



Εργασία-Συστήματα Πολυμέσων

Ονοματεπώνυμο : Σαλαμούρας Στέφανος

A.M. : Π16122

Περιεχόμενα

1. Θέμα 1	3
1.1 Εκφώνηση	3
1.2 Υλοποίηση	3
1.3 Αποτέλεσμα	5
2. Θέμα 2	7
2.1 Εκφώνηση	7
2.2 Υλοποίηση	7
2.3 Αποτέλεσμα	9

1. Θέμα 1

1.1 Εκφώνηση

Έστω ασυμπίεστο video της επιλογής σας, διάρκειας 5 s – 15 s.

Υποθέστε ότι το Frame 1 είναι πάντα I frame και ότι τα επόμενα πλαίσια είναι P frames.

i) Κάθε πλαίσιο P προβλέπεται χωρίς αντιστάθμιση κίνησης από το προηγούμενο πλαίσιο. Υπολογίστε και απεικονίστε την ακολουθία εικόνων σφάλματος και κωδικοποιήστε την χωρίς απώλειες. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

ii) Υλοποιήστε την τεχνική αντιστάθμισης κίνησης για την συμπίεση της ακολουθίας πλαισίων χρησιμοποιώντας αντιστάθμιση κίνησης σε macroblocks 64x64, ακτίνα αναζήτησης $k=32$ και τεχνική σύγκρισης macroblocks της επιλογής σας. Να επιταχυνθεί η διαδικασία υλοποιώντας ιεραρχική αναζήτηση. Υπολογίστε, αποθηκεύστε και απεικονίστε την ακολουθία εικόνων πρόβλεψης και εικόνων σφαλμάτων. Υλοποιήστε τον κωδικοποιητή/αποκωδικοποιητή.

1.2 Υλοποίηση

Για την επίλυση της άσκησης χρησιμοποιήθηκε η Python 3.7.0 και οι κώδικες για τα ερωτήματα i) και ii) υπάρχουν στα αρχεία **thema1_i.py** και **thema1_ii.py** αντίστοιχα, που βρίσκονται στο **source2024.zip**

Αναλυτικά ο κώδικας για το ερώτημα i) έχει ως εξής :

- Στις πρώτες γραμμές, εισάγουμε τις απαραίτητες βιβλιοθήκες(**Numpy**, **Cv2**). Η βιβλιοθήκη OpenCV χρησιμοποιείται για την εργασία με εικόνες και βίντεο στην Python.
- Στη συνέχεια, διαβάζουμε το ασυμπίεστο βίντεο από το αρχείο **video1.mp4**. Ορίζουμε μία μεταβλητή **cap** που αναπαριστά το βίντεο χρησιμοποιώντας τη συνάρτηση **cv2.VideoCapture()**.

- Ορίζουμε το πρώτο καρέ ως καρέ I (intra-frame). Διαβάζουμε το πρώτο καρέ και το μετατρέπουμε σε κλίμακα γκρι χρησιμοποιώντας την `cv2.cvtColor()`.
- Αρχικοποιούμε τον κωδικοποιητή για το αποθηκευμένο βίντεο. Εδώ ορίζουμε τη μορφή του αποθηκευμένου βίντεο σε MP4 χρησιμοποιώντας τον κωδικό H264 για την κωδικοποίηση, το όνομα του αποθηκευμένου αρχείου (**reconstructed_video1.mp4**), το πλαίσιο των FPS (καρέ ανά δευτερόλεπτο) και το μέγεθος του πλαισίου.
- Στη συνέχεια, ξεκινάει ένας βρόχος **while** για κάθε καρέ του βίντεο. Ελέγχουμε αν έχουμε επιτύχει την ανάγνωση ενός καρέ από το βίντεο.
- Μετατρέπουμε το τρέχον καρέ σε κλίμακα γκρι.
- Υπολογίζουμε το σφάλμα μεταξύ του τρέχοντος καρέ και του προηγούμενου. Το σφάλμα υπολογίζεται ως η διαφορά ανά pixel μεταξύ των δύο καρέ σε κλίμακα γκρι.
- Αποθηκεύουμε το πλαίσιο σφάλματος στο αποθηκευμένο βίντεο. Πρέπει να μετατρέψουμε το πλαίσιο σφάλματος σε αντικείμενο **BGR** πριν το αποθηκεύσουμε.
- Το τρέχον καρέ γίνεται το προηγούμενο για το επόμενο βήμα.
- Όταν ο βρόχος **while** τελειώσει, απελευθερώνουμε τους πόρους.

