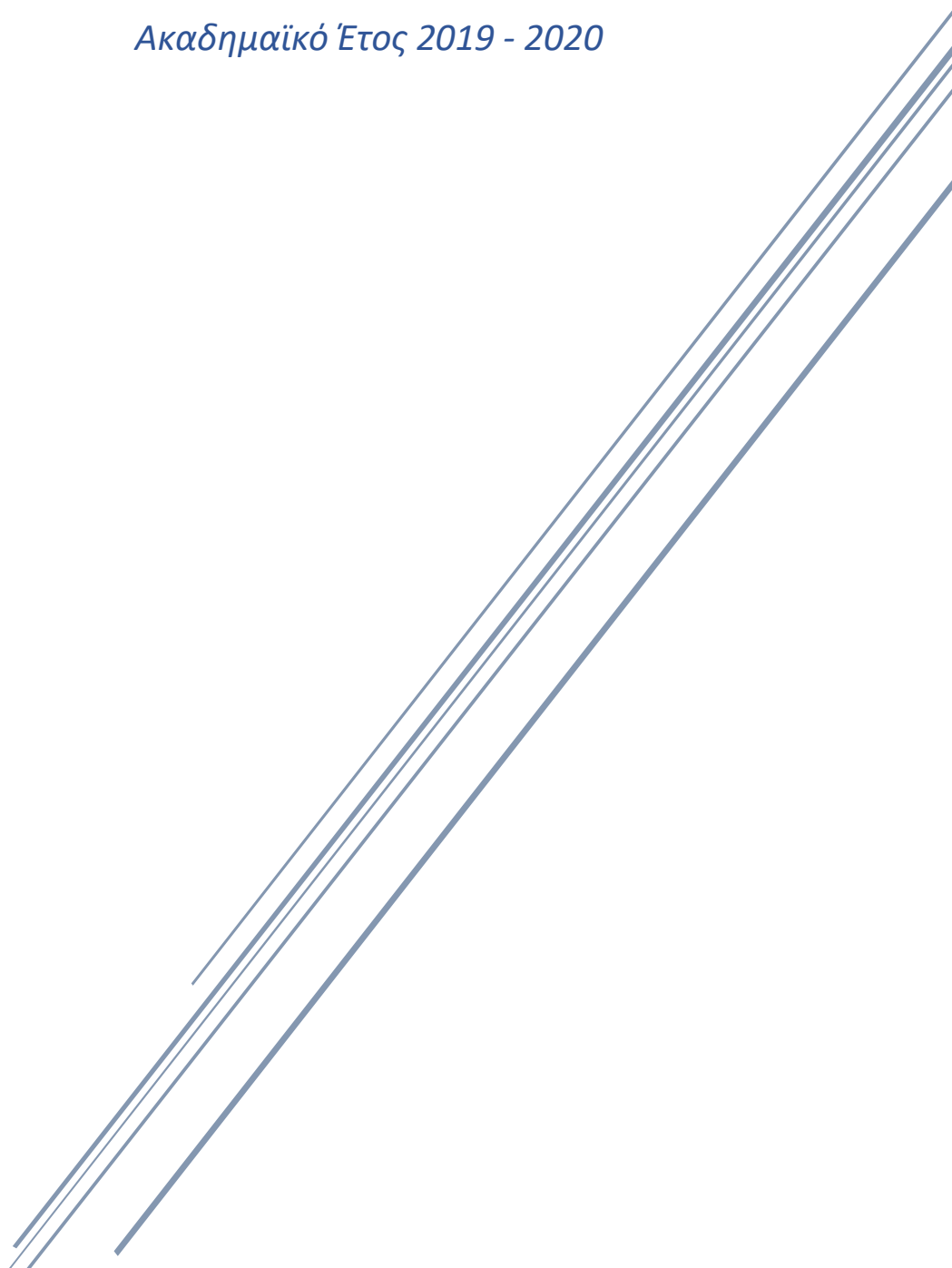


ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ

Α' & Β' Μέρος Απαλλακτικής Εργασία

Ακαδημαϊκό Έτος 2019 - 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

Ορέστης Καραβασίλης, Π17038
Δημήτρης Ματσαγγάνης, Π17068
Αλέξανδρος Σκαρπέλος, Π17122

Περιεχόμενα

A' Μέρος.....	5
Άσκηση 8.17.....	5
1.Εκφώνηση.....	5
2.Κεντρική Ιδέα Υλοποίησης.....	7
2.1. Ερώτημα Α'	7
2.2. Ερώτημα Β'	7
3.Υλοποίηση	9
3.1. Ερώτημα Α'	9
3.1.1. Εισαγωγή Βίντεο	9
3.1.2. Κωδικοποίηση Βίντεο	9
3.1.3. Εγγραφή του Βίντεο	9
3.1.4. Διαχείριση του πλαισίου I	10
3.1.5. Διαδικασία διαχείρισης για τα πλαίσια P	10
3.1.6. Προβολή των πλαισίων σφαλμάτων.....	11
3.1.7. Περιεχόμενο των πλαισίων σφαλμάτων.....	11
3.2 Ερώτημα Β'	12
3.2.1. Εισαγωγή Βίντεο	12
3.2.2. Κωδικοποίηση Βίντεο.....	12
3.2.3. Εγγραφή του Βίντεο	12
3.2.4. Διαχείριση του πλαισίου I	13
3.2.5. Διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία	13
3.2.6. Διαδικασία διαχείρισης για τα πλαίσια P	14
3.2.7. Διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ 16 x 16	14
3.2.8 Διαδικασία υπολογισμού διανυσμάτων κίνησης και λογαριθμική αναζήτηση καλύτερου διανύσματος.....	15
3.2.9 Διαδικασία υπολογισμού της μετρικής SAD	16
3.2.10 Υπολογισμός των μακρομπλόκ σφαλμάτων	16
3.2.11 Ανακατασκευή των πλαισίων	16
3.2.12 Προβολή των πλαισίων σφαλμάτων.....	17
3.2.13. Περιεχόμενο των πλαισίων σφαλμάτων.....	17
4.Αποτελέσματα Υλοποίησης	18



4.1 Ερώτημα Α'	18
4.2 Ερώτημα Β'	19
5.Συμπεράσματα Υλοποίησης.....	22
6. Απαραίτητα εργαλεία για την εκτέλεση.....	23
7. Περιεχόμενα Αρχείου Επίλυσης της Άσκησης	24
8. Οδηγίες Εκτέλεσης Προγραμμάτων	24
Άσκηση 8.18.....	30
1.Εκφώνηση.....	30
2.Κεντρική Ιδέα Υλοποίησης.....	32
3.Υλοποίηση	33
3.1.1. Εισαγωγή Βίντεο	33
3.1.2. Κωδικοποίηση Βίντεο.....	33
3.1.3. Εγγραφή του Βίντεο	33
3.1.4. Διαχείριση του πλαισίου I	34
3.1.5. Διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία	34
3.1.6. Διαδικασία διαχείρισης για τα πλαίσια P	35
3.1.7. Διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ $n \times n$	35
3.1.8 Διαδικασία υπολογισμού διανυσμάτων κίνησης και λογαριθμική αναζήτηση καλύτερου διανύσματος.....	36
3.1.9 Διαδικασία υπολογισμού της μετρικής SAD	37
3.1.10 Διαδικασία εύρεση κίνησης αντικειμένου	37
3.1.11 Αντικατάσταση μακρομπλόκ προσκηνίου - παρασκηνίου	38
3.1.12 Ανακατασκευή των πλαισίων	38
3.1.13 Προβολή των πλαισίων σφαλμάτων.....	39
4. Αποτελέσματα Υλοποίησης	40
5. Συμπεράσματα Υλοποίησης.....	43
6. Σκέψεις Βελτιστοποίησης της Υλοποίησης.....	44
7. Απαραίτητα εργαλεία για την εκτέλεση.....	45
8. Περιεχόμενα Αρχείου Επίλυσης της Άσκησης	46
9. Οδηγίες Εκτέλεσης Προγράμματος.....	46
<i>Β' Μέρος</i>	50
1.Εκφώνηση.....	50
2.Κεντρική Ιδέα Υλοποίησης.....	50

3. Υλοποίηση	51
3.1. Επιλογή πενταψήφιων αριθμών.....	51
3.2. Αντικατάσταση Ψηφίου με το 5	51
3.3. Δημιουργία Εικόνας A	52
3.4. Δημιουργία Εικόνας B	52
3.5. Συνάρτηση Zigzag.....	53
3.6. Συνάρτηση Huffman encoding.....	54
3.7. Υπολογισμός μέσου λόγου συμπίεσης	54
4. Αποτελέσματα Υλοποίησης	55
5. Συμπεράσματα Υλοποίησης.....	55
6. Απαραίτητα εργαλεία για την εκτέλεση.....	56
7. Περιεχόμενα Αρχείου Επίλυσης της Άσκησης	56
8. Οδηγίες Εκτέλεσης Προγράμματος.....	57
Βιβλιογραφία.....	58
Περιεχόμενα Απεσταλμένου Αρχείου	58

Α' Μέρος

Άσκηση 8.17

1. Εκφώνηση

Στην άσκηση αυτή θα υλοποιήσετε την τεχνική της αντιστάθμισης κίνησης και θα μελετήσετε πώς επηρεάζει τα σφάλματα πρόβλεψης. Δείγματα αρχείων βίντεο, μαζί με τη σχετική πληροφορία διαμόρφωσης, παρέχονται στην ενότητα αρχείων προς λήψη του ιστοτόπου www.cengage.com. Εκεί θα βρείτε και τον κώδικα για να διαβάζει και να προβάλλει πλαίσια βίντεο. Υποθέστε ότι το πρώτο πλαίσιο θα είναι πάντα ένα πλαίσιο I και ότι τα υπόλοιπα πλαίσια θα είναι τύπου P. Η υπόθεση αυτή ευσταθεί στην περίπτωση των σύντομων ακολουθιών βίντεο που επεξεργάζεστε, που έχουν μήκος το πολύ 100 πλαισίων.

- a. Στο πρώτο μέρος της άσκησης αυτής, υποθέστε ότι θέλετε να προβλέπετε ολόκληρα P πλαίσια και όχι κατά τμήματα. Η πρόβλεψη κάθε ολόκληρου πλαισίου γίνεται με βάση το προηγούμενο πλαίσιο. Υλοποιήστε μία διαδικασία που να δέχεται είσοδο δύο πλαίσια, υπολογίζει τη διαφορά τους και επιστρέφει ένα πλαίσιο σφαλμάτων. Δεν υπολογίζετε διάνυσμα κίνησης. Να προβάλετε τα πλαίσια σφαλμάτων. Σημειώστε ότι το πληροφοριακό περιεχόμενο του πλαισίου σφαλμάτων θα πρέπει να είναι μικρότερο, συγκρινόμενο με αυτό των πλαισίων.
- b. Στο δεύτερο βήμα θα υλοποιήσετε τεχνική πρόβλεψης κίνησης, η οποία θα υπολογίζει διανύσματα κίνησης ανά μπλοκ. Κάθε μπλοκ θα έχει το τυπικό MPEG μέγεθος 16×16 . Υλοποιήστε μια συνάρτηση που θα δέχεται είσοδο δύο πλαίσια: ένα πλαίσιο αναφοράς, το οποίο θα χρησιμοποιηθεί κατά την αναζήτηση των διανυσμάτων κίνησης, και ένα πλαίσιο στόχος, το οποίο θα προβλεφθεί. Διαιρέστε το πλαίσιο-στόχο σε μακρομπλόκ μεγέθους 16×16 . Εάν το πλάτος και ύψος του πλαισίου δεν είναι πολλαπλάσιο του 16, συμπληρώστε κατάλληλα το πλαίσιο με μαύρα εικονοστοιχεία. Για κάθε μπλοκ στο πλαίσιο-στόχο, ανατρέξτε

στην αντίστοιχη θέση στο πλαίσιο αναφοράς και βρείτε την περιοχή που δίνει το καλύτερο ταίριασμα, όπως έχει εξηγηθεί στο κείμενο του κεφαλαίου. Χρησιμοποιήστε τη μετρική SAD σε περιοχή αναζήτησης που προκύπτει για $k = 16$, έτσι ώστε τα διανύσματα κίνησης να έχουν μέγεθος το πολύ 16 εικονοστοιχείων ως προς κάθε κατεύθυνση. Με βάση το μπλοκ πρόβλεψης, υπολογίστε το μπλοκ σφαλμάτων ως τη διαφορά μεταξύ του αρχικού μπλοκ και του προβλεφθέντος. Αφού αυτή η διαδικασία ολοκληρωθεί για όλα τα μπλοκ, θα προκύψει ένα πλαίσιο σφαλμάτων. Να γίνει η πρόβλεψη όλων των πλαισίων σφαλμάτων. Θα διαπιστώσετε ότι τα πλαίσια σφαλμάτων εμφανίζουν σημαντικά μικρότερη εντροπία σε σύγκριση με τη προηγούμενη περίπτωση, μολονότι απαιτείται περισσότερος χρόνος για τον υπολογισμό τους.

2. ΚΕΝΤΡΙΚΗ ΙΔΕΑ ΥΛΟΠΟΙΗΣΗΣ

2.1. ΕΡΩΤΗΜΑ Α΄

Διαβάζοντας την εκφώνηση της εργασίας και συγκεκριμένα του πρώτου ερωτήματος γίνεται αντιληπτό ότι αναφερόμαστε σε μία είσοδο δύο πλαισίων και μας ζητείται ένα πρόγραμμα, το οποίο θα υπολογίζει τη διαφορά τους (των δύο πλαισίων) και θα επιστέφει το πλαίσιο σφαλμάτων.

Σκεπτόμενοι τα παραπάνω, δημιουργήσαμε δυο πίνακες όπου εκεί αποθηκεύονται οι τιμές των pixels (εικονοστοιχείων).

Στη συνέχεια, αφαιρούμε αυτούς τους δύο πίνακες μεταξύ τους και με αυτό τον τρόπο να υπολογίζουμε τα πλαίσια σφαλμάτων, τα οποία και τα γράφουμε σε ένα καινούργιο βίντεο (output video), το οποίο θα βρίσκεται στον ίδιο φάκελο με το εκτελέσιμο πρόγραμμα.

Για τη συγκεκριμένη υλοποίηση πάρθηκαν οι ακόλουθες παραδοχές – υποθέσεις:

- Επιλέξαμε ένα βίντεο σχεδόν τριών (3'') δευτερολέπτων, δηλαδή λίγων δευτερολέπτων όπως μας προτάθηκε από εσάς. Στο επιλεγμένο βίντεο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου (κάθετη πτώση ενός ανανά) στα ίδια επίπεδα φωτισμού. Το βίντεο πρέπει να βρίσκεται στον ίδιο φάκελο με το αρχείο του προγράμματος (το αρχείο βίντεο : example.m4v).
- Το βίντεο μετατράπηκε από έγχρωμο σε βίντεο με αποχρώσεις του γκρι (grayscale video).

2.2. ΕΡΩΤΗΜΑ Β΄

Συνεχίζοντας στο δεύτερο ερώτημα της άσκησης 8.17, κατανοούμε ότι το ζητούμενο της είναι η υλοποίηση μιας διαδικασίας, η οποία να δέχεται σαν είσοδο δύο πλαίσια (κατά αντιστοιχία με το πρώτο ερώτημα της άσκησης), τα οποία αν δεν έχουν πλάτος και ύψος πολλαπλάσιο του 16 θα συμπληρώνονται με μαύρα pixels (εικονοστοιχεία). Στη συνέχεια το πλαίσιο - στόχος θα διαιρείται σε μακρομπλόκ μεγέθους 16X16 και με τον υπολογισμό

των διανυσμάτων κίνησης και της μετρικής συνάρτησης S.A.D. (Sum of Absolute Differences), θα υπολογίζεται το μακρομπλόκ σφαλμάτων αλλά ταυτόχρονα και η προβολή των πλαισίων σφαλμάτων. Για την προβολή των πλαισίων σφαλμάτων, ομοίως με το προηγούμενο ερώτημα, κρίνεται γόνιμη η δημιουργία ενός βίντεο, το οποίο θα αντικατοπτρίζει τα αποτελέσματα της εκτέλεσης του προγράμματός μας.

Η ιδέα υλοποίησης για την επίλυση στο παρόν ερώτημα είναι ο έλεγχος των frames που διαβάζονται - ανά δύο - και η μετατροπή τους – οπού κρίνεται σκόπιμο – σε πλαίσια διαστάσεων 16 x 16 (η μετατροπή θα γίνεται όταν δεν είναι ήδη 16 x 16).

Στη συνέχεια, θα γίνεται η διαίρεση του πλαισίου - στόχου σε μακρομπλόκ μεγέθους 16 x 16 και θα υπολογίσουμε τα διανύσματα κίνησης για κάθε ένα από τα μακρομπλόκ.

Έπειτα, θα χρησιμοποιείται η μετρική συνάρτηση S.A.D., η οποία θα είναι υπεύθυνη – όπως αντικατοπτρίζει και το όνομα της - για τον υπολογισμό της απόλυτης διαφοράς μεταξύ της υποψήφιας περιοχής και του τμήματος στόχου.

Τέλος, για την προβολή των frames θα πρέπει αρχικά να ανακατασκευάσουμε τα πλαίσια επαναφέροντάς τα στις αρχική τους διαστάσεις.

Για τη συγκεκριμένη υλοποίηση του δεύτερου ερωτήματος, πάρθηκαν οι ακόλουθες παραδοχές – υποθέσεις:

- Επιλέξαμε ένα βίντεο σχεδόν τριών (3'') δευτερολέπτων. Στο επιλεγμένο βίντεο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου (πτώση ενός ανανά) στα ίδια επίπεδα φωτισμού. Το βίντεο πρέπει να βρίσκεται στον ίδιο φάκελο με το αρχείο του προγράμματος (το αρχείο βίντεο : example.m4v).
- Το βίντεο μετατράπηκε από έγχρωμο σε βίντεο με αποχρώσεις του γκρι (grayscale video).
- Η αναζήτηση και η εύρεση του διανύσματος κίνησης, η οποία χρησιμοποιεί την συνάρτηση S.A.D. , θα γίνει με την βοήθεια της λογαριθμικής αναζήτησης. Με αυτό το τρόπο επιταχύνεται στο σύνολο της η διαδικασία.

