τελικό βίντεο κουκουβάγιας:**



## Θέμα 3

Εκφώνηση: Να εντοπίσετε τα παρακάτω 10 είδη συμβάντων ήχου στις ηχογραφήσεις: speech, dog, cat, alarm/bell/ringing, dishes, frying, blender, running\_water, vacuum\_cleaner, electric\_shaver\_toothbrush. Για να είναι άξιο λόγου ένα συμβάν, πρέπει να έχει διάρκεια τουλάχιστον 250 ms. Δύο συμβάντα του ίδιου είδους θεωρούνται ξεχωριστά, αν απέχουν τουλάχιστον 150 ms, αλλιώς θεωρούνται ως ένα συμβάν. Δημιουργήστε ένα csv αρχείο με τα αποτελέσματα. Κάθε γραμμή έχει τη μορφή: [filename (string)][,][onset (in seconds)(float)][,][offset (in seconds) (float)][,][event\_label(string)]  
Για παράδειγμα: File1.wav,0.25,0.721,dog. Η πρώτη στήλη είναι το όνομα αρχείου, η δεύτερη η αρχή του συμβάντος σε δευτερόλεπτα (μέχρι 3 δεκαδικά), η τρίτη το τέλος του συμβάντος σε δευτερόλεπτα (μέχρι 3 δεκαδικά) και η τελευταία στήλη το είδος του συμβάντος (βλ. παραπάνω). Αν σε ένα αρχείο, υπάρχουν περισσότερα του ενός συμβάντα, προσθέτουμε γραμμές στο csv, μία ανά συμβάν. Τα ηχητικά συμβάντα μπορεί να επικαλύπτονται. Αν σε κάποιο αρχείο δεν υπάρχει κανένα συμβάν, γράφουμε none στην τελευταία στήλη και -1, -1 στους χρόνους αρχής και τέλους, π.χ., File1.wav,-1,-1,none.

### Δομή Αρχείων

Τα αποτελέσματα του 3<sup>ου</sup> θέματος βρίσκονται στο αρχείο με όνομα «Θέμα3ο.csv» καθώς και στο αρχείο «Θέμα3.txt» σε περίπτωση τεχνικού θέματος.

### Πληροφορίες

Για την υλοποίηση της άσκησης χρησιμοποιήθηκαν:

- Ο audio editor “Audacity”
- Excel για το .csv αρχείο

# Βιβλιογραφία

## Βιβλιογραφία

Η γνώση που αποκτήθηκε έτσι ώστε να ολοκληρωθεί η συγκεκριμένη εργασία, εκτός από τις διαλέξεις του μαθήματος «Συστήματα Πολυμέσων» και τις διαφάνειες χρησιμοποιήθηκε και το βιβλίο του μαθήματος «Συστήματα Πολυμέσων, Αλγόριθμοι, Πρότυπα και εφαρμογές» των Parag Havaladar και Gerard Medioni. Πιο συγκεκριμένα:

Κεφάλαιο 8, σελίδες 253-276 που περιλαμβάνει τη διαδικασία την αντιστάθμισης κίνησης, τον ψευδοκώδικα του κωδικοποιητή και αποκωδικοποιητή καθώς και την θεωρία της ιεραρχικής αναζήτησης.

Επίσης χρησιμοποιήθηκε και βιβλιογραφία από το διαδίκτυο που είναι παρακάτω:

<https://docs.scipy.org/doc/>: Για την εντροπία

<https://docs.opencv.org/master/>: Για τη διαχείριση των frames

<https://www.sciencedirect.com/topics/engineering/motion-compensation> : Για την αντιστάθμιση κίνησης

[https://web.stanford.edu/class/ee398b/handouts/lectures/02-Motion\\_Compensation.pdf](https://web.stanford.edu/class/ee398b/handouts/lectures/02-Motion_Compensation.pdf) : Για την αντιστάθμιση κίνησης

## Βιβλιοθήκες Python

Θέμα 1.1) cv2, numpy, scipy

Θέμα 1.2) cv2, numpy, scipy, os, pickle

Θέμα 2) cv2, numpy, os