Αναλυτικά ο κώδικας για το ερώτημα ii) έχει ως εξής :

- **Φόρτωση του βίντεο:** Αυτό το τμήμα φορτώνει ένα βίντεο από ένα αρχείο (στη μεταβλητή **video_path**). Στη συνέχεια, χρησιμοποιεί τη βιβλιοθήκη OpenCV για να διαβάσει το βίντεο frame-by-frame. Τα καρέ του βίντεο αποθηκεύονται στη λίστα **frames**.
- **Υλοποίηση της αντιστάθμισης κίνησης:** Αυτή η συνάρτηση **motion_compensation** χρησιμοποιείται για την υλοποίηση της αντιστάθμισης κίνησης.
 - **Είσοδος:** Η συνάρτηση παίρνει μια λίστα από καρέ **frames** ως είσοδο, καθώς και ένα προαιρετικό όρισμα **k** που καθορίζει την ακτίνα αναζήτησης για την αντιστάθμιση κίνησης.

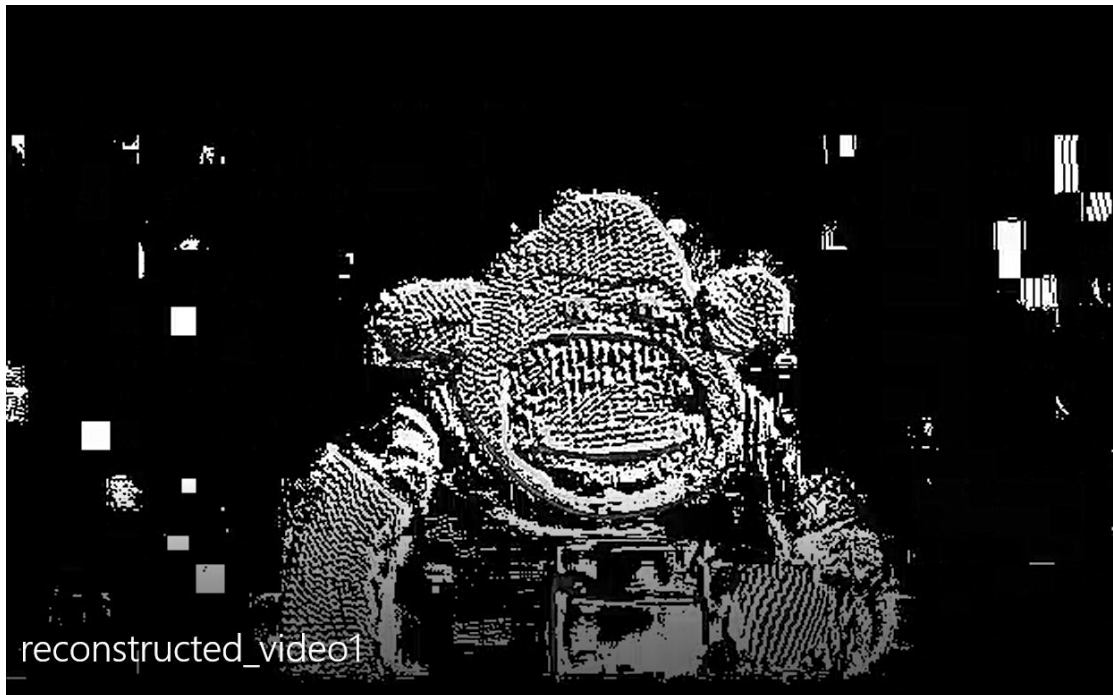
- **Λειτουργία:** Για κάθε καρέ εκτός από το πρώτο (το οποίο είναι I-frame), η συνάρτηση υπολογίζει τα προβλεπόμενα καρέ και τις αποκεντρώσεις. Για κάθε μπλοκ 64x64 του τρέχοντος καρέ, υπολογίζεται το καλύτερο κίνησης με τη χρήση του αθροίσματος απόλυτων διαφορών (SAD) μεταξύ του τρέχοντος μπλοκ και ενός παρόμοιου μπλοκ από το προηγούμενο καρέ. Το καλύτερο κίνησης χρησιμοποιείται για να προβλέψει το τρέχον μπλοκ, και έτσι παράγεται το προβλεπόμενο καρέ και το απόκεντρο. Το προβλεπόμενο καρέ προστίθεται στο προηγούμενο καρέ για να παραχθεί το προβλεπόμενο καρέ για το τρέχον καρέ.
- **Απεικόνιση των προβλεπόμενων καρέ και των αποκεντρώσεων:**
 - **Είσοδος:** Η συνάρτηση παίρνει ως είσοδο τις λίστες **frames** και **residuals**, που περιέχουν αντίστοιχα τα προβλεπόμενα καρέ και τις αποκεντρώσεις για κάθε καρέ του βίντεο.
 - **Λειτουργία:** Η συνάρτηση δημιουργεί δύο υπο-παραθέσεις στο γραφικό παράθυρο. Σε κάθε υπο-παραθέση, προστίθεται μια εικόνα για κάθε καρέ ή αποκέντρωση, αντίστοιχα. Η απεικόνιση γίνεται χρησιμοποιώντας τη βιβλιοθήκη Matplotlib. Οι προβλεπόμενες εικόνες εμφανίζονται σε έγχρωμη μορφή, ενώ οι αποκεντρώσεις εμφανίζονται σε κλίμακα του γκρι.