3.ΥΛΟΠΟΙΗΣΗ

Ακολουθεί η ενότητα στην οποία θα αναλυθούν εκτενώς τα βήματα που ακολουθήθηκαν για την επίλυση των ερωτημάτων της άσκησης 8.17 .

3.1. ΕΡΩΤΗΜΑ Α΄

Για το πρώτο ερώτημα της άσκησης 8.17 έχουμε τα ακόλουθα:

3.1.1. Εισαγωγή Βίντεο

Αρχικά, χάρις στη βιβλιοθήκη της γλώσσας Python (opencv-python) χρησιμοποιήσαμε τη μέθοδο cv2.VideoCapture, ώστε να διαβάσει το πρόγραμμα μας το βίντεο. Αυτό όμως απαιτεί την εγκατάσταση της βιβλιοθήκης opencv όπως αναφέρεται στην ενότητα 6 που έπεται.

3.1.2. Κωδικοποίηση Βίντεο

Στη συνέχεια της υλοποίησης, ακολουθεί η διαδικασία της κωδικοποίησης του βίντεο, για το οποίο πρέπει να επιστέψουμε τα πλαίσια σφαλμάτων. Ειδικότερα, το παραπάνω γίνεται εφικτό χρησιμοποιώντας την μέθοδο cv2.VideoWriter fourcc() της βιβλιοθήκης cv2 (opencv-python), και το τρόπο κωδικοποίησης 'XVID'.

3.1.3. Εγγραφή του Βίντεο

Στο επόμενο βήμα της υλοποίησης μας, χρησιμοποιούμε τη μέθοδο cv2.VideoWriter (αντίστοιχα της βιβλιοθήκης cv2 (opencv-python)). Με το τρόπο αυτό το πρόγραμμα μας καταφέρνει να διαβάσει τα βασικά στοιχεία που χρειάζεται για την ολοκλήρωση του, τα οποία είναι:

- Το όνομα του αρχείου βίντεο (το οποίο θα δίνεται από το χρήστη κατά την έναρξη της εφαρμογής).
- Τη κωδικοποίηση 'XVID' (βλ. προηγούμενη ενότητα).
- Τα πλαίσια ανά δευτερόλεπτο (frames per second).
- Το ύψος και το πλάτος για κάθε πλαίσιο (μέγεθος πλαισίου – size of frame).

3.1.4. Διαχείριση του πλαισίου I

Λαμβάνοντας ως δεδομένο, ότι το πρώτο πλαίσιο θα είναι πάντα ένα πλαίσιο τύπου I, αφού αποθηκευτεί θα μετατρέπεται σε gray scale με την βοήθεια της συνάρτησης `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)` της βιβλιοθήκης `cv2` (opencv-python).

3.1.5. Διαδικασία διαχείρισης για τα πλαίσια P

Ειδικότερα, για τον υπολογισμό της διαφοράς δύο πλαισίων και τη προβολή των πλαισίων σφαλμάτων είναι αναγκαία συνθήκη να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα από το σύστημα. Όσο, η συνθήκη είναι αληθής το τρέχων πλαίσιο θα είναι το επόμενο πλαίσιο P και η επαναληπτική διαδικασία θα συνεχίζεται (μέχρι να μην υπάρχει πλαίσιο P για διάβασμα).

Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε gray scale με την βοήθεια της μεθόδου `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`, της προαναφερμένης βιβλιοθήκης. Η διαφορά των δύο πλαισίων P, ορίζεται η διαφορά μεταξύ του τωρινού πλαισίου και του προηγούμενου πλαισίου και υπολογίζεται με τη βοήθεια της μεθόδου `np.subtract` (της βιβλιοθήκης `numpy`).

Σε κάθε επανάληψη και για τη διατήρησης της αναδρομικότητας της διαδικασίας διαχείρισης των πλαισίων P, αποθηκεύουμε το τρέχων πλαίσιο ως προηγούμενο μετά από κάθε επανάληψη.

Το κάθε πλαίσιο διαφορών αποθηκεύεται σε έναν πίνακα (array), τον οποίο θα τον αξιοποιήσουμε για την εύρεση του πλαισίου σφαλμάτων (βλ. την επόμενη ενότητα).

Προφανώς, στην περίπτωση που δεν υπάρχει επόμενο πλαίσιο P, η επαναληπτική διαδικασία τερματίζει.

3.1.6. Προβολή των πλαισίων σφαλμάτων

Σχετικά με την προβολή των πλαισίων σφαλμάτων, σε συμφωνία με όσα ειπώθηκαν παραπάνω, επιλέχθηκε η δημιουργία ενός νέου βίντεο (το οποίο θα αποθηκεύεται στον ίδιο φάκελο με το εκτελέσιμο και με όνομα που θα ορίζει ο χρήστης).

Ειδικότερα, για κάθε δύο πλαίσια, θα υπολογίζεται η διαφορά τους και επιστέφεται το πλαίσιο σφαλμάτων, το οποίο θα γράφεται σε βίντεο και μετά το πέρας της επαναληπτικής διαδικασίας θα αποθηκεύεται στον φάκελο με τα αρχεία του κώδικα.

Ακόμη, είναι δυνατό να προβληθεί και από το ίδιο το πρόγραμμα, εάν αυτό επιλεγεί από τον χρήστη στη σχετική προτροπή – ερώτηση του προγράμματος.

Για την υλοποίηση της διαδικασίας αποθήκευσης του βίντεο χρησιμοποιήθηκε η μέθοδος `cv2.VideoWriter` (η οποία “γράφει” το αρχείο στο δίσκο), ενώ για την αναπαραγωγή αυτού η συνάρτηση `cv2.VideoCapture(filename)` (η οποία φορτώνει το ήδη εγγεγραμμένο αρχείο με βάση το όνομα που του είχε ορίσει νωρίτερα ο χρήστης).

3.1.7. Περιεχόμενο των πλαισίων σφαλμάτων

Αποσκοπώντας στην άντληση κάποιου συμπεράσματος σχετικά με το περιεχόμενο των πλαισίων σφαλμάτων, κρίθηκε αναγκαία εκτός από τον υπολογισμό του χρόνου εκτέλεσης (βιβλιοθήκη `time` και εύρεση της μεταβολής του χρόνου) της διαδικασίας, η εύρεση της εντροπίας του τελικού – εξαγόμενου βίντεο. Για τον υπολογισμό αυτής, υλοποιήθηκε η συνάρτηση `entropy calculation`, (για την εύρεση της οποίας χρησιμοποιήθηκε η `entropy` της βιβλιοθήκης `scipy.stats`) η οποία θα καλείται από το αρχείο `function.py`,

όπου βρίσκονται, όπως θα δούμε παρακάτω, στην ολότητά τους οι συναρτήσεις που δημιουργήσαμε.

3.2 ΕΡΩΤΗΜΑ Β΄

Ακολουθούμε την ίδια διαδικασία εκτενής επεξήγησης, που ακολουθήσαμε για το πρώτο ερώτημα, και για το δεύτερο της άσκησης 8.17 και έτσι έχουμε τα ακόλουθα:

3.2.1. Εισαγωγή Βίντεο

Όμοια με το πρώτο ερώτημα της άσκησης 8.17, το πρώτο βήμα για την επίλυση του τιθέμενου προβλήματος, είναι η εύρεση και ανάγνωση του αρχείου βίντεο με το οποίο θα εργαστούμε στη συνέχεια της παρούσας υλοποίησης. Για να επιτευχθεί αυτό χρειάζεται να χρησιμοποιήσουμε τη μέθοδο cv2.VideoCapture, η οποία είναι μέρος της βιβλιοθήκης που σχετίζεται με τα ζητήματα της “υπολογιστικής όρασης” (computer vision, opencv-python).

3.2.2. Κωδικοποίηση Βίντεο

Και η διαδικασία της κωδικοποίηση του βίντεο γίνεται στα πρότυπα του πρώτου ερωτήματος, χρησιμοποιώντας την μέθοδο cv2.VideoWriter_fourcc() της βιβλιοθήκης cv2 (opencv-python), και το τρόπο κωδικοποίησης 'XVID'.

3.2.3. Εγγραφή του Βίντεο

Σημαντικό στάδιο της υλοποίησής μας, είναι ο καθορισμός των παραμέτρων για την εγγραφή του βίντεο. Η μέθοδος που χρησιμοποιούμε για την άντληση των βασικών παραμέτρων είναι η cv2.VideoWriter. Τα βασικά στοιχεία που χρειάζονται για την ολοκλήρωση της διαδικασίας, είναι τα παρακάτω:



- Το όνομα του αρχείου βίντεο (το οποίο θα δίνεται από το χρήστη κατά την έναρξη της εφαρμογής).
- Τη κωδικοποίηση 'XVID'.
- Τα πλαίσια ανά δευτερόλεπτο (frames per second).
- Το ύψος και το πλάτος για κάθε πλαίσιο (μέγεθος πλαισίου – size of frame).

3.2.4. Διαχείριση του πλαισίου I

Λαμβάνοντας ως δεδομένο, ότι το πρώτο πλαίσιο θα είναι πάντα ένα πλαίσιο τύπου I, αυτό θα αποθηκεύεται και θα μετατρέπεται σε gray scale με την βοήθεια της συνάρτησης `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)` της βιβλιοθήκης cv2 (opencv-python). Στο επόμενο βήμα θα παρουσιαστεί η διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία.

3.2.5. Διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία

Η διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία χρειάζεται ως ορίσματα το πλαίσιο (frame) και έναν ακέραιο αριθμό (πχ. 16), ο οποίος είναι υπεύθυνος για την συνθήκη συμπλήρωσης με μαύρα εικονοστοιχεία ένα πλαίσιο.

Θυμίζουμε ότι από την υπόθεση της άσκηση, ένα πλαίσιο του οποίου το ύψος και το πλάτος δεν είναι πολλαπλάσιο του 16, θα πρέπει να δέχεται τη συγκεκριμένη διαδικασία.

Ειδικότερα, για κάθε πλαίσιο θα γίνεται η διαίρεση του ύψους και του πλάτους με το 16. Έπειτα, το αποτέλεσμα πολλαπλασιάζεται με το 16 και η τιμή αυτή αφαιρείται από την αρχική τιμή του ύψους και του πλάτους αντίστοιχα (με τη βοήθεια της βιβλιοθήκης numpy).

Για τη κάθε τιμή που προκύπτει θα γίνεται η κατάλληλη τροποποίηση στο πίνακα των εικονοστοιχείων, γεμίζοντας τις επιπλέον θέσεις με μαύρα εικονοστοιχεία.

3.2.6. Διαδικασία διαχείρισης για τα πλαίσια P

Ειδικότερα, για τον υπολογισμό της διαφοράς δύο πλαισίων και τη προβολή των πλαισίων σφαλμάτων είναι αναγκαία συνθήκη να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα από το σύστημα. Όσο, η συνθήκη είναι αληθής το τρέχων πλαίσιο θα είναι το επόμενο πλαίσιο P και η επαναληπτική διαδικασία θα συνεχίζεται (μέχρι να μην υπάρχει πλαίσιο P για διάβασμα).

Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε gray scale με την βοήθεια της μεθόδου `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`. Η διαφορά των δύο πλαισίων P, ορίζεται η διαφορά μεταξύ του τωρινού πλαισίου και του προηγούμενου πλαισίου και υπολογίζεται με τη βοήθεια της μεθόδου `np.subtract` (της βιβλιοθήκης numpy).

3.2.7. Διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ 16 x 16

Η διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ με διαστάσεις 16 x 16, υλοποιήθηκε μέσω μιας δομής επανάληψης for (for loop statement), χωρίς τη χρήση κάποιας “εξωτερικής” συνάρτησης.

Ειδικότερα, κάθε γραμμή και στήλη του μακρομπλόκ ορίζεται ως ένας πίνακας της μορφής: $[i : i+16, j : j+16]$.

Δηλαδή, σε κάθε επανάληψη ο πίνακας του μακρομπλόκ είναι προγραμματισμένος να έχει πλήθος γραμμών από την τιμή του i έως και την τιμή του i+16, ενώ ομοίως το πλήθος στηλών από την τιμή του j έως και την τιμή του j+16.

Με αυτό το τρόπο επιτυγχάνεται η διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ διαστάσεων 16 x 16.

3.2.8 Διαδικασία υπολογισμού διανυσμάτων κίνησης και λογαριθμική αναζήτηση καλύτερου διανύσματος

Για την επιτυχή επίλυση του ζητούμενου προβλήματος, αναγκαία κρίνεται η διαδικασία του υπολογισμού των διανυσμάτων κίνησης, καθώς και η εφαρμογή της λογαριθμικής αναζήτησης για την εύρεση του καλύτερου διανύσματος.

Αυτό επιτυγχάνεται με την υλοποίηση της – εξωτερικής του εκτελέσιμου (θα βρίσκεται στο φάκελο *functions.py* όπως και όλες οι δημιουργημένες συναρτήσεις) - συνάρτησης compute motion vector.

Η περιοχή αναζήτησης της συνάρτησης καθορίζεται από την παράμετρο αναζήτησης K , η οποία είναι ίση με ον ακέραιο αριθμό 16 ($K = 16$). Στόχος της διαδικασίας εύρεσης του διανύσματος κίνησης είναι να βρεθεί η περιοχή εκείνη του πλαισίου αναφοράς ταιριάζει καλύτερα με το μακρομπλόκ στόχο (να ταιριάζει καλύτερα ως προς όλα τα υποψήφια τμήματα στην περιοχή της κάθε αναζήτησης).

Ο τύπος αναζήτησης του καλύτερου διανύσματος, που χρησιμοποιήθηκε στην επίλυση της συγκεκριμένης άσκησης είναι η λογαριθμική αναζήτηση, καθώς στη ζητούμενη υπόθεση της άσκησης οι υπολογισμοί θα πρέπει να είναι όσο το δυνατόν πιο γρήγοροι, διότι επηρεάζουν σημαντικά στην ολότητα της την παρούσα υλοποίηση.

Ως εκ τούτου, το παραπάνω υλοποιείται ως εξής: η μετρική υπολογίζεται επαναληπτικά σε εννέα διακριτές θέσεις, οι οποίες καθορίζονται από το K

$((K/2, K/2), (0, K/2), (-K/2, K/2), (-K/2, 0), (-K/2, -K/2), (0, -K/2), (K/2, -K/2), (K/2, 0))$.

Εναρκτήριο βήμα είναι ο υπολογισμός της μετρικής για το κέντρο της περιοχής και στη συνέχεια για τις υπόλοιπες περιοχές που προαναφέρονται. Στην επόμενη επανάληψη, το κέντρο της περιοχής αναζήτησης μετακινείται στην καλύτερη θέση που υπολογίστηκε στο προηγούμενο βήμα, ενώ το βήμα αναζήτησης μειώνεται στο μισό.

Η παραπάνω δομή επανάληψης συνεχίζεται μέχρις ότου το μέγεθος του βήματος να γίνει ίσο με 1. Σε αυτή τη περίπτωση, από το καλύτερο ταίριασμα της τελευταίας επανάληψης αναδεικνύεται το καλύτερο διάνυσμα κίνησης.

3.2.9 Διαδικασία υπολογισμού της μετρικής SAD

Στη συνέχεια της υλοποίησής μας, δημιουργήσαμε την - εξωτερική - συνάρτηση `sad` (Sum of Absolute Differences), η οποία υπολογίζει το άθροισμα της απόλυτης τιμής των διαφορών μεταξύ δύο μακρομπλόκ.

Η συγκεκριμένη συνάρτηση δημιουργείται και χρησιμοποιείται αποκλειστικά από το αρχείο συναρτήσεων `function.py` και στην συνάρτηση εύρεσης του διανύσματος κίνησης, `compute_motion_vector`.

3.2.10 Υπολογισμός των μακρομπλόκ σφαλμάτων

Ως μακρομπλόκ σφαλμάτων ορίζεται η διαφορά μεταξύ του αρχικού μακρομπλόκ και του προβλεφθέντος από το πρόγραμμα.

Έτσι, μετά από την εύρεση του καλύτερου διανύσματος κίνησης (βλ. προηγούμενη ενότητα 3.2.8), και με κατευθυντήρια γραμμή τον παραπάνω ορισμό του μακρομπλόκ σφαλμάτων, θα υπολογίσουμε το μακρομπλόκ σφαλμάτων για όλα τα μακρομπλόκ.

3.2.11 Ανακατασκευή των πλαισίων

Δεδομένης της ανάγκης που έχουμε για να δημιουργήσουμε βίντεο με τα πλαίσια σφαλμάτων, θα πρέπει να ακολουθήσουμε την αντίστροφη διαδικασία της διαίρεσης των πλαισίων σε μακρομπλόκ.

Η διαδικασία αυτή θα μπορούσε να τιτλοδοτηθεί ως διαδικασία ενοποίησης όλων των 16 x 16 μακρομπλόκ, επιστρέφοντας με αυτό το τρόπο το αρχικό πλαίσιο, δηλαδή το πλαίσιο σφαλμάτων.

3.2.12 Προβολή των πλαισίων σφαλμάτων

Με βάση τα προαναφερθέντα, και με απώτερο σκοπό τη προβολή των πλαισίων σφαλμάτων, επιλέχθηκε η δημιουργία ενός νέου εξαγόμενου βίντεο.

