



---

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

---

# ΕΡΓΑΣΙΑ

## Συστημάτων Πολυμέσων

Ακαδημαϊκό έτος 2021-2022

ΑΥΓΕΡΙΝΟΣ ΧΡΗΣΤΟΣ - Π19020

ΒΙΤΑΚΗΣ ΑΘΑΝΑΣΙΟΣ - Π19247

ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ - Π19130

ΑΛΕΞΑΝΔΡΗΣ ΙΩΑΝΝΗΣ - Π19006

# Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ .....	3
2. ΑΣΚΗΣΗ 1.1 .....	3
2.1. Εκφώνηση.....	3
2.2. Εκτέλεση .....	3
2.3. Επεξήγηση .....	5
Κωδικοποίηση Huffman.....	5
3. ΑΣΚΗΣΗ 1.2 .....	6
3.1. Εκφώνηση.....	6
3.2. Εκτέλεση .....	6
3.3. Επεξήγηση .....	8
to_macro_blocks(k, arr).....	9
upper_level(source, target) .....	9
hierarchical_search(image).....	9
has_motion(macroblock1, macroblock2).....	9
to_lower_level(movement_blocks,hierimg1,hierimg2) .....	9
calculate_sad(macro1, macro2).....	9
get_sad(mc1, mc2, i).....	9
create_image(height, width, macroblocks).....	10
4. ΑΣΚΗΣΗ 2 .....	10
4.1. Εκφώνηση.....	10
4.2. Εκτέλεση .....	10
4.3. Επεξήγηση .....	12
frame_to_macroblocks(frame, window = 32) .....	12
macroblocks_to_frame(macroblocks).....	12
fit_size(x, window = 32) .....	12
5. ΑΝΑΦΟΡΕΣ ΒΙΒΛΙΟΓΡΑΦΙΑ.....	13

## 1. ΕΙΣΑΓΩΓΗ

Για την υλοποίηση της εργασίας επιλέξαμε να χρησιμοποιήσουμε την γλώσσα προγραμματισμού **Python 3.9** χρησιμοποιώντας μια σειρά από βιβλιοθήκες.

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι:

- Numpy
- Cv2
- Pickle

Τα βίντεο που χρησιμοποιήθηκαν για τις ανάγκες των ασκήσεων τα προμηθευτήκαμε από το διαδίκτυο και βρίσκονται στον φάκελο **videos** .

Επίσης παρέχεται ο φάκελος **screenshots** ο οποίος περιλαμβάνει στιγμιότυπα από την εκτέλεση των προγραμμάτων.