1.3 Αποτέλεσμα

Εκτελώντας τον κώδικα του πρώτου ερωτήματος βλέπουμε πως από το αρχικό βίντεο **video1.mp4**(που βρίσκεται στο **auxiliary2024.zip**) παίρνουμε το παραγόμενο βίντεο **reconstructed_video1.mp4** όπως φαίνεται παρακάτω :



video1.mp4



reconstructed_video1.mp4

2. Θέμα 2

2.1 Εκφώνηση

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

2.2 Υλοποίηση

Για την επίλυση της άσκησης χρησιμοποιήθηκε η Python 3.7.0 και ο κώδικας υπάρχει στο αρχείο **thema2.py** που βρίσκεται στο **source2024.zip**

Αναλυτικά ο κώδικας έχει ως εξής :

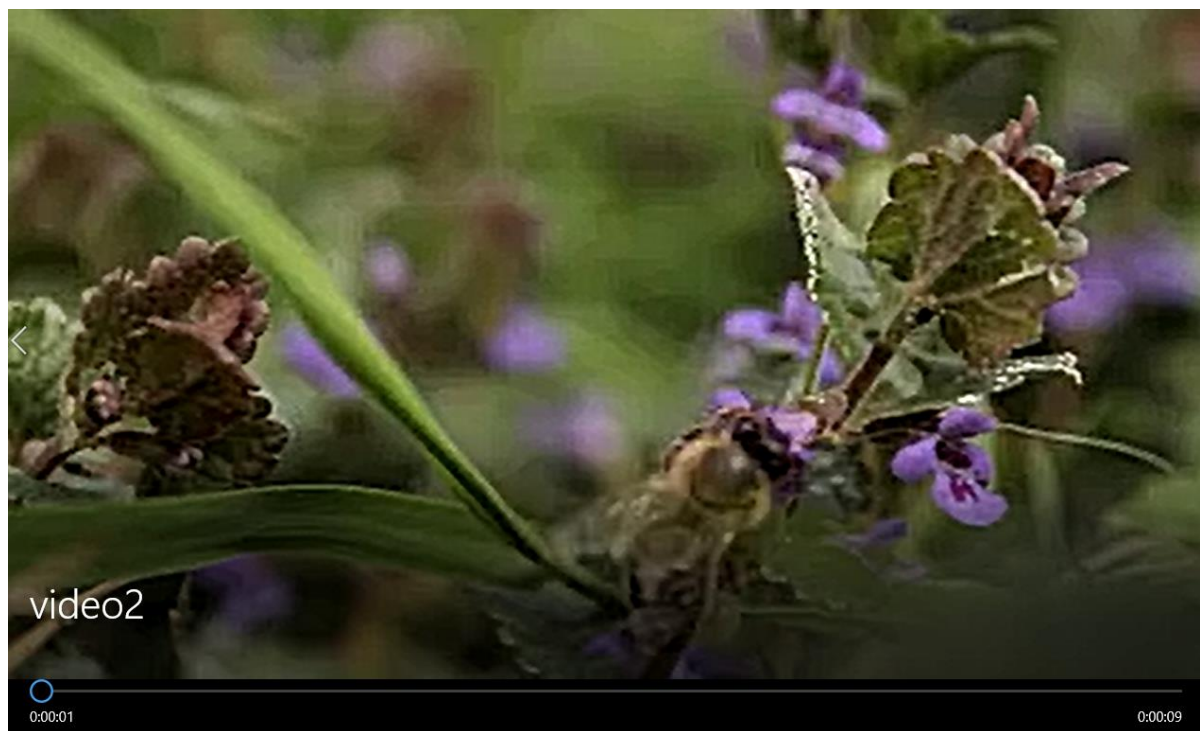
- **Εισαγωγή της βιβλιοθήκης OpenCV:** Η εντολή **import cv2** εισάγει αυτή τη βιβλιοθήκη στο πρόγραμμα.
- **Φόρτωση του αρχείου βίντεο:** Η συνάρτηση **cv2.VideoCapture('video2.mp4')** χρησιμοποιείται για να ανοίξει το αρχείο βίντεο που ορίζεται ως είσοδος.
- **Ανάγνωση και προετοιμασία του πρώτου καρέ:** Το πρώτο καρέ αναγνωρίζεται με τη χρήση της συνάρτησης **video_capture.read()**, και η ασπρόμαυρη και εικόνα σε χρωματικό χώρο HSV αποθηκεύονται στις μεταβλητές **previous_gray** και **hsv** αντίστοιχα.
- **Επιλογή περιοχής ενδιαφέροντος (ROI):** Η συνάρτηση **cv2.selectROI()** εμφανίζει το βίντεο και ζητά από τον χρήστη να επιλέξει την περιοχή του ενδιαφέροντος (ROI), όπου θα εξαφανιστεί το αντικείμενο.