Αυτό επιτυγχάνεται με τον χωρισμό σε μακρομπλόκ διαστάσεων 16×16 , ανά δύο πλαίσια, για τα οποία θα υπολογίζονται τα διανύσματα κίνησης και θα βρίσκεται το καλύτερο με βάση τη λογαριθμική αναζήτηση (βλ. ενότητα 3.2.8).

Στη συνέχεια, θα υλοποιείται ο υπολογισμός του μακρομπλόκ σφαλμάτων και για τα 16 μακρομπλόκ του πλαισίου, θα προκύπτει το πλαίσιο σφαλμάτων, το οποίο γράφεται στο νέο βίντεο και μετά το τέλος της επαναληπτικής διαδικασίας είτε θα αποθηκεύεται στον φάκελο του εκτελέσιμου αρχείου, είτε θα προβάλλεται από το εκτελέσιμο, στη περίπτωση που αυτό επιλεγεί από τον χρήστη στη σχετική προτροπή του προγράμματος.

Για την υλοποίηση της διαδικασίας αποθήκευσης του βίντεο χρησιμοποιήθηκε η μέθοδος `cv2.VideoWriter` (η οποία “γράφει” το αρχείο στο δίσκο), ενώ για την αναπαραγωγή αυτού η συνάρτηση `cv2.VideoCapture(filename)` (η οποία φορτώνει το ήδη εγγεγραμμένο αρχείο με βάση το όνομα που του είχε ορίσει νωρίτερα ο χρήστης).

3.2.13. Περιεχόμενο των πλαισίων σφαλμάτων

Αποσκοπώντας στην άντληση κάποιου συμπεράσματος σχετικά με το περιεχόμενο των πλαισίων σφαλμάτων, κρίθηκε αναγκαία εκτός από τον υπολογισμό του χρόνου εκτέλεσης (βιβλιοθήκη `time` και εύρεση της μεταβολής του χρόνου) της διαδικασίας, η εύρεση της εντροπίας του τελικού – εξαγόμενου βίντεο. Για τον υπολογισμό αυτής, υλοποιήθηκε η συνάρτηση `entropy calculation`, (για την εύρεση της οποίας χρησιμοποιήθηκε η `entropy` της βιβλιοθήκης `scipy.stats`) η οποία θα καλείται από το αρχείο `function.py`, όπου βρίσκονται, όπως θα δούμε παρακάτω, στην ολότητά τους οι συναρτήσεις που δημιουργήσαμε.

4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

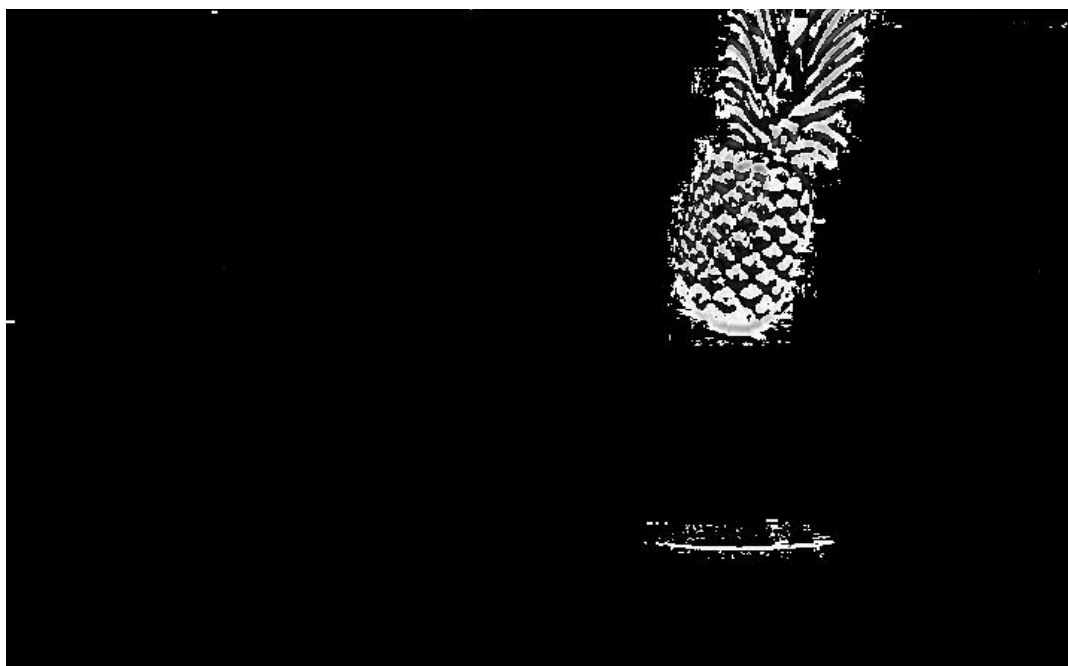
Σε αυτή την ενότητα θα παρατεθούν τα αποτελέσματα της παρούσας υλοποίησης για τα ερωτήματα της άσκησης 8-17.

4.1 ΕΡΩΤΗΜΑ Α΄

Για το πρώτο ερώτημα έχουμε τα ακόλουθα αποτελέσματα:



Εικόνα 1 : Στιγμιότυπο του αρχικού βίντεο *example.m4v* .



Εικόνα 2 : Στιγμιότυπο του εξαγόμενου βίντεο *test817a.avi* .

```
Command Prompt - 8-17a.py
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

Give the file name of the output video: test817a

Creating output video with name test817a.avi...

Success!
The output video, as test817a.avi, has been added in your program's folder.

Entropy: 0.5068367747444783

Time elapsed: 0.4998903274536133s

Do you want to play the output video? (yes/no):
```

Εικόνα 3 : Παράθεση των ενδείξεων απόδοσης της υλοποίησης, για το δοκιμαστικό βίντεο (εντροπία χρόνος διαδικασίας).

4.2 ΕΡΩΤΗΜΑ Β΄

Ακολουθούμε την ίδια διαδικασία για το δεύτερο ερώτημα της άσκησης 8-17.

Παραθέτουμε τα παρακάτω στιγμιότυπα:



Εικόνα 4 : Στιγμιότυπο του αρχικού βίντεο example.m4v



Εικόνα 5 : Στιγμιότυπο του εξαγόμενου βίντεο test817b.avi .

```
Command Prompt - 8-17b.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17

C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17b.py

Select the action you want:
a) Find blocks difference (motion prediction method).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

Give the file name of the output video: test817b

Creating output video with name test817b.avi...
Frame #1 of 60 Completed.
Frame #2 of 60 Completed.
Frame #3 of 60 Completed.
Frame #4 of 60 Completed.
Frame #5 of 60 Completed.
Frame #6 of 60 Completed.
Frame #7 of 60 Completed.
Frame #8 of 60 Completed.
Frame #9 of 60 Completed.
Frame #10 of 60 Completed.
Frame #11 of 60 Completed.
```

Εικόνα 6 : Εκκίνηση της διαδικασίας της άσκησης 8-17b .

```
Command Prompt - 8-17b.py
Frame #40 of 60 Completed.
Frame #41 of 60 Completed.
Frame #42 of 60 Completed.
Frame #43 of 60 Completed.
Frame #44 of 60 Completed.
Frame #45 of 60 Completed.
Frame #46 of 60 Completed.
Frame #47 of 60 Completed.
Frame #48 of 60 Completed.
Frame #49 of 60 Completed.
Frame #50 of 60 Completed.
Frame #51 of 60 Completed.
Frame #52 of 60 Completed.
Frame #53 of 60 Completed.
Frame #54 of 60 Completed.
Frame #55 of 60 Completed.
Frame #56 of 60 Completed.
Frame #57 of 60 Completed.
Frame #58 of 60 Completed.
Frame #59 of 60 Completed.
Frame #60 of 60 Completed.

Success!
The output video, as dddd.avi, has been added in your program's folder.

Entropy: 0.2347560984615053

Time elapsed: 271.9253089427948s

Do you want to play the video? (yes/no):
```

Εικόνα 7 : Ολοκλήρωση της διαδικασίας και παράθεση των ενδείξεων απόδοσης της υλοποίησης, για το δοκιμαστικό βίντεο (εντροπία χρόνος διαδικασίας).

5.ΣΥΜΠΕΡΑΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτή την ενότητα θα παρουσιαστούν τα συμπεράσματα και οι παρατηρήσεις μας μετά την υλοποίηση – εκτέλεση αμφότερων των ερωτημάτων της άσκησης 8.17. Ακολουθούν τα συμπεράσματα στα οποία καταλήξαμε:

- Με την τεχνική της πρόβλεψης πλαισίων χωρίς την τεχνική της πρόβλεψης κίνησης (στο πρώτο ερώτημα – error frames) παρατηρούμε ότι ο χρόνος ολοκλήρωσης της εκτέλεσης του προγράμματος είναι πολύ μικρότερος συγκριτικά με την τεχνική που χρησιμοποιήθηκε στο δεύτερο ερώτημα (με τη πρόβλεψη κίνησης). Με βάση τις παραπάνω εικόνες των αποτελεσμάτων των δύο εκτελέσιμων (βλ. *Εικόνα 3* και *Εικόνα 7*), ο λόγος των χρόνων εκτέλεσης είναι ο παρακάτω:

$$\text{Time_Elapsed 8-17a} / \text{Time_Elapsed 8-17b} = 0.499 / 271.925 = 0.0018$$

- Αντίθετα με την προηγούμενη παρατήρηση μας, το πρώτο ερώτημα έχει μεγαλύτερη εντροπία σχετικά με την εντροπία του δεύτερου ερωτήματος. Ειδικότερα, αυτή η παρατήρηση μας απεικονίζεται και στη παραπάνω εικόνα. Ο λόγος των εντροπιών παρατίθεται παρακάτω:

$$\text{Entropy 8-17a} / \text{Entropy 8-17b} = 0.506 / 0.234 = 2.162$$

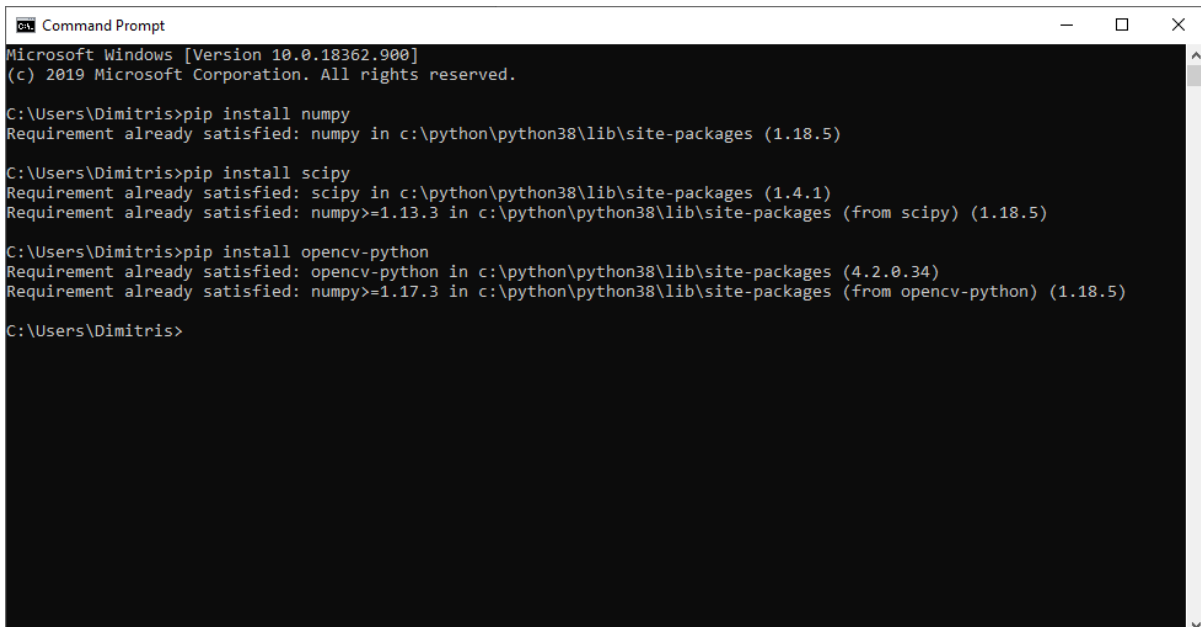
Ως εκ τούτου, με βάση όλα τα παραπάνω συμπεραίνουμε ότι η τεχνική της πρόβλεψης κίνησης (που υλοποιεί τη χρήση της μετρικής S.A.D. και τη τεχνική της λογαριθμικής αναζήτησης), αν και χρειάζεται περισσότερο χρόνο για την εκτέλεσή της δίνει καλύτερα αποτελέσματα (βλ. *λόγο εντροπιών*), επηρεάζοντας θετικά τα σφάλματα πρόβλεψης.

6. ΑΠΑΡΑΙΤΗΤΑ ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

Για την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python** (έκδοση Python 3.8.3). Επιπρόσθετα, οι παρακάτω βιβλιοθήκες είναι απαραίτητες για την εκτέλεση του προγράμματος με βάση τις σχεδιαστικές προδιαγραφές του:

- `os (pathlib)` : για την εύρεση του βίντεο που επιθυμεί ο χρήστης.
- `entropy (scipy)` : για τον υπολογισμό της εντροπίας.
- `numpy` : για το χειρισμό των υπολογιστικών δεδομένων (αριθμητικών) από τη Python.
- `time` : για την καταμέτρηση του χρόνου εκτέλεσης.
- `cv2 (opencv-python)` : για την εισαγωγή των βίντεο (εικόνων) και γενικότερα για το κομμάτι της υπολογιστικής όρασης (computer vision).

Ακολουθεί εικόνα από τη διαδικασία εγκατάστασης των απαραίτητων βιβλιοθηκών.



```
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris>pip install numpy
Requirement already satisfied: numpy in c:\python\python38\lib\site-packages (1.18.5)

C:\Users\Dimitris>pip install scipy
Requirement already satisfied: scipy in c:\python\python38\lib\site-packages (1.4.1)
Requirement already satisfied: numpy>=1.13.3 in c:\python\python38\lib\site-packages (from scipy) (1.18.5)

C:\Users\Dimitris>pip install opencv-python
Requirement already satisfied: opencv-python in c:\python\python38\lib\site-packages (4.2.0.34)
Requirement already satisfied: numpy>=1.17.3 in c:\python\python38\lib\site-packages (from opencv-python) (1.18.5)

C:\Users\Dimitris>
```

Εικόνα 8 : Διαδικασία εγκατάστασης μερικών εκ των απαραίτητων βιβλιοθηκών.

7. ΠΕΡΙΕΧΟΜΕΝΑ ΑΡΧΕΙΟΥ ΕΠΙΛΥΣΗΣ ΤΗΣ ΆΣΚΗΣΗΣ

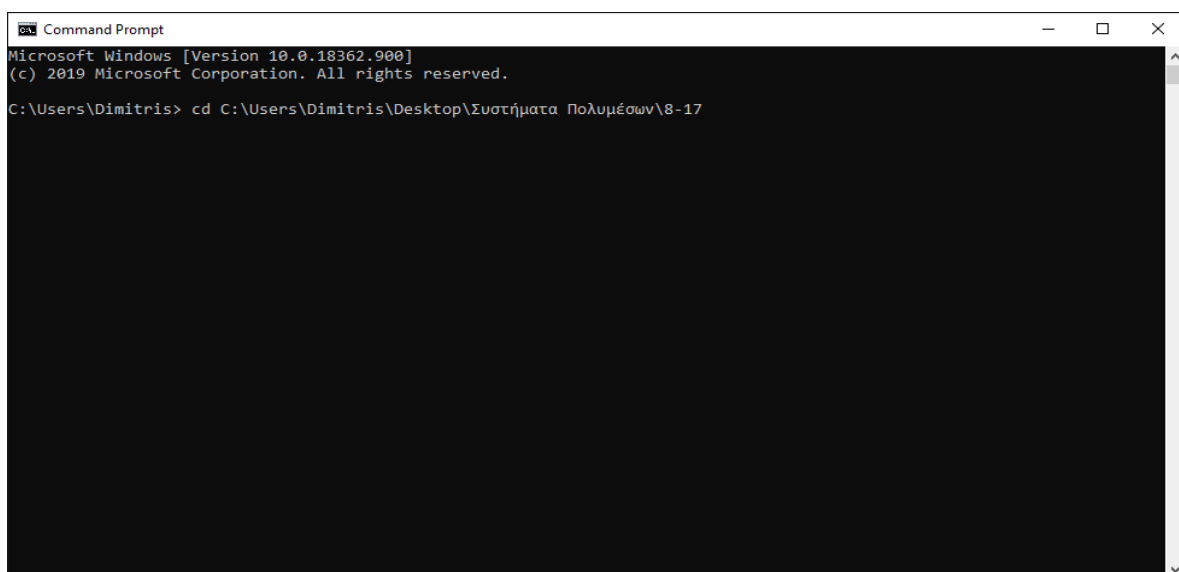
Το αρχείο της παρούσας άσκησης είναι το αρχείο 8-17, το οποίο θα περιέχει τρία ξεχωριστά αρχεία της γλώσσα προγραμματισμού Python:

- 8-17a.py : Η επίλυση του πρώτου ερωτήματος της άσκησης 8.17.
- 8-17b.py : Η επίλυση του δεύτερου ερωτήματος της άσκησης 8.17.
- functions.py : Το αρχείο που εμπεριέχει όλες τις συναρτήσεις που δημιουργήσαμε, η οποίες καλούνται από τα επιμέρους αρχεία επίλυσης (όπου χρειάζονται).