## 2. ΑΣΚΗΣΗ 1.1

### 2.1. 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/αποκωδικοποιητή.

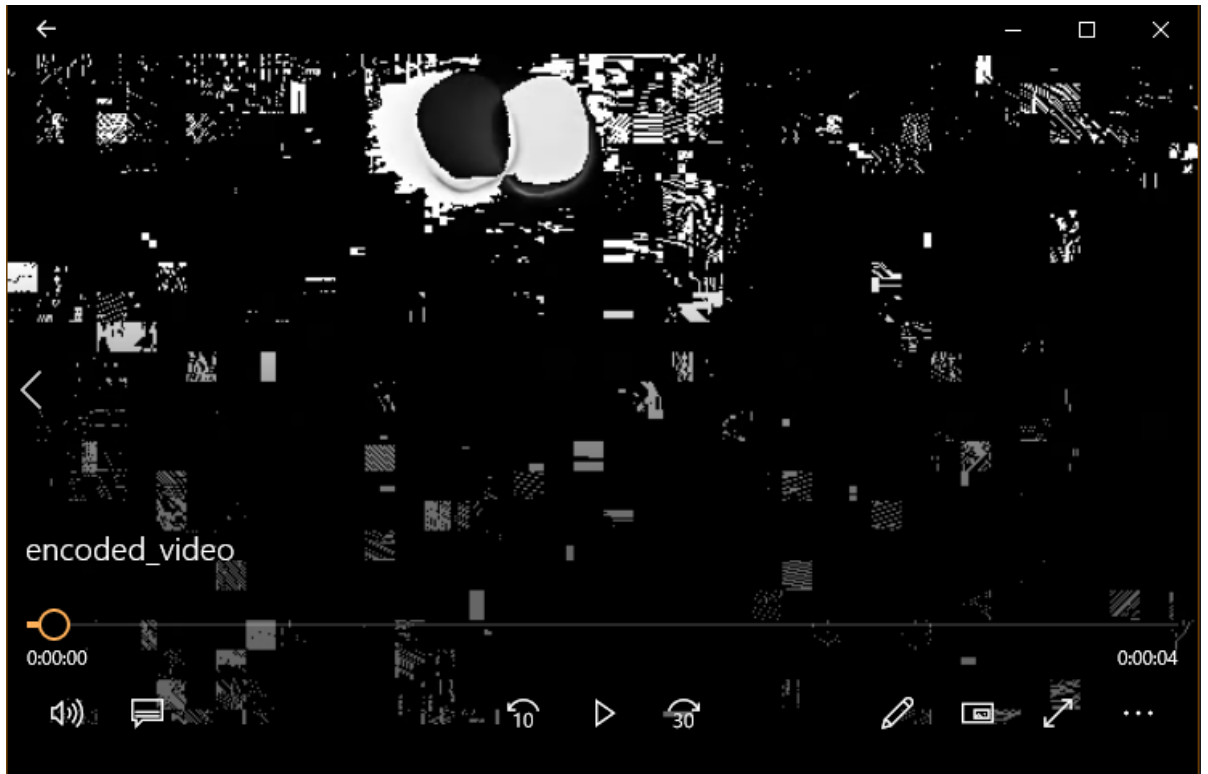
### 2.2. Εκτέλεση

Η επίλυση της άσκησης βρίσκεται στο αρχείο **1\_1.py** και στο συμπληρωματικό αρχείο *huffmancodec.py* Για να το εκτελέσουμε:

- Ανοίγουμε την γραμμή εντολών και μεταβαίνουμε στον φάκελο MultimediaSystems2022
- Εκτελούμε τη εντολή python **1\_1.py**

Από την εκτέλεση του παραπάνω προγράμματος δημιουργούνται τα βίντεο με όνομα

- *encoded\_video.avi*



- *decoded\_video.avi*



### 2.3. Επεξήγηση

Για τις ανάγκες της άσκησης χρησιμοποιήθηκε ένα βίντεο 5 δευτερολέπτων. Το αρχείο βρίσκεται στον φάκελο MultimediaSystems2022\videos\video3.mp4

Αρχικά φορτώνουμε το πρώτο Frame του βίντεο ,έπειτα για να βρούμε τα υπόλοιπα αφαιρούμε το τρέχον Frame με το προηγούμενο και κωδικοποιούμε την διαφορά με την τεχνική Huffman.Έτσι δημιουργείται το βίντεο encoded\_video.avi.

Ύστερα για να ανακατασκευάσουμε το σήμα εκτελούμε την αντίστροφη διαδικασία , καθώς αποκωδικοποιούμε κάθε frame και αθροίζουμε το τρέχον με το προηγούμενο του.

Ο αποκωδικοποιητής έχει κατασκευάσει το σήμα ωστόσο σε κάθε frame περιλαμβάνει κάποιο λάθος το οποίο συσσωρεύεται σε κάθε βήμα και τελικά οδηγούμαστε σε ένα ανακατασκευασμένο βίντεο με αρκετό θόρυβο και έλλειψη λεπτομέρειας.

### Κωδικοποίηση Huffman

Προκειμένου να κωδικοποιήσουμε την ακολουθία εικόνων σφάλματος αρχικά μετατρέπουμε τον δισδιάστατο πίνακα με τις τιμές της κάθε εικόνας σφάλματος σε μονοδιάστατο πίνακα. Έπειτα βρίσκουμε την συχνότητα κάθε συμβόλου για τον πίνακα και την δίνουμε στην συνάρτηση HuffmanCodec.from\_frequencies όπου μας επιστρέφει ένα codec. Τέλος χρησιμοποιούμε το codec για να καλέσουμε την συνάρτηση encode με όρισμα την εικόνα σφάλματος που θέλουμε και την κωδικοποιούμε. Η αντίστοιχη διαδικασία πραγματοποιείτε για αποκωδικοποιήσουμε το κωδικοποιημένο frame με την διαφορά ότι καλούμε την codec.decode().

### 3. ΑΣΚΗΣΗ 1.2

#### 3.1. 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.

Υλοποιήστε την τεχνική αντιστάθμισης κίνησης για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε macroblocks 32x32, ακτίνα αναζήτησης  $k=16$  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.