- **Αρχικοποίηση του optical flow και των παραμέτρων εντοπισμού αντικειμένων:** Εδώ ορίζονται οι αρχικές τιμές για τον υπολογισμό του optical flow και η αρχική ιστόγραμμα για την περιοχή ενδιαφέροντος. Αυτές οι τιμές είναι απαραίτητες για την εφαρμογή της τεχνικής αντιστάθμισης κίνησης.
- **Προετοιμασία του βίντεο εξόδου:** Εδώ ορίζονται οι παράμετροι του βίντεο εξόδου, όπως το πλάτος, το ύψος και ο ρυθμός καρέ. Αυτές οι παράμετροι είναι απαραίτητες για τη δημιουργία του νέου βίντεο.
- **Επεξεργασία του βίντεο frame-by-frame:** Αυτός ο βρόχος χρησιμοποιείται για να διαβάσει κάθε frame του αρχικού βίντεο μία φορά. Η μεταβλητή `_` χρησιμοποιείται για να αγνοηθεί η επιστρεφόμενη τιμή (που είναι **True** αν το διάβασμα είναι επιτυχές ή **False** αν δεν υπάρχουν άλλα frames), ενώ η μεταβλητή `frame` περιέχει το προσεχές frame.
- **Μετατροπή του frame σε grayscale και υπολογισμός του optical flow:** Το προηγούμενο frame σε grayscale (`previous_gray`) συγκρίνεται με το τρέχον frame σε grayscale (`gray_frame`) για να υπολογιστεί η ροή του οπτικού ρεύματος (optical flow) με τη χρήση του αλγορίθμου Farneback.
- **Εφαρμογή της τεχνικής αντιστάθμισης κίνησης:** Εδώ εφαρμόζεται η αντιστάθμιση κίνησης στο frame. Αναλύουμε κάθε pixel του ROI (ορισμένο από τις συντεταγμένες `roi_x`, `roi_y`, `roi_w` και `roi_h`) και του εφαρμόζουμε την κίνηση που υπολογίστηκε. Αν οι νέες συντεταγμένες `new_y` και `new_x` είναι εντός των ορίων του frame, αντικαθιστούμε την τρέχουσα τιμή του pixel με αυτήν της νέας τοποθεσίας.
- **Εγγραφή του επεξεργασμένου καρέ στο βίντεο εξόδου:** Κάθε frame που έχει επεξεργαστεί και περιέχει την εξαφανισμένη περιοχή αποθηκεύεται στο βίντεο εξόδου.
- **Εμφάνιση του επεξεργασμένου καρέ:** Κάθε frame που έχει επεξεργαστεί και περιέχει την εξαφανισμένη περιοχή εμφανίζεται στην οθόνη για επαλήθευση.
- **Έλεγχος για διακοπή της διαδικασίας:** Ο βρόχος ελέγχει αν πατηθεί το κουμπί 'q' για να διακοπεί η εκτέλεση του προγράμματος.

- **Αποδέσμευση των πόρων:** Κατά την ολοκλήρωση, οι πόροι που χρησιμοποιήθηκαν για την ανάγνωση και την εγγραφή του βίντεο απελευθερώνονται.

2.3 Αποτέλεσμα

Εκτελώντας τον κώδικα βλέπουμε πως από το αρχικό βίντεο **video2.mp4** (που βρίσκεται στο **auxiliary2024.zip**) παίρνουμε το παραγόμενο βίντεο **output_video2.mp4** όπου έχουν εξαφανιστεί τα φύλλα στα αριστερά όπως φαίνεται παρακάτω :



video2.mp4



output_video2.mp4