8. ΟΔΗΓΙΕΣ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ

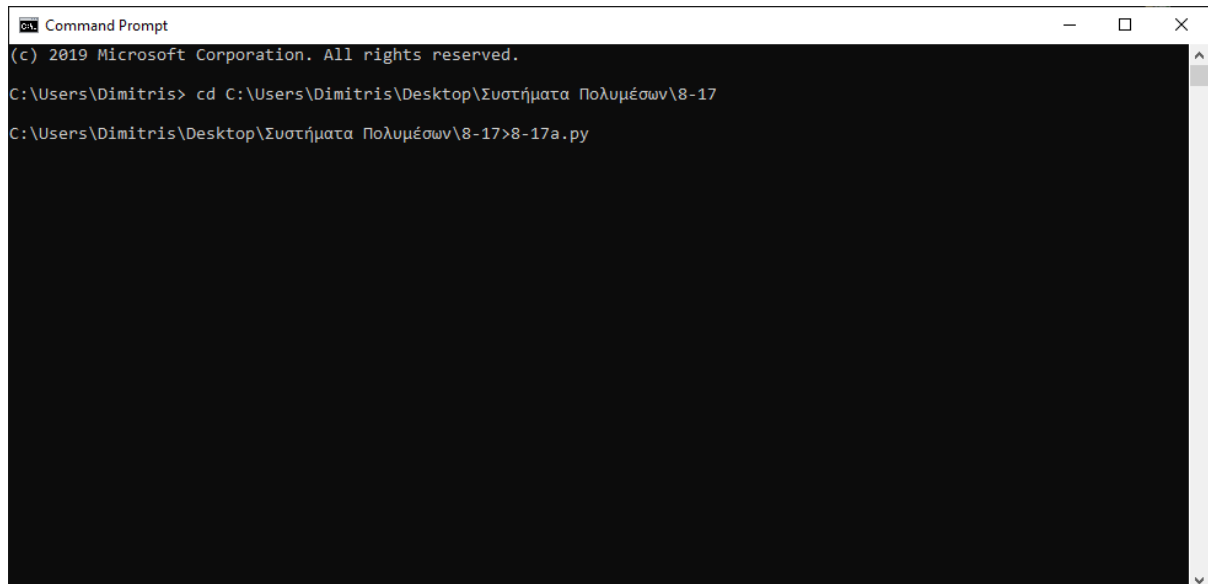
Όπως προαναφέρθηκε παραπάνω κάθε ερώτημα της άσκησης υλοποιείται από ξεχωριστό αρχείο. Αμφότερα τα προγράμματα λειτουργούν παρόμοια στο κομμάτι της αλληλεπίδρασης με το χρήστη (UI – User Interface). Ειδικότερα, κατά την εκκίνηση του προγράμματος ο χρήστης μπορεί να επιλέξει πληκτρολογώντας το αντίστοιχο γράμμα τη λειτουργία που επιθυμεί. Πιο συγκεκριμένα, για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

- Μεταβαίνουμε στον φάκελο, όπου βρίσκεται το εκτελέσιμο:
Συστήματα Πολυμέσων → 8-17 (Εικόνα 9).



Εικόνα 9 : Μετάβαση στο φάκελο του εκτελέσιμου.

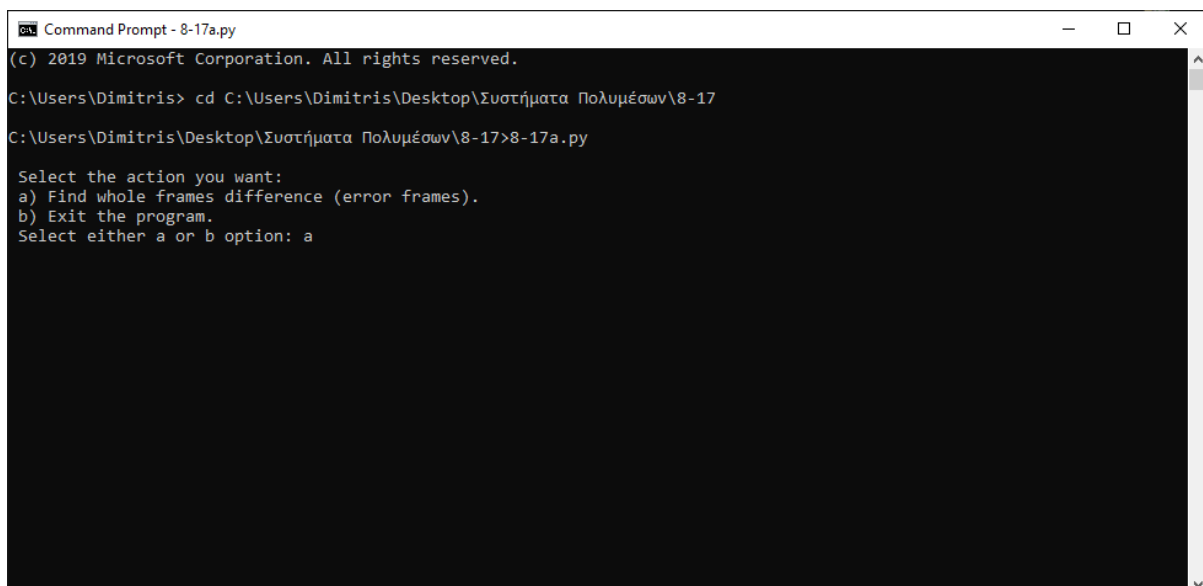
- Πληκτρολογούμε το όνομα του αρχείου 8-17a.py (για το πρώτο ερώτημα ή 8-17b.py για το δεύτερο ερώτημα), ώστε να ανοίξει το αντίστοιχο εκτελέσιμο (Εικόνα 10).



```
Command Prompt
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py
```

Εικόνα 10 : Επιλογή του εκτελέσιμου που επιθυμούμε (εδώ 8-17a).

Όταν ανοίξει το πρόγραμμα εμφανίζεται το μενού αλληλεπίδρασης του προγράμματος, όπου ο χρήστης καλείται να επιλέξει ποια λειτουργία επιθυμεί (Εικόνα 11 και 12).

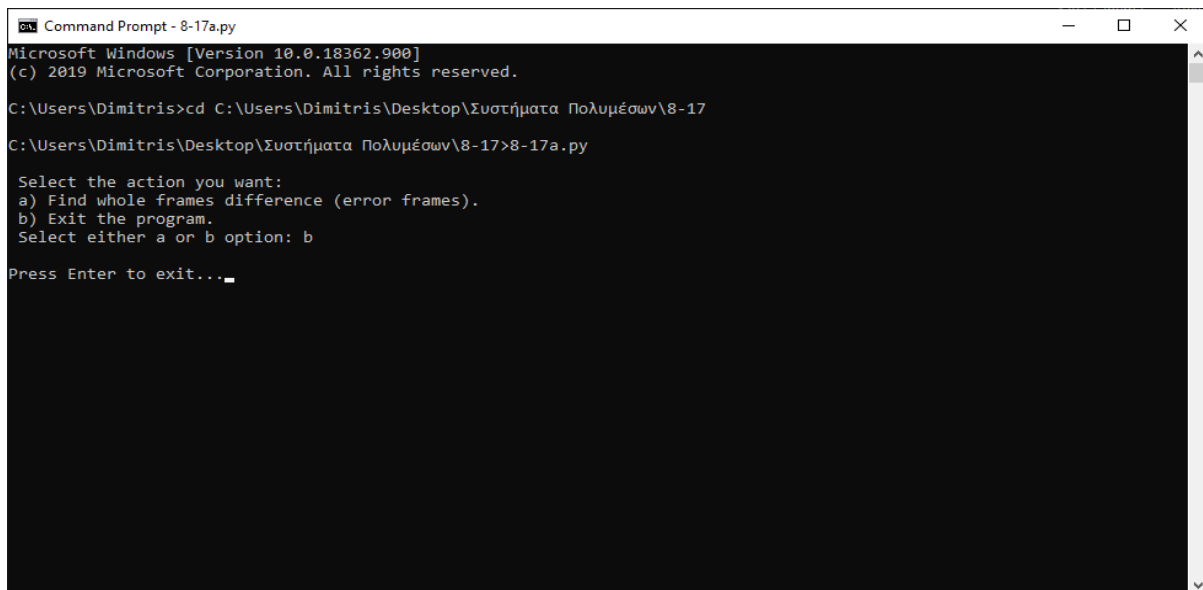


```
Command Prompt - 8-17a.py
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: a
```

Εικόνα 11 : Αλληλεπίδραση με το μενού του εκτελέσιμου (1/2).

Και στα δύο εκτελέσιμα επίλυσης, εάν ο χρήστης επιθυμεί να τρέξει το πρόγραμμα τότε πληκτρολογεί το: a, ενώ πληκτρολογεί το: b, εάν την έξοδο από το πρόγραμμα.



```
Command Prompt - 8-17a.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris>cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

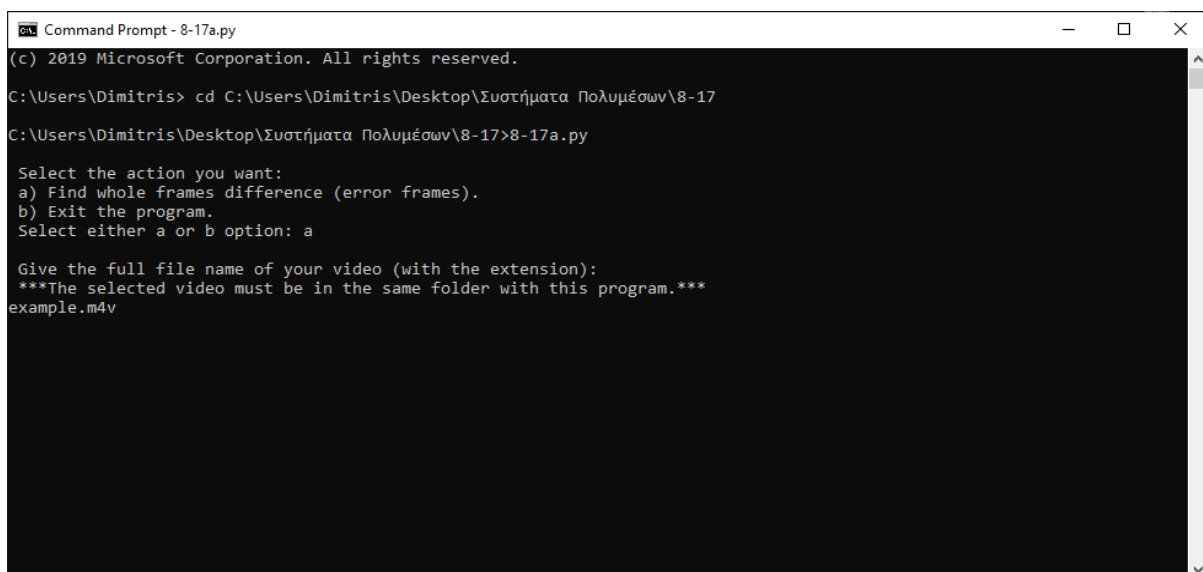
Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: b

Press Enter to exit...
```

Εικόνα 12: Αλληλεπίδραση με το μενού του εκτελέσιμου (2/2).

Στην περίπτωση που επιλεγθεί το a (σε οποιοδήποτε από τα 8-17a.py ή 8-17b.py) τότε για να δημιουργηθεί και να εξαχθεί το αντίστοιχο βίντεο πρέπει να ακολουθηθούν οι παρακάτω διαδικασίες:

- Ο χρήστης αρχικά, θα πρέπει να εισάγει το όνομα του αρχείου, συμπεριλαμβάνοντας την κατάληξη του, (το βίντεο που θα χρησιμοποιηθεί για την εκτέλεση της διαδικασίας π.χ.: *example.m4v*, Εικόνα 13).



```
Command Prompt - 8-17a.py
(c) 2019 Microsoft Corporation. All rights reserved.

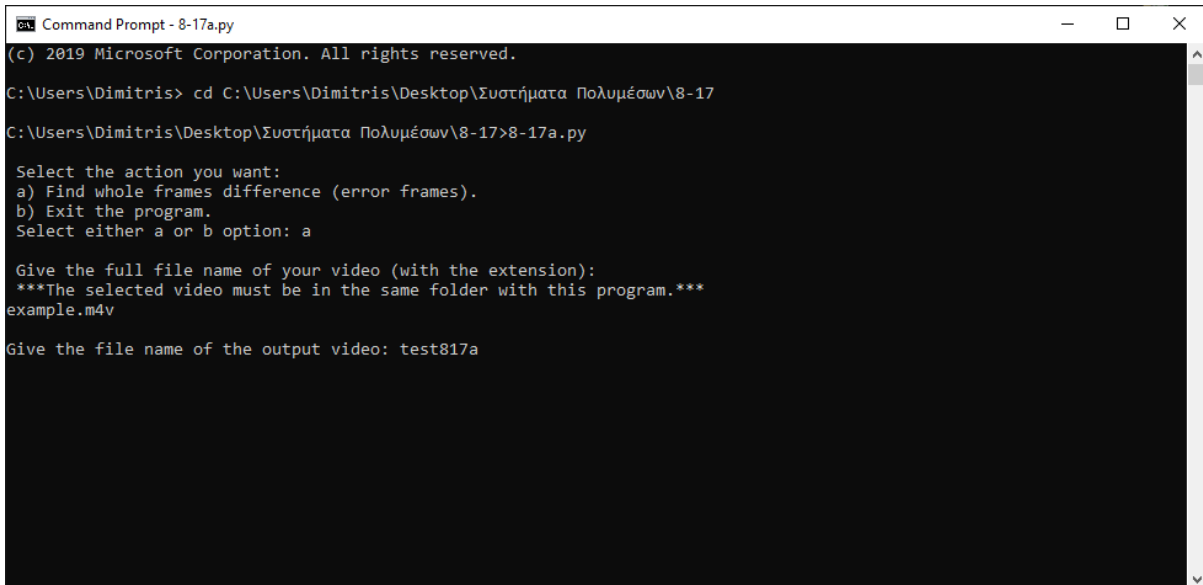
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v
```

Εικόνα 13 : Εισαγωγή του βίντεο που θα υποστεί την επεξεργασία.

- Εισάγει το όνομα του εξαγομένου αρχείου - χωρίς την κατάληξη – για να αποθηκευτεί το νέο βίντεο (π.χ. *test817a.avi*, Εικόνα 14 και Εικόνα 15 και *test817b.avi*, Εικόνα 16 και Εικόνα 17).



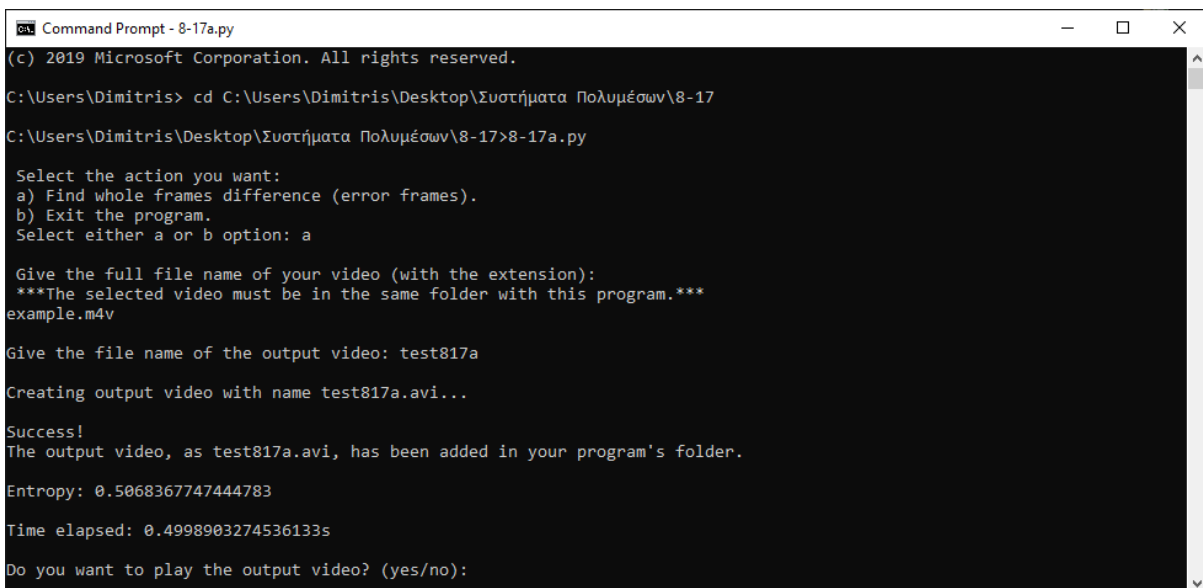
```
Command Prompt - 8-17a.py
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

Give the file name of the output video: test817a
```

Εικόνα 14 : Εισαγωγή του ονόματος του εξαγομένου αρχείου (*test817a.py*).



```
Command Prompt - 8-17a.py
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17a.py

Select the action you want:
a) Find whole frames difference (error frames).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

Give the file name of the output video: test817a

Creating output video with name test817a.avi...

Success!
The output video, as test817a.avi, has been added in your program's folder.

Entropy: 0.5068367747444783

Time elapsed: 0.4998903274536133s

Do you want to play the output video? (yes/no):
```

Εικόνα 15 : Πληροφοριακό περιεχόμενο διαδικασίας.

```
Command Prompt - 8-17b.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17

C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-17>8-17b.py

Select the action you want:
a) Find blocks difference (motion prediction method).
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

Give the file name of the output video: test817b

Creating output video with name test817b.avi...
Frame #1 of 60 Completed.
Frame #2 of 60 Completed.
Frame #3 of 60 Completed.
Frame #4 of 60 Completed.
Frame #5 of 60 Completed.
Frame #6 of 60 Completed.
Frame #7 of 60 Completed.
Frame #8 of 60 Completed.
Frame #9 of 60 Completed.
Frame #10 of 60 Completed.
Frame #11 of 60 Completed.
```