#### 3.2. Εκτέλεση

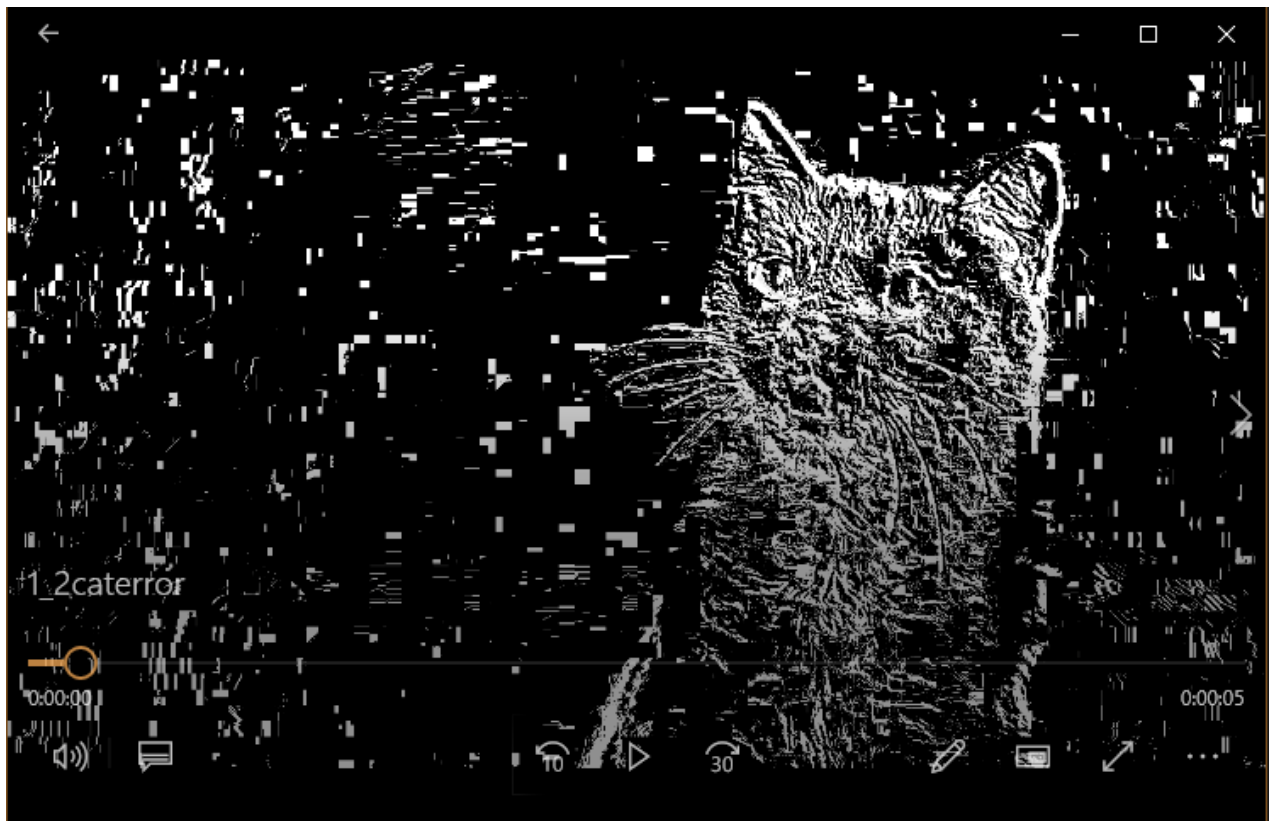
Η επίλυση της άσκησης είναι τα αρχεία **1\_2Encoder.py** , **1\_2Decoder.py** και **functions.py**.

Το αρχείο functions.py περιέχει όλες τις απαραίτητες συναρτήσεις προκειμένου να εκτελεστούν τα 2 αρχεία.