Εικόνα 16 : Έναρξη της διαδικασίας για το δεύτερο ερώτημα (test817b.avi).

```
Command Prompt - 8-17b.py
Frame #40 of 60 Completed.
Frame #41 of 60 Completed.
Frame #42 of 60 Completed.
Frame #43 of 60 Completed.
Frame #44 of 60 Completed.
Frame #45 of 60 Completed.
Frame #46 of 60 Completed.
Frame #47 of 60 Completed.
Frame #48 of 60 Completed.
Frame #49 of 60 Completed.
Frame #50 of 60 Completed.
Frame #51 of 60 Completed.
Frame #52 of 60 Completed.
Frame #53 of 60 Completed.
Frame #54 of 60 Completed.
Frame #55 of 60 Completed.
Frame #56 of 60 Completed.
Frame #57 of 60 Completed.
Frame #58 of 60 Completed.
Frame #59 of 60 Completed.
Frame #60 of 60 Completed.

Success!
The output video, as dddd.avi, has been added in your program's folder.

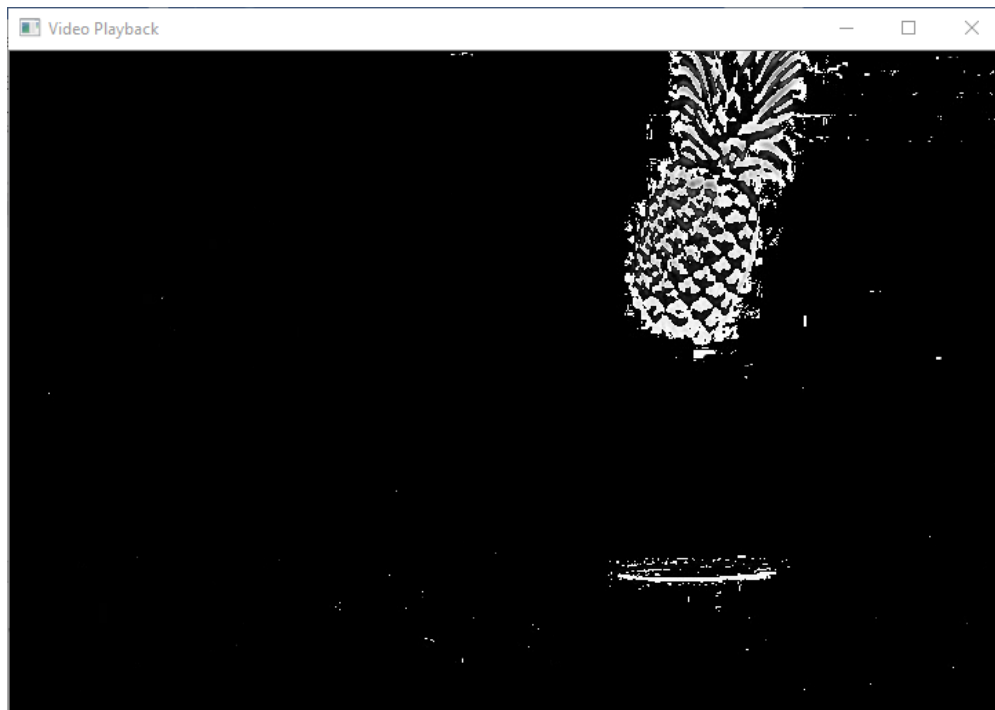
Entropy: 0.2347560984615053

Time elapsed: 271.9253089427948s

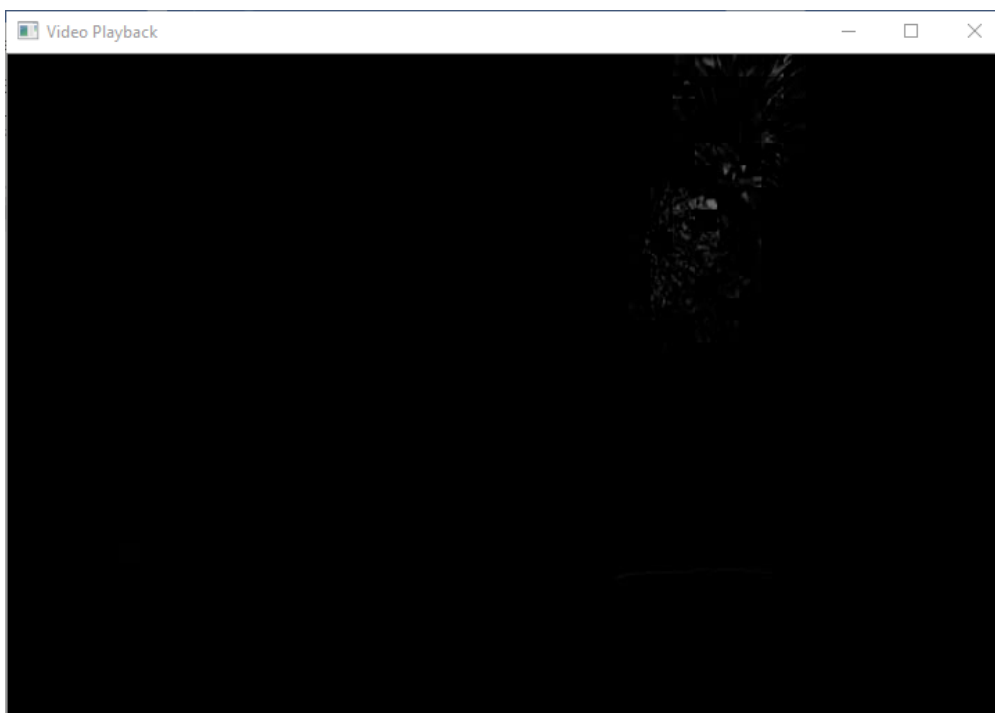
Do you want to play the video? (yes/no):
```

Εικόνα 17 : Λήξη της διαδικασίας και πληροφοριακό περιεχόμενο του δεύτερου ερωτήματος.

- Εάν ο χρήστης επιθυμεί την προβολή του παραπάνω αρχείου, τότε στη σχετική προτροπή του εκάστοτε εκτελέσιμου θα πρέπει να πληκτρολογήσει “yes” (Εικόνα 16 για το πρώτο ερώτημα και Εικόνα 17 για το δεύτερο ερώτημα). Σε αντίθετη περίπτωση πληκτρολογεί “no” .



Εικόνα 18 : Αποτέλεσμα προβολής του πρώτου ερωτήματος.



Εικόνα 19 : Αποτέλεσμα προβολής του δεύτερου ερωτήματος.

Άσκηση 8.18

1. Εκφώνηση

Σε αυτήν την άσκηση θα δείτε ότι η τεχνική της τμηματικής πρόβλεψης με βάση την αντιστάθμιση κίνησης, μπορεί επίσης να χρησιμοποιηθεί σε εφαρμογές εκτός συμπίεσης. Μία τέτοια ενδιαφέρουσα εφαρμογή είναι η απομάκρυνση αντικειμένων ή προσώπων από τη ροή του βίντεο. Για παράδειγμα, έστω ένα βίντεο στο οποίο η κάμερα δεν έχει κινηθεί και το παρασκήνιο είναι σχετικά στατικό, αλλά κινούνται ορισμένα αντικείμενα στο προσκήνιο. Στόχος σας είναι να προσεγγίσετε το αντικείμενο χρησιμοποιώντας μπλοκ και στη συνέχεια να αντικαταστήσετε αυτά τα μπλοκ με παρασκήνιο, σαν να μην ήταν ποτέ παρόν το αντικείμενο. Στη γενική περίπτωση, η λύση είναι πολύ δύσκολη, αλλά στο πλαίσιο αυτής της άσκησης θα επεξεργαστείτε ορισμένες απλούστερες ιδέες. Κατά την υλοποίηση, βεβαιωθείτε ότι μπορείτε να χειρίζεστε το μέγεθος του μπλοκ ως παράμετρο, προκειμένου να ελέγξετε πόσο καλά λειτουργεί ο αλγόριθμος απομάκρυνσης αντικειμένων για διάφορα μεγέθη μακρομπλοκ.

- Κατ' αρχήν, φορτώστε ένα σύνολο πλαισίων βίντεο. Υποθέστε ότι το πρώτο πλαίσιο είναι αποκλειστικά πλαίσιο παρασκηνίου και δεν περιέχει αντικείμενα σε κίνηση. Δοθέντος ενός πλαισίου, n , προχωρήστε στη διαίρεσή του σε μπλοκ. Υπολογίστε ένα διάνυσμα κίνησης ανά μπλοκ με βάση το προηγούμενο πλαίσιο (αναφοράς). Για τα μπλοκ παρασκηνίου, θα πρέπει να προκύψουν μη μηδενικά διανύσματα κίνησης. Παρακολουθήστε όλα αυτά τα διανύσματα κίνησης. Μπορείτε ακόμη και να τα απεικονίσετε για κάθε μπλοκ, καθώς προβάλετε το βίντεο.
- Στη συνέχεια, βρείτε τα μπλοκ που αντιστοιχούν σε μη μηδενικά διανύσματα κίνησης. Αντικαταστήστε κάθε τέτοιο μπλοκ με το αντίστοιχο μπλοκ παρασκηνίου. Τα εικονοστοιχεία παρασκηνίου θα πρέπει να προέρχονται από προηγούμενο πλαίσιο, στο οποίο δεν υπήρχε αντικείμενο σε κίνηση. Η αντικατάσταση όλων αυτών των μπλοκ θα οδηγήσει στην απομάκρυνση των αντικειμένων,

διότι τα αντικείμενα αντικαθίστανται από το παρασκήνιο. Επαναλάβετε τη διαδικασία, χρησιμοποιώντας μακρομπλόκ διαφορετικού μεγέθους.

- Είναι πιθανό να αντιμετωπίσετε ασυνέχειες και παρενέργειες στα όρια των μπλοκ που αντικαθίστανται. Πως θα ελαχιστοποιήσετε αυτά τα φαινόμενα;

Η προτεινόμενη λύση θα λειτουργήσει ικανοποιητικά, μόνο όταν υποθέστε ότι η κάμερα δεν κινείται, όταν τα αντικείμενα που κινούνται υπόκεινται σε λεία κίνηση και ότι δεν παρατηρούνται αλλαγές κατεύθυνσης. Όμως στη γενική περίπτωση, η κάμερα κινείται, τα αντικείμενα κινούνται με όχι αυστηρό τρόπο και επιπλέον, παρατηρούνται αλλαγές στον φωτισμό, καθώς η κάμερα κινείται.

- Αλλά ας υποθέσουμε ότι μπορείτε να εντοπίσετε όλα τα μακρομπλόκ ενός πλαισίου που αντιστοιχούν στο παρασκήνιο. Πως μπορείτε να αξιοποιήσετε αυτό το γεγονός, πλέον της χρήσης αντιστάθμισης κίνησης σε επίπεδο μακρομπλόκ;

2. ΚΕΝΤΡΙΚΗ ΙΔΕΑ ΥΛΟΠΟΙΗΣΗΣ

Στη παρούσα άσκηση μας ζητείται η απόκρυψη κάποιου (ή κάποιων) αντικειμένου ή προσώπου από τη ροή ενός βίντεο. Το αντικείμενο ή το πρόσωπο, που είναι καθορισμένο να αντικατασταθεί, θα κινείται στο προσκήνιο και σκοπός της άσκησης είναι να αντικατασταθεί από το παρασκήνιο χρησιμοποιώντας τη τεχνική της τμηματικής πρόβλεψης με βάση τη μέθοδο της αντιστάθμιση κίνησης.

Ειδικότερα, η κίνηση του αντικειμένου θα εντοπίζεται μέσω μιας συνάρτησης, η οποία θα εφαρμόζει τη τεχνική εύρεσης διανυσμάτων κίνησης (η συνάρτηση αυτή θα βρίσκεται σε εξωτερικό αρχείο και θα καλείται, θα ακολουθήσει ενότητα με ειδική αναφορά στην εν λόγω συνάρτηση).

Τέλος, σημαντικό τεκμήριο για την αποδοτικότητα του αλγορίθμου θα αποτελέσει ο χειρισμός του μέγεθος του μπλοκ, ως παράμετρο.

Για τη συγκεκριμένη υλοποίηση πάρθηκαν οι ακόλουθες παραδοχές – υποθέσεις:

- Επιλέξαμε ένα βίντεο σχεδόν τριών (3) δευτερολέπτων, δηλαδή λίγων δευτερολέπτων όπως μας προτάθηκε από εσάς. Στο επιλεγμένο βίντεο η κάμερα μένει σταθερή και υπάρχει μόνο η κίνηση ενός αντικειμένου (κάθετη πτώση ενός ανανά) στα ίδια επίπεδα φωτισμού. Το βίντεο πρέπει να βρίσκεται στον ίδιο φάκελο με το αρχείο του προγράμματος (το αρχείο βίντεο : example.m4v).
- Το βίντεο μετατράπηκε από έγχρωμο σε βίντεο με αποχρώσεις του γκρι, (grayscale video), με τη βοήθεια βιβλιοθηκών της Python.
- Η αναζήτηση και η εύρεση του διανύσματος κίνησης, η οποία χρησιμοποιεί την συνάρτηση S.A.D. , θα γίνει με την βοήθεια της λογαριθμικής αναζήτησης.

3.ΥΛΟΠΟΙΗΣΗ

Ακολουθεί η ενότητα στην οποία θα αναλυθούν σε βάθος τα βήματα που ακολουθήθηκαν για την επίλυση της άσκησης 8.18 .

3.1.1. Εισαγωγή Βίντεο

Αρχικά, χάρις στη βιβλιοθήκη της γλώσσας Python (opencv-python) χρησιμοποιήσαμε τη μέθοδο cv2.VideoCapture, ώστε να διαβάσει το πρόγραμμα μας το βίντεο. Αυτό όμως απαιτεί την εγκατάσταση της βιβλιοθήκης opencv-python όπως αναφέρεται στην ενότητα 6 που έπεται.

3.1.2. Κωδικοποίηση Βίντεο

Στη συνέχεια της υλοποίησης, ακολουθεί η διαδικασία της κωδικοποίηση του βίντεο, για το οποίο πρέπει να επιστέψουμε τα πλαίσια σφαλμάτων. Ειδικότερα, το παραπάνω γίνεται εφικτό χρησιμοποιώντας την μέθοδο cv2.VideoWriter_fourcc() της βιβλιοθήκης cv2 (opencv-python), και το τρόπο κωδικοποίησης 'XVID'.

3.1.3. Εγγραφή του Βίντεο

Στο επόμενο βήμα της υλοποίησης μας, χρησιμοποιούμε τη μέθοδο cv2.VideoWriter (αντίστοιχα της βιβλιοθήκης cv2 (opencv-python)). Με το τρόπο αυτό, το πρόγραμμα μας καταφέρνει να διαβάσει τα βασικά στοιχεία που χρειάζεται για την ολοκλήρωση του, τα οποία είναι:

- Το όνομα του αρχείου βίντεο (το οποίο θα δίνεται από το χρήστη κατά την έναρξη της εφαρμογής).
- Τη κωδικοποίηση 'XVID' (βλ. προηγούμενη ενότητα).
- Τα πλαίσια ανά δευτερόλεπτο (frames per second).
- Το ύψος και το πλάτος για κάθε πλαίσιο (μέγεθος πλαισίου – size of frame).

3.1.4. Διαχείριση του πλαισίου I

Λαμβάνοντας ως δεδομένο, ότι το πρώτο πλαίσιο θα είναι πάντα ένα πλαίσιο τύπου I, αφού αποθηκευτεί θα μετατρέπεται σε gray scale με την βοήθεια της συνάρτησης `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)` της βιβλιοθήκης cv2 (opencv-python).

3.1.5. Διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία

Η διαδικασία συμπλήρωσης ενός πλαισίου με μαύρα εικονοστοιχεία χρειάζεται ως ορίσματα το πλαίσιο (frame) και έναν ακέραιο αριθμό (πχ. 16), ο οποίος είναι υπεύθυνος για την συνθήκη συμπλήρωσης με μαύρα εικονοστοιχεία ένα πλαίσιο.

Θυμίζουμε ότι από την υπόθεση της άσκηση, ένα πλαίσιο του οποίου το ύψος και το πλάτος δεν είναι πολλαπλάσιο του 16, θα πρέπει να δέχεται τη συγκεκριμένη διαδικασία.

Ειδικότερα, για κάθε πλαίσιο θα γίνεται η διαίρεση του ύψους και του πλάτους με το 16. Έπειτα, το αποτέλεσμα πολλαπλασιάζεται με το 16 και η τιμή αυτή αφαιρείται από την αρχική τιμή του ύψους και του πλάτους αντίστοιχα (με τη βοήθεια της βιβλιοθήκης numpy).

Για τη κάθε τιμή που προκύπτει θα γίνεται η κατάλληλη τροποποίηση στο πίνακα των εικονοστοιχείων, γεμίζοντας τις επιπλέον θέσεις με μαύρα εικονοστοιχεία.

3.1.6. Διαδικασία διαχείρισης για τα πλαίσια P

Ειδικότερα, για τον υπολογισμό της διαφοράς δύο πλαισίων και τη προβολή των πλαισίων σφαλμάτων είναι αναγκαία συνθήκη να υπάρχουν διαθέσιμα πλαίσια P για διάβασμα από το σύστημα. Όσο, η συνθήκη είναι αληθής το τρέχων πλαίσιο θα είναι το επόμενο πλαίσιο P και η επαναληπτική διαδικασία θα συνεχίζεται (μέχρι να μην υπάρχει πλαίσιο P για διάβασμα).

Το πλαίσιο που έχει διαβαστεί, μετατρέπεται σε gray scale με την βοήθεια της μεθόδου `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`. Η διαφορά των δύο πλαισίων P, ορίζεται η διαφορά μεταξύ του τωρινού πλαισίου και του προηγούμενου πλαισίου και υπολογίζεται με τη βοήθεια της μεθόδου `np.subtract` (της βιβλιοθήκης numpy).

3.1.7. Διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ $n \times n$

Η διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ με διαστάσεις $n \times n$, υλοποιείται με τρόπο παρεμφερή με το δεύτερο ερώτημα της άσκησης 8-17 με την ειδοποιός διαφορά να βρίσκεται στο μέγεθος του μακρομπλόκ (block size).

Ειδικότερα, για τις ανάγκες της άσκησης, το εκτελέσιμο θα προτρέπει το χρήστη, μέσω σχετικής ερώτησης, να ορίσει το μέγεθος του μπλοκ με απώτερο σκοπό να μπορεί ο κάθε χρήστης να συγκρίνει τον τρόπο λειτουργίας και την αποδοτικότητα του αλγορίθμου που υλοποιήθηκε.

Η διαδικασία διαίρεσης των πλαισίων, γίνεται μέσω μιας δομής επανάληψης for (for loop statement), χωρίς τη χρήση κάποιας “εξωτερικής” συνάρτησης.

Ειδικότερα, κάθε γραμμή και στήλη του μακρομπλόκ ορίζεται ως ένας πίνακας της μορφής: `[i : i + block_size, j : j + block_size]`, με το `block_size` να είναι ένας ακέραιος αριθμός που θα εισάγεται από το χρήστη στη σχετική ερώτηση του προγράμματος.

Πιο αναλυτικά, σε κάθε επανάληψη ο πίνακας του μακρομπλόκ είναι προγραμματισμένος να έχει πλήθος γραμμών από την τιμή του `i` έως και την τιμή του `i + block_size`, ενώ ομοίως το πλήθος στηλών από την τιμή του `j` έως και την τιμή του `j + block_size`.

Με αυτό το τρόπο επιτυγχάνεται η διαδικασία διαίρεσης των P πλαισίων σε μακρομπλόκ διαστάσεων $n \times n$, με το n να είναι ένας ακέραιος αριθμός ορισμένος από το χρήστη.

3.1.8 Διαδικασία υπολογισμού διανυσμάτων κίνησης και λογαριθμική αναζήτηση καλύτερου διανύσματος

Για την επιτυχή επίλυση του ζητούμενου προβλήματος, αναγκαία κρίνεται η διαδικασία του υπολογισμού των διανυσμάτων κίνησης, καθώς και η εφαρμογή της λογαριθμικής αναζήτησης για την εύρεση του καλύτερου διανύσματος.

Αυτό επιτυγχάνεται με την υλοποίηση της – εξωτερικής του εκτελέσιμου (θα βρίσκεται στο φάκελο *functions.py* όπως και όλες οι δημιουργημένες συναρτήσεις) - συνάρτησης compute motion vector.

Η περιοχή αναζήτησης της συνάρτησης καθορίζεται από την παράμετρο αναζήτησης K , η οποία είναι ίση με ον ακέραιο αριθμό 16 ($K = 16$). Στόχος της διαδικασίας εύρεσης του διανύσματος κίνησης είναι να βρεθεί η περιοχή εκείνη του πλαισίου αναφοράς ταιριάζει καλύτερα με το μακρομπλόκ στόχο (να ταιριάζει καλύτερα ως προς όλα τα υποψήφια τμήματα στην περιοχή της κάθε αναζήτησης).

Ο τύπος αναζήτησης του καλύτερου διανύσματος, που χρησιμοποιήθηκε στην επίλυση της συγκεκριμένης άσκησης είναι η λογαριθμική αναζήτηση, καθώς στη ζητούμενη υπόθεση της άσκησης οι υπολογισμοί θα πρέπει να είναι όσο το δυνατόν πιο γρήγοροι, διότι επηρεάζουν σημαντικά στην ολότητα της την παρούσα υλοποίηση.

Ως εκ τούτου, το παραπάνω υλοποιείται ως εξής: η μετρική υπολογίζεται επαναληπτικά σε εννέα διακριτές θέσεις, οι οποίες καθορίζονται από το K

$((K/2, K/2), (0, K/2), (-K/2, K/2), (-K/2, 0), (-K/2, -K/2), (0, -K/2), (K/2, -K/2), (K/2, 0))$.

Εναρκτήριο βήμα είναι ο υπολογισμός της μετρικής για το κέντρο της περιοχής και στη συνέχεια για τις υπόλοιπες περιοχές που προαναφέρονται. Στην επόμενη επανάληψη, το κέντρο της περιοχής αναζήτησης μετακινείται στην

καλύτερη θέση που υπολογίστηκε στο προηγούμενο βήμα, ενώ το βήμα αναζήτησης μειώνεται στο μισό.

Η παραπάνω δομή επανάληψης συνεχίζεται μέχρις ότου το μέγεθος του βήματος να γίνει ίσο με 1. Σε αυτή τη περίπτωση, από το καλύτερο ταίριασμα της τελευταίας επανάληψης αναδεικνύεται το καλύτερο διάνυσμα κίνησης.

3.1.9 Διαδικασία υπολογισμού της μετρικής SAD

Στη συνέχεια της υλοποίησής μας, δημιουργήσαμε την - εξωτερική - συνάρτηση sad (Sum of Absolute Differences), η οποία υπολογίζει το άθροισμα της απόλυτης τιμής των διαφορών μεταξύ δύο μακροπλόκ.

Η συγκεκριμένη συνάρτηση δημιουργείται και χρησιμοποιείται αποκλειστικά από το αρχείο συναρτήσεων `function.py` και στην συνάρτηση εύρεσης του διανύσματος κίνησης, `compute_motion_vector`.

3.1.10 Διαδικασία εύρεση κίνησης αντικειμένου

Εφόσον έχουμε υπολογίσει επιτυχώς τα διανύσματα κίνησης, είμαστε σε θέση να μπορούμε να “καταλάβουμε” τη κίνηση κάποιου αντικειμένου ή προσώπου στο προσκήνιο.

Το να “καταλάβουμε” που υπάρχει κίνηση στο προσκήνιο της ροής του βίντεο, αποτελεί σημείο καμπής της υλοποίησής μας, τη καίρια σημασία του θα τη δούμε αναλυτικά στην επόμενη ενότητα.

Η παραπάνω διαδικασία υλοποιείται μέσω της συνθήκης ελέγχου – εύρεσης της κίνησης που ακολουθεί:

$$motion_vector[0] + motion_vector[1] \neq 0 \quad [Συνάρτηση A]$$

Όταν, η παραπάνω συνθήκη είναι αληθής υπάρχει κίνηση στα μακροπλόκ του προσκηνίου – αντικειμένου.

3.1.11 Αντικατάσταση μακρομπλόκ προσκηνίου - παρασκηνίου

Όπως προαναφέρθηκε η εύρεση των μακρομπλόκ κίνησης του αντικείμενου στο προσκήνιο είναι πολύ σημαντικό για τη παρούσα άσκηση, καθώς μόνο με αυτό το τρόπο θα εντοπίσουμε τα μακρομπλόκ που έχουν μη μηδενικά διανύσματα κίνησης, τα οποία πρέπει να αντικατασταθούν με τα αντίστοιχα μακρομπλόκ του παρασκηνίου (τα αντίστοιχα μακρομπλόκ του αρχικού frame).

Με βάση την συνθήκη που ορίσαμε στη προηγούμενη ενότητα (συνθήκη εύρεσης κίνησης αντικείμενου – Συνάρτηση A) και όταν αυτή είναι αληθής (true) θα βρίσκουμε το μακρομπλόκ κίνησης, το οποίο θα αντικαθίσταται από το αντίστοιχο μπλοκ παρασκηνίου του προηγούμενου πλαισίου (πλαίσιο αναφοράς – reference frame).

3.1.12 Ανακατασκευή των πλαισίων

Δεδομένης της ανάγκης που έχουμε για να δημιουργήσουμε βίντεο με τα πλαίσια σφαλμάτων, θα πρέπει να ακολουθήσουμε την αντίστροφη διαδικασία της διαίρεσης των πλαισίων σε μακρομπλόκ.

Η διαδικασία αυτή θα μπορούσε να τιτλοδοτηθεί ως διαδικασία ενοποίησης όλων των $n \times n$ μακρομπλόκ (το n είναι ένας ακέραιος αριθμός ορισμένος από το χρήστη βλ. ενότητα 3.1.8), επιστρέφοντας με αυτό το τρόπο το αρχικό πλαίσιο, δηλαδή το πλαίσιο σφαλμάτων. Για την επιτυχή διαδικασία ανακατασκευής των πλαισίων χρησιμοποιείται η βιβλιοθήκη numpy.

3.1.13 Προβολή των πλαισίων σφαλμάτων

Με βάση τα προαναφερθέντα, και με απώτερο σκοπό τη προβολή των πλαισίων σφαλμάτων, επιλέχθηκε η δημιουργία ενός νέου εξαγόμενου βίντεο.

Αυτό επιτυγχάνεται με τον χωρισμό σε μακρομπλόκ διαστάσεων $n \times n$, ανά δύο πλαίσια, για τα οποία θα υπολογίζονται τα διανύσματα κίνησης και θα βρίσκεται το καλύτερο με βάση τη λογαριθμική αναζήτηση (βλ. ενότητα 3.1.8).

Στη συνέχεια, θα υλοποιείται ο υπολογισμός του μακρομπλόκ σφαλμάτων και για τα n μακρομπλόκ του κάθε πλαισίου, θα προκύπτει το πλαίσιο σφαλμάτων, το οποίο γράφεται στο νέο βίντεο και μετά το τέλος της επαναληπτικής διαδικασίας είτε θα αποθηκεύεται στον φάκελο του εκτελέσιμου αρχείου, είτε θα προβάλλεται από το εκτελέσιμο, στη περίπτωση που αυτό επιλεγεί από τον χρήστη στη σχετική προτροπή του προγράμματος.

Για την υλοποίηση της διαδικασίας αποθήκευσης του βίντεο χρησιμοποιήθηκε η μέθοδος `cv2.VideoWriter` (η οποία “γράφει” το αρχείο στο δίσκο), ενώ για την αναπαραγωγή αυτού η συνάρτηση `cv2.VideoCapture(filename)` (η οποία φορτώνει το ήδη εγγεγραμμένο αρχείο με βάση το όνομα που του είχε ορίσει νωρίτερα ο χρήστης).

4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτή την ενότητα θα παρατεθούν τα αποτελέσματα της παρούσας υλοποίησης για το ερώτημα της άσκησης 8-18. Ακολουθούν τα σχετικά αποτελέσματα:



Εικόνα 20 : Στιγμιότυπο του αρχικού βίντεο *example.m4v* .



Εικόνα 21 : Στιγμιότυπο του εξαγόμενου βίντεο με μέγεθος μακρομπλόκ 4 (*block_size_4.avi*).



Εικόνα 22 : Στιγμιότυπο του εξαγόμενου βίντεο με μέγεθος μακρομπλόκ 8 (*block_size_8.avi*).



Εικόνα 23 : Στιγμιότυπο του εξαγόμενου βίντεο με μέγεθος μακρομπλόκ 16 (*block_size_16.avi*).



Εικόνα 24 : Στιγμιότυπο του εξαγόμενου βίντεο με μέγεθος μακρομπλόκ 32 (block_size_32.avi).

5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτή την ενότητα θα παρουσιαστούν τα συμπεράσματα και οι παρατηρήσεις μας μετά την υλοποίηση – εκτέλεση της άσκησης 8.18. Ακολουθούν τα συμπεράσματα στα οποία καταλήξαμε:

- Με βάση τα παραγόμενα βίντεο της παρούσας εκτέλεσης παρατηρούμε ότι δεν επιτεύχθηκε η πλήρης απομάκρυνση του αντικειμένου. Ειδικότερα, κάποια εικονοστοιχεία του κινούμενου αντικειμένου στο προσκήνιο δεν αντικαταστάθηκαν από τα αντίστοιχα του παρασκήνιου. Πιθανολογούμε ότι το παραπάνω οφείλεται στις αλλαγές που υπάρχουν στο παρασκήνιο και στα επίπεδα φωτισμού (πχ. σκιά αντικειμένου).
- Η παρέμβαση της υλοποίησης μας στο τελικό εκτελέσιμο είναι φανερή, καθώς παρατηρείται μια σχετική θόλωση στα εικονοστοιχεία του προσκήνιου που τέθηκαν προς αντικατάσταση από τα αντίστοιχα του παρασκήνιου.
- Όσο μεγαλύτερες είναι οι τιμές της παραμέτρου για το χειρισμό του μεγέθους του μακρομπλόκ (`block_size`), τόσο το τελικό βίντεο είναι “χειρότερο”, καθώς το αντικείμενο που κινείται στο προσκήνιο διακρίνεται εν μέρη και η αντικατάσταση δεν είναι επιτυχής. Θα ακολουθήσουν σχετικές εικόνες.

6. ΣΚΕΨΕΙΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΗΣ ΥΛΟΠΟΙΗΣΗΣ

Συνδυαστικά με την παραπάνω ενότητα και των ζητημάτων – ελαττωμάτων που αναφέρθηκαν, στη παρούσα ενότητα θα αναφέρουμε κάποιους τρόπους και προτάσεις που πιθανολογούμε πως θα βελτιστοποιήσουν την υλοποίησή μας και θα αντιμετωπιστούν τα ζητήματα που παρατηρήθηκαν.

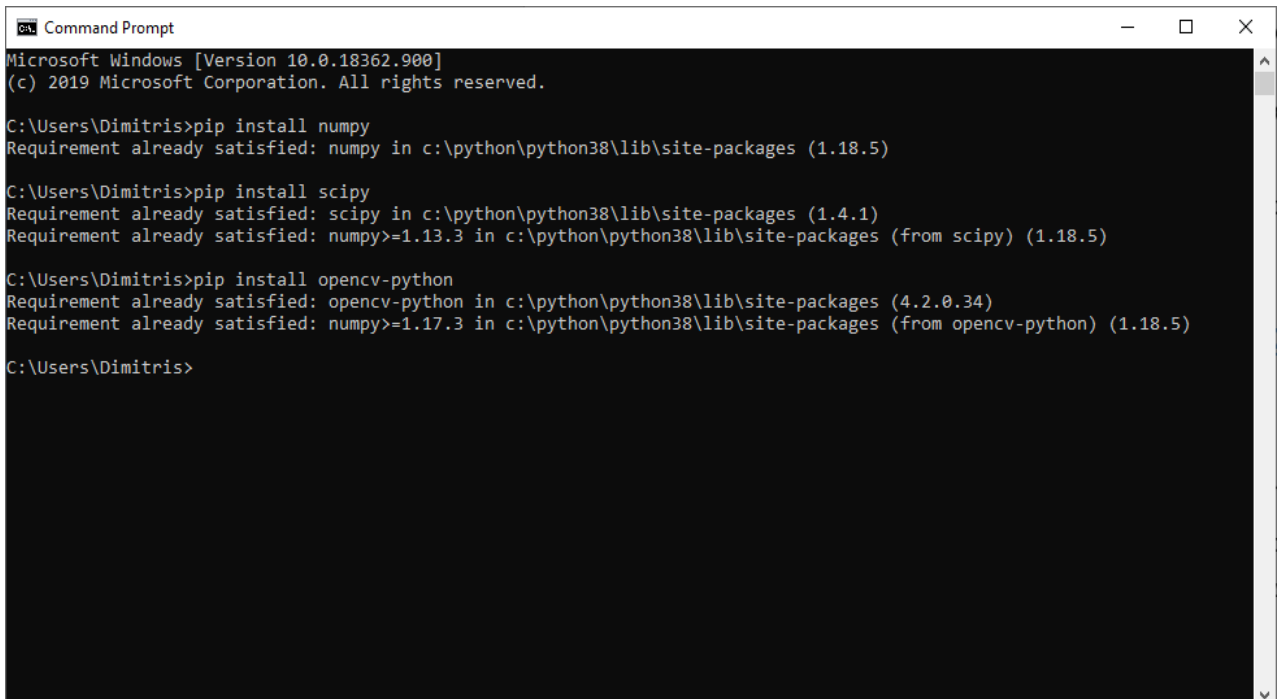
Ειδικότερα:

- Τα διαφορετικά επίπεδα φωτισμού (πχ. σκιά ανανά) και τα χαρακτηριστικά του παρασκήνιου (πχ. διαφορετικά χρώματα επιφανειών) επηρεάζουν μια διαδικασία τμηματικής πρόβλεψης με βάση την αντιστάθμιση κίνησης σε μια ροή βίντεο. Η επιλογή ενός “καλύτερου” βίντεο που θα πληρεί αυτά τα κριτήρια πιστεύουμε πως θα ωφελούσε την παρούσα υλοποίηση.
- Στο συγκεκριμένο βίντεο το αντικείμενο που κινείται (συγκεκριμένα ο ανανάς) δεν καλύπτει όλο το παρασκήνιο και θα μπορούσαμε να εφαρμόσουμε την αντικατάσταση μακρομπλόκ από το ίδιο πλαίσιο και όχι από το προηγούμενο. Μια τέτοια τεχνική πιθανολογούμε ότι θα βοηθούσε τη διαδικασία.

7. ΑΠΑΡΑΙΤΗΤΑ ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

Για την επίλυση της άσκησης 8-18 χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python** (έκδοση Python 3.8.3). Επιπρόσθετα, οι παρακάτω βιβλιοθήκες είναι απαραίτητες για την εκτέλεση του προγράμματος με βάση τις σχεδιαστικές προδιαγραφές του:

- **os (pathlib)** : για την εύρεση του βίντεο που επιθυμεί ο χρήστης.
- **numpy** : για το χειρισμό των υπολογιστικών δεδομένων (αριθμητικών) από τη Python.
- **time** : για την καταμέτρηση του χρόνου εκτέλεσης.
- **cv2 (opencv-python)** : για την εισαγωγή των βίντεο (εικόνων) και γενικότερα για το κομμάτι της υπολογιστικής όρασης (computer vision).



```
Command Prompt
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris>pip install numpy
Requirement already satisfied: numpy in c:\python\python38\lib\site-packages (1.18.5)

C:\Users\Dimitris>pip install scipy
Requirement already satisfied: scipy in c:\python\python38\lib\site-packages (1.4.1)
Requirement already satisfied: numpy>=1.13.3 in c:\python\python38\lib\site-packages (from scipy) (1.18.5)

C:\Users\Dimitris>pip install opencv-python
Requirement already satisfied: opencv-python in c:\python\python38\lib\site-packages (4.2.0.34)
Requirement already satisfied: numpy>=1.17.3 in c:\python\python38\lib\site-packages (from opencv-python) (1.18.5)

C:\Users\Dimitris>
```

Εικόνα 25 : Διαδικασία εγκατάστασης μερικών εκ των απαραίτητων βιβλιοθηκών.