Το αρχείο **1\_2Encoder.py** υλοποιεί την δημιουργία και την κωδικοποίηση της ακολουθίας πρόβλεψης και εικόνων σφαλμάτων. Για να εκτελέσουμε τον κώδικα:

- Ανοίγουμε την γραμμή εντολών και μεταβαίνουμε στον φάκελο **MultimediaSystems2022**
- Εκτελούμε τη εντολή python **1\_2Encoder.py**

Το αποτέλεσμα βρίσκεται στον φάκελο videos ως **1\_2catererror.avi**



Το αρχείο **1\_2Decoder.py** υλοποιεί την αποκωδικοποίηση της ακολουθίας πρόβλεψης και εικόνων σφαλμάτων. Για να εκτελέσουμε τον κώδικα:

- Ανοίγουμε την γραμμή εντολών και μεταβαίνουμε στον φάκελο **MultimediaSystems2022**
- Εκτελούμε τη εντολή `python 1_2Decoder.py`

Το αποτέλεσμα βρίσκεται στον φάκελο:

`MultimediaSystems2022\videos\catDecoded.avi`



### 3.3. Επεξήγηση

Για την επίλυση της άσκησης χρησιμοποιούμε ένα βίντεο 5 δευτερολέπτων στο οποίο βασίζουμε την επεξεργασία που γίνεται. Το αρχείο βρίσκεται στον φάκελο MultimediaSystems2022\videos\cat.mp4

Μετατρέπουμε κάθε frame του video σε αποχρώσεις του γκρι , και έπειτα βρίσκουμε τα ιεραρχικά επίπεδα για το προηγούμενο και το τωρινό frame. Μετά χωρίζουμε το frame σε macroblocks και βρίσκουμε την κινητικότητα σε κάθε macroblock. Αφού βρούμε την κινητικότητα στο πιο χαμηλό επίπεδο έπειτα ελέγχουμε την κινητικότητα και στα πάνω επίπεδα και τα αποθηκεύουμε. Ύστερα υπολογίζουμε την μετρική sad στα block με κινητικότητα και αποθηκεύουμε αυτό με την μικρότερη ως διάνυσμα προβλεπόμενης εικόνας – blocks κίνησης . Τέλος αποθηκεύουμε την εικόνα σφαλμάτων.

Για την αποκωδικοποίηση μετατρέπουμε κάθε frame του βίντεο σφάλματων σε macroblock και εισάγουμε το διάνυσμα κίνησης που αντιστοιχεί στο καθένα. Έπειτα μετατρέπουμε το νέο macroblock με τα διανύσματα κίνησης πίσω σε frame και επαναλαμβάνουμε ως το τέλος του βίντεο.



`to_macro_blocks(k, arr)`

Δέχεται ως όρισμα την ακτίνα *k* που θέλουμε να έχουν τα macroblocks καθώς και το frame. Επιστρέφει έναν πίνακα από macroblocks.

`upper_level(source, target)`

Δέχεται ως όρισμα την εικόνα αναφοράς και την εικόνα στόχο που θέλουμε να φτάσουμε υποδειγματολειπτώντας. Επιστρέφει πίνακα με τα επίπεδα της ιεραρχικής αναζήτησης για την εικόνα αναφοράς και εικόνα στόχο.

`hierarchical_search(image)`

Δέχεται ως όρισμα την εικόνα και την υποδειγματολητεί κατά 2 μέχρι να φτάσει στις διαστάσεις του τελευταίου επιπέδου.

`has_motion(macroblock1, macroblock2)`

Η συνάρτηση ελέγχει αν υπάρχει κίνηση μεταξύ 2 macroblock. Η διαδικασία είναι ότι βρίσκουμε πόσα μηδενικά υπάρχουν στην διαφορά του πρώτου macroblock με του δεύτερου. Αν αυτή είναι άνω της τάξης του 10% τότε θεωρούμε ότι υπάρχει κινητικότητα και η συνάρτηση επιστρέφει true ή false αν δεν ισχύει.

`to_lower_level(movement_blocks, hierimg1, hierimg2)`

Η συνάρτηση αυτή δέχεται ως όρισμα τα διανύσματα κίνησης και τους πίνακες που φτιάξαμε από την συνάρτηση upper\_level και μετακινούμαστε από το τελευταίο επίπεδο προς το πρώτο. Κατά την μετακίνηση μας από το χαμηλότερο επίπεδο προς τα από πάνω ελέγχουμε κάθε φορά αν έχει την ίδια κινητικότητα. Όταν φτάσουμε στο αρχικό η συνάρτηση επιστρέφει τα frames στόχου και αναφοράς χωρισμένα σε macroblocks και έναν πίνακα με την κινητικότητα κάθε επιπέδου.

`calculate_sad(macro1, macro2)`

Δέχεται ως είσοδο 2 macroblocks και υπολογίζει την μετρική sad.