8. ΠΕΡΙΕΧΟΜΕΝΑ ΑΡΧΕΙΟΥ ΕΠΙΛΥΣΗΣ ΤΗΣ ΆΣΚΗΣΗΣ

Το αρχείο της παρούσας άσκησης είναι το αρχείο 8-18, το οποίο θα περιέχει δύο ξεχωριστά αρχεία της γλώσσα προγραμματισμού Python:

- 8-18.py : Η επίλυση του ερωτήματος της άσκησης 8.18.
- functions.py : Το αρχείο που εμπεριέχει όλες τις συναρτήσεις που δημιουργήσαμε, η οποίες καλούνται από το αρχείο επίλυσης.

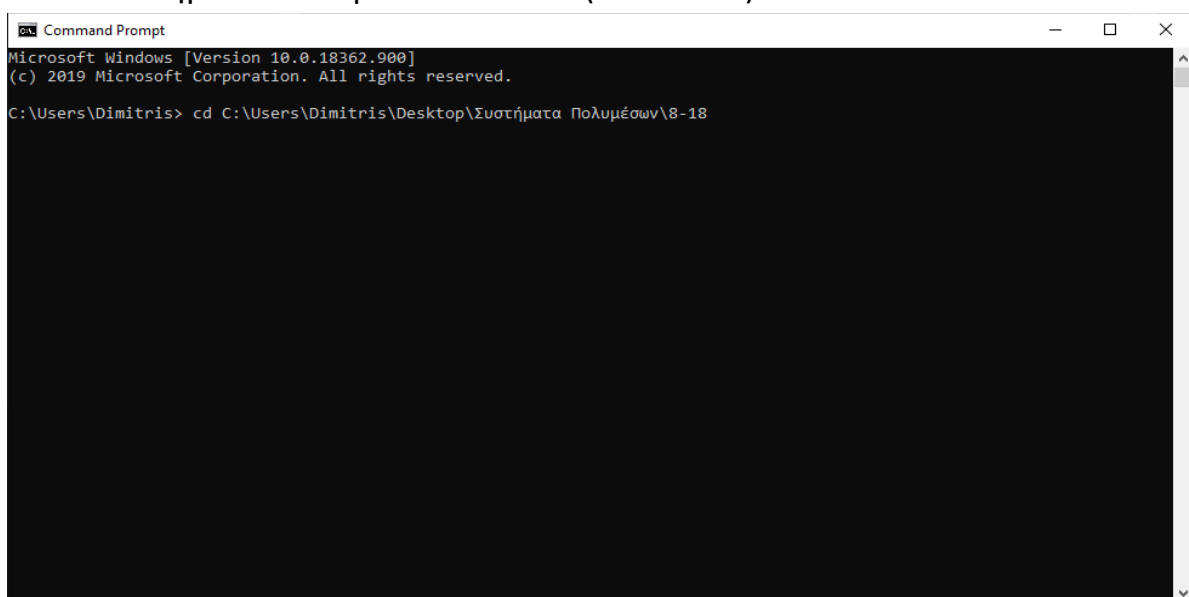
9. ΟΔΗΓΙΕΣ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Όπως προαναφέρθηκε παραπάνω το ερώτημα της άσκησης υλοποιείται σε ξεχωριστό αρχείο από τις δημιουργημένες συναρτήσεις. Το πρόγραμμα 8-18 λειτουργεί παρόμοια στο κομμάτι της αλληλεπίδρασης με το χρήστη (UI – User Interface) με τα ερωτήματα της προηγούμενης άσκησης 8-17.

Ειδικότερα, κατά την εκκίνηση του προγράμματος ο χρήστης μπορεί να επιλέξει πληκτρολογώντας το αντίστοιχο γράμμα τη λειτουργία που επιθυμεί.

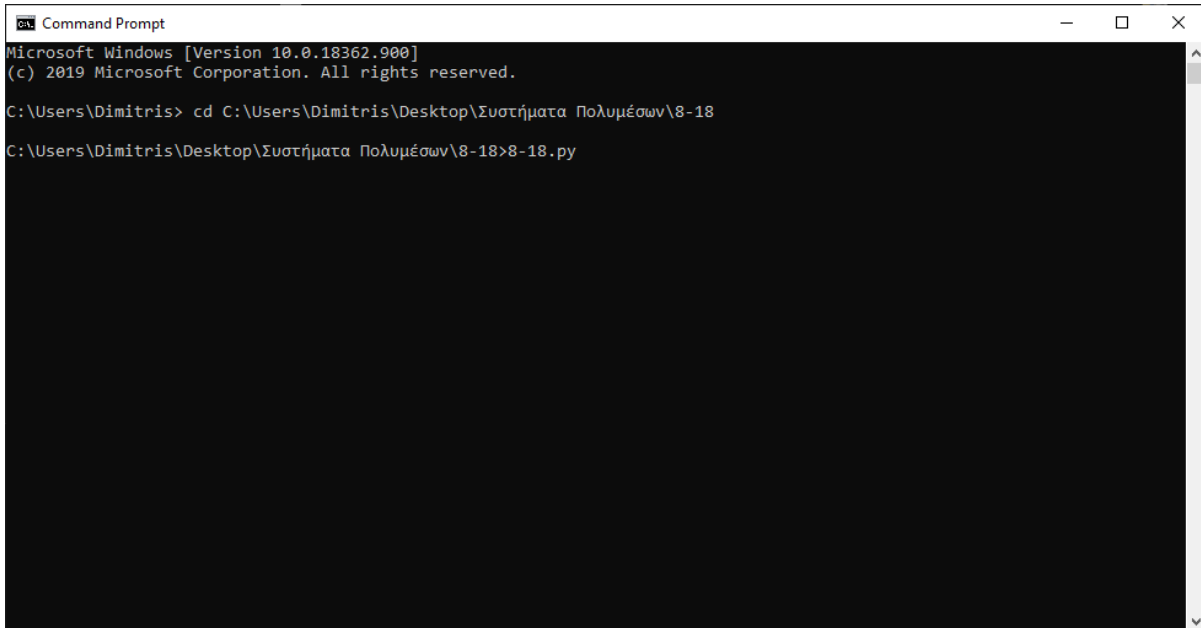
Πιο συγκεκριμένα, για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

- Μεταβαίνουμε στον φάκελο, όπου βρίσκεται το εκτελέσιμο:
Συστήματα Πολυμέσων → 8-18 (Εικόνα 26).



Εικόνα 26 : Μετάβαση στο φάκελο του εκτελέσιμου.

- Πληκτρολογούμε το όνομα του αρχείου 8-18.py, ώστε να ανοίξει το εκτελέσιμο (Εικόνα 27).

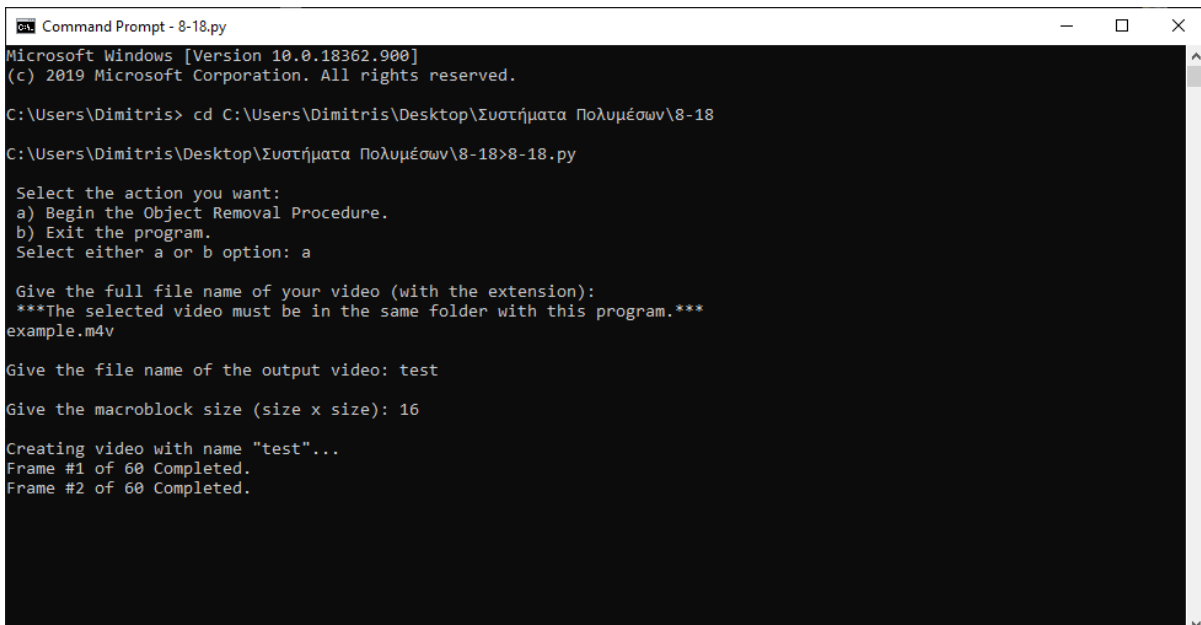


```
Command Prompt
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18>8-18.py
```

Εικόνα 27 : Επιλογή του εκτελέσιμου (εδώ 8-18.py).

Όταν ανοίξει το πρόγραμμα εμφανίζεται το μενού αλληλεπίδρασης του προγράμματος, όπου ο χρήστης καλείται να επιλέξει ποια λειτουργία επιθυμεί (Εικόνα 28).



```
Command Prompt - 8-18.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18>8-18.py

Select the action you want:
a) Begin the Object Removal Procedure.
b) Exit the program.
Select either a or b option: a

Give the full file name of your video (with the extension):
***The selected video must be in the same folder with this program.***
example.m4v

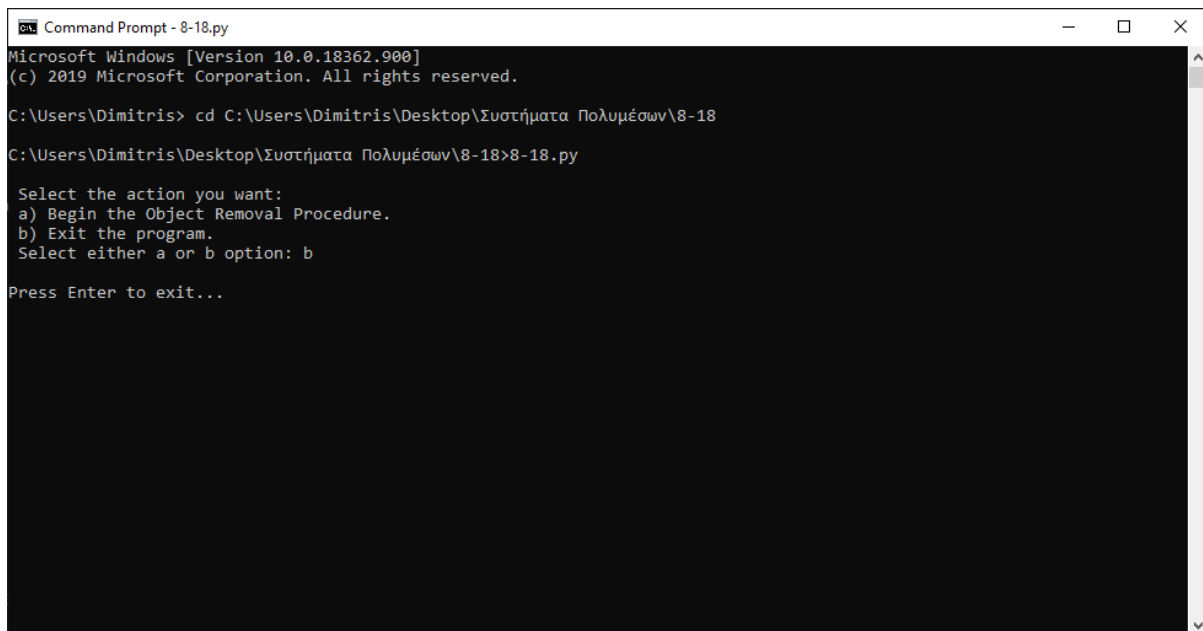
Give the file name of the output video: test

Give the macroblock size (size x size): 16

Creating video with name "test"...
Frame #1 of 60 Completed.
Frame #2 of 60 Completed.
```

Εικόνα 28: Αλληλεπίδραση με το μενού του εκτελέσιμου και έναρξη της διαδικασίας.

Στο εκτελέσιμο επίλυσης, εάν ο χρήστης επιθυμεί να τρέξει το πρόγραμμα τότε πληκτρολογεί το: a, ενώ πληκτρολογεί το: b, εάν την έξοδο από το πρόγραμμα (βλ. *Εικόνα 29*).



```
Command Prompt - 8-18.py
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Dimitris> cd C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18
C:\Users\Dimitris\Desktop\Συστήματα Πολυμέσων\8-18>8-18.py

Select the action you want:
a) Begin the Object Removal Procedure.
b) Exit the program.
Select either a or b option: b

Press Enter to exit...
```

Εικόνα 29 : Αλληλεπίδραση με το μενού του εκτελέσιμου και έξοδο από τη διαδικασία.

Στην περίπτωση που επιλεγθεί το a τότε για να δημιουργηθεί και να εξαχθεί το αντίστοιχο βίντεο πρέπει να ακολουθηθούν οι παρακάτω διαδικασίες:

- Ο χρήστης αρχικά, θα πρέπει να εισάγει το όνομα του αρχείου, συμπεριλαμβάνοντας την κατάληξη του, (το βίντεο που θα χρησιμοποιηθεί για την εκτέλεση της διαδικασίας π.χ.: *example.m4v*, βλ. *Εικόνα 28*).
- Εισάγει το όνομα του εξαγομένου αρχείου - χωρίς την κατάληξη – για να αποθηκευτεί το νέο βίντεο (π.χ. *test.avi*, βλ. *Εικόνα 28*).
- Εάν ο χρήστης επιθυμεί την προβολή του παραπάνω αρχείου, τότε στη σχετική προτροπή του εκάστοτε εκτελέσιμου θα πρέπει να πληκτρολογήσει “yes”, ενώ σε αντίθετη περίπτωση “no” (*Εικόνα 30*).


```
Command Prompt - 8-18.py
Frame #39 of 60 Completed.
Frame #40 of 60 Completed.
Frame #41 of 60 Completed.
Frame #42 of 60 Completed.
Frame #43 of 60 Completed.
Frame #44 of 60 Completed.
Frame #45 of 60 Completed.
Frame #46 of 60 Completed.
Frame #47 of 60 Completed.
Frame #48 of 60 Completed.
Frame #49 of 60 Completed.
Frame #50 of 60 Completed.
Frame #51 of 60 Completed.
Frame #52 of 60 Completed.
Frame #53 of 60 Completed.
Frame #54 of 60 Completed.
Frame #55 of 60 Completed.
Frame #56 of 60 Completed.
Frame #57 of 60 Completed.
Frame #58 of 60 Completed.
Frame #59 of 60 Completed.
Frame #60 of 60 Completed.

Success!
The output video, as test.avi, has been added in your program's folder.

Time elapsed: 308.84316658973694s

Do you want to play the video? (yes/no):
```

Εικόνα 30 : Ολοκλήρωση της διαδικασίας



Εικόνα 31 : Αποτέλεσμα προβολής του ερωτήματος.

Β' Μέρος

1. Εκφώνηση

- Έστω ομάδα N φοιτητών που έχει δηλωθεί στο μάθημα ($1 \leq N \leq 3$).
- Έστω ότι A_i οι αντίστοιχοι πενταψήφιοι αριθμοί μητρώου των φοιτητών αυτών.
- Κατασκευάστε συνθετική εικόνα διαστάσεων 104 γραμμών x 200 στηλών ως εξής: Κάθε γραμμή σχηματίζεται από πεντάδες αριθμών και κάθε πεντάδα είναι ένας A_i που επιλέγεται κάθε φορά τυχαία. Επίσης, σε κάθε πεντάδα, προτού αυτή ενσωματωθεί στην εικόνα, επιλέγεται τυχαία ένα ψηφίο και αντικαθίσταται με το 5.
- Αφού κατασκευάσετε την εικόνα, διαιρέστε τη σε macroblock 8x8 και σε κάθε macroblock εφαρμόστε τη διαδικασία που περιγράφεται στο Σχήμα 7-7 (σελ. 226) του βιβλίου.

Κεντρικό Ερώτημα: Ποιος είναι ο μέσος λόγος συμπίεσης της εικόνας που προκύπτει συνενώνοντας τα macroblocks των κβαντισμένων DCT συντελεστών (έστω εικόνα B) σε σχέση με την αρχική εικόνα (έστω εικόνα A), αν η διαδικασία κατασκευής της εικόνας A επαναληφθεί 100 φορές. Χρησιμοποιήστε μόνο κωδικοποίηση Huffman για τις εικόνες A και B και όχι άλλες μεθόδους όπως, π.χ., Κωδικοποίηση Μήκους Διαδρομής.

2. ΚΕΝΤΡΙΚΗ ΙΔΕΑ ΥΛΟΠΟΙΗΣΗΣ

Το κύριο ζητούμενο που πραγματεύεται η παρούσα υλοποίηση είναι η διαχείριση πενταψήφιων αριθμών (αριθμοί μητρώων φοιτητών).