`get_sad(mc1, mc2, i)`

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 to\_lower\_level. Η συνάρτηση ύστερα ελέγχει όλα τα γειτονικά 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.

`create_image(height, width, macroblocks)`

Η συνάρτηση δέχεται ως όρισμα το ύψος, πλάτος και έναν πίνακα από macroblocks και δημιουργεί μια εικόνα. Ο τρόπος που επιτυγχάνεται αυτό είναι ότι πρώτα ενώνουμε τα macroblocks κάθετα και ύστερα οριζόντια.

## 4. ΑΣΚΗΣΗ 2

### 4.1. 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.

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

### 4.2. Εκτέλεση

Η επίλυση της άσκησης βρίσκεται στο αρχείο **2.py**

- Ανοίγουμε την γραμμή εντολών και μεταβαίνουμε στον φάκελο MultimediaSystems2022
- Εκτελούμε τη εντολή python **2.py**

*Το πρώτο παράθυρο απεικονίζει το αρχικό βίντεο.*



*Το δεύτερο παράθυρο απεικονίζει το βίντεο με την εξαφανισμένη μπάλα.*



### 4.3. Επεξήγηση

Για τις ανάγκες της άσκησης χρησιμοποιήθηκε ένα βίντεο 5 δευτερολέπτων. Το αρχείο βρίσκεται στον φάκελο MultimediaSystems2022\videos\video4.mp4

Σκοπός μας είναι η απομάκρυνση της κινούμενης μπάλας από τον διάδρομο bowling με μακρομπλόκ από το πρώτο πλαίσιο του βίντεο πριν εμφανιστούν. Επιλέξαμε το πρώτο πλαίσιο επειδή τα διανύσματα κίνησης είναι σχεδόν μηδενικά λόγω του ότι το παρασκήνιο είναι στατικό. Αφού χωρίσαμε το βίντεο σε μακρομπλόκ μεγέθους 32 το κάθε επόμενο πλαίσιο δανείζεται τα μακρομπλόκ του προηγούμενου από τη μέση και κάτω. Εφόσον το δεύτερο πλαίσιο παίρνει τα μακρομπλόκ παρασκήνιου του πρώτου, το τρίτο του δεύτερου κλπ έχουμε ως αποτέλεσμα την αφαίρεση των κινούμενων χαρακτήρων από το βίντεο.

#### `frame_to_macroblocks(frame, window = 32)`

Η συνάρτηση δέχεται ένα οποιοδήποτε πλαίσιο (frame) και επιστρέφει έναν πίνακα που περιέχει όλα τα μακρομπλόκ. Πιο συγκεκριμένα μεταβάλουμε το εκάστοτε πλαίσιο ώστε να μπορεί να διαιρεθεί σε 32 x 32 μακρομπλοκ και σε περίπτωση που το πλαίσιο έχει μεγαλύτερη διάσταση από το αρχικό συμπληρώνουμε μαύρα εικονοστοιχεία. Τέλος εξετάζουμε κάθε γραμμή και στήλη του νέου πλαισίου που δημιουργήθηκε με βήμα 32(προεπιλογή) και το αποθηκεύουμε.

#### `macroblocks_to_frame(macroblocks)`

Η συνάρτηση αυτή παίρνει σαν παράμετρο ένα αποτέλεσμα από την `frame_to_macroblocks` της μορφής **(γραμμή, στήλη, μακρομπλόκ, 32, 32, 3)** και ξαναφτιάχνει το πλαίσιο. Πιο συγκεκριμένα ανατρέχουμε τον πίνακα των μακρομπλοκ και ξαναβάζουμε τα pixels πίσω σε μορφή πλαισίου. Το αποτέλεσμα μπορεί να προβληθεί ως εικόνα.

#### `fit_size(x, window = 32)`

Επιστρέφει έναν ακέραιο αριθμό  $y$  που διαιρείται από το μέγεθος του μακρομπλόκ (32) και είναι ο πλησιέστερος μεγαλύτερος αριθμός στον  $x$

## 5. ΑΝΑΦΟΡΕΣ ΒΙΒΛΙΟΓΡΑΦΙΑ

Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές

Parag Havaladar, Gerard Medioni

Επιμέλεια Ελληνικής έκδοσης Άγγελος Πικράκης, Ph.D

Σημείωσης μαθήματος "Συστήματα Πολυμέσων"

Διδάσκον Άγγελος Πικράκης, Ph.D