Οι αριθμοί μητρώων ή αλλιώς πεντάδες αριθμών πρέπει κάθε φορά να διαχειρίζονται με τον κατάλληλο τρόπο, ώστε δημιουργείται η εικόνα 104 γραμμών και 200 στηλών που ζητείται.

Καταρχάς, για κάθε πεντάδα αριθμών που εισάγεται θα επιλέγεται τυχαία ένα ψηφίο και θα αντικαθίσταται από το ψηφίο 5.

Η διαδικασία της κατασκευής εικόνας θα γίνεται με επαναληπτικό τρόπο. Ειδικότερα, κάθε αριθμός που βρίσκεται στον πίνακα, θα επιλέγεται με τυχαίο τρόπο ώστε να συμπληρώνει συνεχόμενες θέσεις στον πίνακα της εικόνας. Η

διαδικασία επαναλαμβάνεται μέχρις ότου να δημιουργηθεί η απαιτούμενη εικόνα.

Τέλος, για την εύρεση του μέσου λόγου συμπίεσης, απαιτείται οι εικόνες να κωδικοποιηθούν με τον αλγόριθμο Huffman, θα αναλυθεί εκτενώς παρακάτω.

3. ΥΛΟΠΟΙΗΣΗ

Ακολουθεί η ενότητα στην οποία θα αναλυθούν εκτενώς τα βήματα που ακολουθήθηκαν για την επίλυση των ερωτημάτων της παραπάνω άσκησης.

3.1. Επιλογή πενταψήφιων αριθμών

Για την επίλυση της άσκησης αναγκαία κρίνεται η δημιουργία του πίνακα όπου θα αποθηκεύονται οι πενταψήφιοι αριθμοί μητρώου, οι οποίοι θα είναι από ένας μέχρι και τρεις και θα ορίζονται από το χρήστη κατά την εκκίνηση του προγράμματος.

3.2. Αντικατάσταση Ψηφίου με το 5

Βασική πρωταρχική διαδικασία για τη παρούσα υλοποίηση αποτελεί η διαδικασία κατά την οποία μετατρέπεται ένα τυχαίο ψηφίο και αντικαθίσταται με το 5.

Αυτό επιτυγχάνεται χάρις στην έτοιμη συνάρτηση που διαθέτει η `python` (γενικά υπάρχουν αντίστοιχες σε αρκετές γλώσσες προγραμματισμού) την `random.randint()`.

Με αυτό το τρόπο, για κάθε αριθμό που είναι αποθηκευμένος στον πίνακα θα επιλέγεται ένα τυχαίο ψηφίο, το οποίο και θα αντικαθίσταται από τον αριθμό 5.

3.3. Δημιουργία Εικόνας A

Στην υλοποίηση μας θεωρούμε πως η εικόνα A είναι η αρχική εικόνα, δηλαδή αυτή η οποία θα δημιουργείται με βάση τους πενταψήφιους αριθμούς μητρώου.

Η διαδικασία κατασκευής της θα είναι επαναληπτική (όπως προαναφέρθηκε). Σε αυτή την υποενοότητα θα αναλυθεί η διαδικασία για την πρώτη επανάληψη.

Αρχικά, με την βοήθεια της συνάρτησης `np.zeros` θα δημιουργηθεί ο πίνακας της εικόνας, ο οποίος θα έχει τις ζητούμενες διαστάσεις των 104 γραμμών x 200 στηλών και θα είναι γεμάτος – αρχικά - με μηδενικά.

Η διαδικασία συμπλήρωσης του πίνακα με τα ψηφία των αριθμών του πίνακα A είναι από μόνη της μια επαναληπτική διαδικασία.

Ακόμη, παρατηρούμε ότι το μήκος των αριθμών είναι 5 και ο αριθμός των στηλών 200, έτσι ο αριθμός των επαναλήψεων για το πλάτος της εικόνας θα είναι 40.

Τέλος, μετά την τυχαία επιλογή ενός αριθμού από τον αρχικό πίνακα αυτός τοποθετείται πέντε θέσεις μετά από το αντίστοιχο τελευταίο ψηφίο του προηγούμενου αριθμού κ.ο.κ.

3.4. Δημιουργία Εικόνας B

Στην υλοποίηση μας θεωρούμε πως η εικόνα B είναι η κατασκευασμένη εικόνα μετά από την ένωση των μακρομπλόκ των κβαντισμένων DCT συντελεστών.

Για την κατασκευή της εικόνας B ακολουθήθηκε η διαδικασία του βιβλίου (εικόνα 7-7 της σελίδας 226).

Αρχικά δημιουργήθηκε ένας πίνακας διαστάσεων διαστάσεις 104 γραμμών και 200 στηλών (η αρχικοποίηση του οποίου έγινε με την η συνάρτηση `np.zeros`)και θα είναι γεμάτος με μηδενικά, ωστόσο θα μπορεί να δέχεται και δεκαδικούς αριθμούς.

Έπειτα, η εικόνα B θα χωριστεί σε μακρομπλόκ 8 x 8, ώστε να πραγματοποιηθεί η διαδικασία για κάθε μακρομπλόκ.

Στη συνέχεια εισάγουμε τους DCT συντελεστές, όπως αυτοί προκύπτουν από την εικόνα A, οποία επίσης έχει χωριστεί σε μακρομπλόκ 8 x 8.

Για το επόμενο στάδιο απαραίτητη προϋπόθεση ήταν η ύπαρξη ενός πίνακα με τους συντελεστές κβάντισης (αντίστοιχος πίνακας του βιβλίου).

Έτσι, θα προκύψει η εισαγωγή των κβαντισμένων DCT συντελεστών στον πίνακα της εικόνας B.

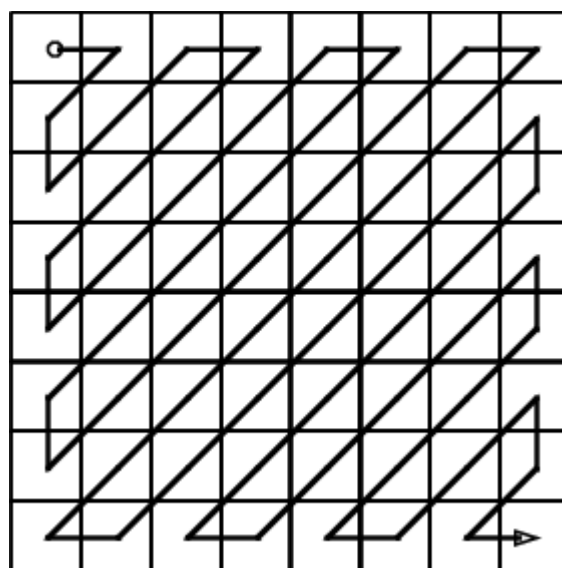
Δηλαδή, ο συγκεκριμένος πίνακας προκύπτει από την διαίρεση του πίνακα με τους DCT συντελεστές με τους αντίστοιχους συντελεστές κβάντισης, διατηρώντας το ακέραιο μέρος του αποτελέσματος.

3.5. Συνάρτηση Zigzag

Η συνάρτηση Zigzag χρησιμεύει στην εισαγωγή των μακρομπλόκ της εικόνας A, όσο και της εικόνας B σε έναν μονοδιάστατο πίνακα.

Αυτές οι τιμές στην συνέχεια θα μας χρησιμεύσει στην κωδικοποίηση Huffman (ακολουθεί η αντίστοιχη συνάρτηση).

Στην συγκεκριμένη συνάρτηση ελέγχεται η τωρινή θέση του πίνακα διότι παίζει σημαντικό ρόλο στην σειρά που θα εισαχθούν όπως φαίνεται και παρακάτω στην ενδεικτική φωτογραφία.



Εικόνα 32

3.6. Συνάρτηση Huffman encoding

Ο βασικός στόχος της άσκησης είναι η κωδικοποίηση Huffman. Σε αυτό είχε δοθεί και μεγάλη βαρύτητα στα εξ' αποστάσεως μαθήματα.

Η κωδικοποίηση αυτή υλοποιείται χάρις στην συνάρτηση `Huffman_encoding`. Η διαδικασία γίνεται πιο εύκολα επειδή προηγουμένως έχει υλοποιηθεί η συνάρτηση `ZigZag`. Αυτό συμβαίνει διότι η μέθοδος `Huffman.codebook (collections.Counter(zigzagged).items())` χρησιμοποιεί τα στοιχεία `macroblock` που προκύπτουν από την συνάρτηση `ZigZag`.

Εφόσον έχει γίνει η διαδικασία για όλα τα μακρομπλόκ, έπεται ως επακόλουθο την απόδοση ενός κωδικού για κάθε σύμβολο του μονοδιάστατου πίνακα ενώ το αποτέλεσμα της κωδικοποίησης θα είναι αποθηκευμένο στον πίνακα `output`.

Προφανώς ένα σύμβολο που εμφανίζεται συχνότερα από κάποιο άλλο θα έχει μεγαλύτερου μήκους κωδικού.

3.7. Υπολογισμός μέσου λόγου συμπίεσης

Τέλος, για τη διαδικασία του υπολογισμού και την αποθήκευση του μέσου λόγου συμπίεσης των δύο εικόνων, απαραίτητη είναι η καταμέτρηση των bits της κάθε εικόνας.

Τα αποτελέσματα θα αποθηκεύονται σε έναν πίνακα και ανανεώνονται κάθε φορά που μια εικόνα A και B υπόκειται στις προαναφερθέντες διαδικασίες.

Ο λόγος συμπίεσης είναι το αποτέλεσμα της διαίρεσης του πλήθους των bits της εικόνας A προς τα αντίστοιχο άθροισμα της εικόνας B.

Όμως, ο μέσος λόγος, είναι το ζητούμενο της άσκησης. Ο μέσος λόγος συμπίεσης υπολογίζεται ως το αποτέλεσμα της διαίρεσης μεταξύ του αθροίσματος των λόγων συμπίεσης προς το μήκος του, δηλαδή:

$$\text{sum(comp_ratios)/len(comp_ratios)}$$

4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτή την ενότητα θα παρατεθούν τα αποτελέσματα της παρούσας υλοποίησης.

Ακολουθούν τα αποτελέσματα μιας ενδεικτικής εκτέλεσης του προγράμματος για τους πενταψήφιους αριθμούς μητρώου των μελών της ομάδας μας: 17038 17068 και 17122.

```
C:\Users\Dimitris>cd Desktop  
  
C:\Users\Dimitris\Desktop>multimedia_part_b.py  
Please input the team number IDs in the format id_1 id_2 id_3. (with spaces in between)  
  
17038 17068 17122  
Number IDs are: ['17038', '17068', '17122']  
Number IDs after randomly changing a digit are: ['15038', '17568', '17152']  
Avg compression ratio: 1.8839480015063317
```

Εικόνα 33 : Αποτελέσματα του εκτελέσιμου.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Με βάση το αποτέλεσμα που προηγήθηκε, παρατηρούμε ότι ο μέσος λόγος συμπίεσης είναι 1.8839480015063317.

Το αποτέλεσμα αυτό μεταφράζεται ότι 100 Β εικόνες δημιουργήθηκαν με μέσο λόγο συμπίεσης τον προαναφερθέντα αριθμό και στα πρότυπα της η τεχνικής που περιγράφεται στη σελίδα 226 του βιβλίου.

Τέλος, παρατηρούμε ότι η παραπάνω διαδικασία είναι πιο αποτελεσματική από τη συμπίεση μιας εικόνας μόνο με την τεχνική Huffman.

6. ΑΠΑΡΑΙΤΗΤΑ ΕΡΓΑΛΕΙΑ ΓΙΑ ΤΗΝ ΕΚΤΕΛΕΣΗ

Για την επίλυση της άσκησης του Β' Μέρους χρησιμοποιήθηκε η γλώσσα προγραμματισμού **Python** (έκδοση Python 3.8.3). Επιπρόσθετα, οι παρακάτω βιβλιοθήκες είναι απαραίτητες για την εκτέλεση του προγράμματος με βάση τις σχεδιαστικές προδιαγραφές του:

- `numpy` : για το χειρισμό των υπολογιστικών δεδομένων (αριθμητικών) από τη Python.
- `random` : για την παρουσία ψευδοτυχαίων μεταβλητών (πχ. αντικατάσταση ψηφίου με το 5).
- `huffman` : για τη συμπίεση μιας εικόνας με την τεχνική Huffman.
- `collections` : για τη συλλογή των δεδομένων.
- `cv2` (`opencv-python`) : για την εισαγωγή των βίντεο (εικόνων) και γενικότερα για το κομμάτι της υπολογιστικής όρασης (`computer vision`).

```
C:\Users\Dimitris>pip install huffman
Collecting huffman
  Downloading huffman-0.1.2-py2.py3-none-any.whl (4.6 kB)
Installing collected packages: huffman
Successfully installed huffman-0.1.2
```

Εικόνα 34 : Διαδικασία εγκατάστασης μερικών εκ των απαραίτητων βιβλιοθηκών.

7. ΠΕΡΙΕΧΟΜΕΝΑ ΑΡΧΕΙΟΥ ΕΠΙΛΥΣΗΣ ΤΗΣ ΆΣΚΗΣΗΣ

Το αρχείο της παρούσας άσκησης είναι το αρχείο Β Μέρος το οποίο θα περιέχει:

- `multimedia_part_b.py` : Η επίλυση του ερωτήματος της άσκησης.

8. ΟΔΗΓΙΕΣ ΕΚΤΕΛΕΣΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

Κατά την εκκίνηση του προγράμματος ο χρήστης μπορεί να επιλέξει πληκτρολογώντας τον αντίστοιχο αριθμό μητρώου που επιθυμεί.

Πιο συγκεκριμένα, για να εκτελέσουμε τον κώδικα ακολουθούμε τα παρακάτω βήματα:

- Μεταβαίνουμε στον φάκελο, όπου βρίσκεται το εκτελέσιμο:
- Γράφουμε το όνομα του προς εκτέλεση αρχείου (multimedia_part_b.py).
- Επιλέγουμε από έναν μέχρι τρεις αριθμούς μητρώου (στη μορφή πενταψήφιων αριθμών με κενά μεταξύ τους αν είναι περισσότεροι από έναν).

Ακολουθεί ενδεικτική εικόνα:

```
C:\Users\Dimitris>cd Desktop
C:\Users\Dimitris\Desktop>multimedia_part_b.py
Please input the team number IDs in the format id_1 id_2 id_3. (with spaces in between)
17038 17068 17122
Number IDs are: ['17038', '17068', '17122']
Number IDs after randomly changing a digit are: ['15038', '17568', '17152']
Avg compression ratio: 1.8839480015063317
```

Εικόνα 35 : Αποτελέσματα Εκτέλεσης – Οδηγίες.

Βιβλιογραφία

1. «Συστήματα Πολυμέσων - Αλγόριθμοι, Πρότυπα και Εφαρμογές», Parag Havaladar, Gerard Medioni - Επιμέλεια Ελληνικής έκδοσης Άγγελος Πικράκης, Ph.D
2. Σημειώσεις μαθήματος «Συστήματα Πολυμέσων», Διδάσκων Άγγελος Πικράκης, Ph.D
3. Χρήσιμος Σύνδεσμος για τη μέθοδο Huffman (<https://pypi.org/project/huffman/>).

Περιεχόμενα Απεσταλμένου Αρχείου

Το τελικό αρχείο της εργασίας (Εργασία Συστήματα Πολυμέσων.zip) θα περιέχει τα παρακάτω:

1. Ένα φάκελο με τίτλο 8-17, ο οποίος θα περιέχει τα ακόλουθα:

- Τρία αρχεία Python (8-17a.py , 8-17b.py και functions.py).
- Το προτεινόμενο βίντεο για εκτέλεση συνδυαστικά με τα προγράμματα example.m4v .
- Το φάκελο Application's Screenshots, ο οποίος θα συμπεριλαμβάνει τα στιγμιότυπα εκτέλεσης που σχετίζονται με το ερώτημα 8-17.
- Το φάκελο Exported_Videos, ο οποίος θα συμπεριλαμβάνει τα εξαγόμενα βίντεο από τις δοκιμαστικές εκτελέσεις (test817a.avi και test817b.avi).
- Το αρχείο κειμένου 8-17_Results.txt, που εμπεριέχει, στατιστικά, τις επιδόσεις των εκτελέσιμων.

2. Ένα φάκελο με τίτλο 8-18, ο οποίος θα περιέχει τα παρακάτω:

- Δύο αρχεία Python (8-18.py και functions.py).
- Το προτεινόμενο βίντεο για εκτέλεση συνδυαστικά με τα προγράμματα example.m4v .
- Το φάκελο Application's Screenshots, ο οποίος θα συμπεριλαμβάνει τα στιγμιότυπα εκτέλεσης που σχετίζονται με το ερώτημα 8-18.

- Το φάκελο Exported_Videos, ο οποίος θα συμπεριλαμβάνει τα εξαγόμενα βίντεο από τις δοκιμαστικές εκτελέσεις (block_size_4.avi, block_size_8.avi, block_size_16.avi και block_size_32.avi).
- Το αρχείο κειμένου 8-18_Results.txt, που εμπεριέχει, στατιστικά, τις επιδόσεις του εκτελέσιμου για τις διαφορετικές δοκιμές στις οποίες υποβλήθηκε.

3. Ένα φάκελο με τίτλο Β Μέρος, ο οποίος θα περιέχει τα παρακάτω:

- Ένα εκτελέσιμο αρχείο Python (multimedia_part_b.py).
- Το φάκελο Application's Screenshots, ο οποίος θα συμπεριλαμβάνει τα στιγμιότυπα εκτέλεσης του ερωτήματος.

4. Το παρόν έγγραφο Εργασία Συστήματα Πολυμέσων.pdf, το οποίο περιλαμβάνει όλα όσα ζητήθηκαν από την εκφώνηση της εργασίας.
5. Ένα αρχείο κειμένου με τα ονόματα και τους Αριθμούς Μητρώου των μελών της ομάδας (Μέλη.txt).