



Πανεπιστήμιο Πειραιώς
University of Piraeus

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Απαλλακτική εργασία για το μάθημα Συστήματα Πολυμέσων.

Σεπτέμβριος 2020

Υλοποιήθηκε από τον φοιτητή

Βεργιάννη Νικόλαο - Π16170

Σημειώσεις

Η εργασία υλοποιήθηκε σε γλώσσα προγραμματισμού Python 3.7 χρησιμοποιώντας τις κατάλληλες βιβλιοθήκες και μεθόδους.

Στον φάκελο περιέχονται όλα τα αρχεία σε A μέρος και B μέρος καθώς και τα video που χρησιμοποιήθηκαν. Για την εκτέλεση τους ανοίγουμε την γραμμή εντολών μεταβαίνουμε στον αντίστοιχο φάκελο με την εντολή `cd ../#.py` και εκτελούμε το αντίστοιχο .py αρχείο.

Στο παρών έγγραφο αναλύεται το A μέρος της εργασίας. Για το B μέρος ανοίξτε τον αντίστοιχο φάκελο και το .pdf αρχείο.

Τα βίντεο που χρησιμοποιήθηκαν είναι σε χαμηλότερη ανάλυση για λόγους ευκολίας και χρόνου εκτέλεσης των διαδικασιών.

A Μέρος Εργασίας

Το Α μέρος της εργασίας περιλαμβάνει την υλοποίηση των προγραμματιστικών ασκήσεων 8.17,8.18

Ασκηση 8.17

Η επίλυση της άσκησης βρίσκεται στο φάκελο 8.17.

Θα χρησιμοποιήσουμε 2 μικρού μήκους βίντεο τα οποία θα επεξεργαστούμε.

Α μέρος 8.17

Πλαίσια σφαλμάτων

Το πρώτο αρχείο **error_frames.py** υλοποιεί τα πλαίσια σφάλματος μεταξύ των πλαισίων του βίντεο.

Φορτώνουμε με τη βοήθεια της βιβλιοθήκης `opencv-python` όλα τα πλαίσια του βίντεο, αποθηκεύοντας πάντα το προηγούμενο, εκτός από το πρώτο που είναι παντα τύπου I.

Η συνάρτηση `cv2.absdiff` μας δίνει την δυνατότητα να βρούμε τη διαφορά των δυο πλαισίων άμεσα.

Εκτελούμε τον κώδικα και παρατηρούμε το παράθυρο που δείχνει για όλα τα πλαίσια B τη διαφορά τους με το προηγούμενο. Μολις προβληθούν όλα τα πρόγραμμα τερματίζει.

Στον φάκελο υπάρχουν τα video μετά την εξαγωγή σε grayscale.

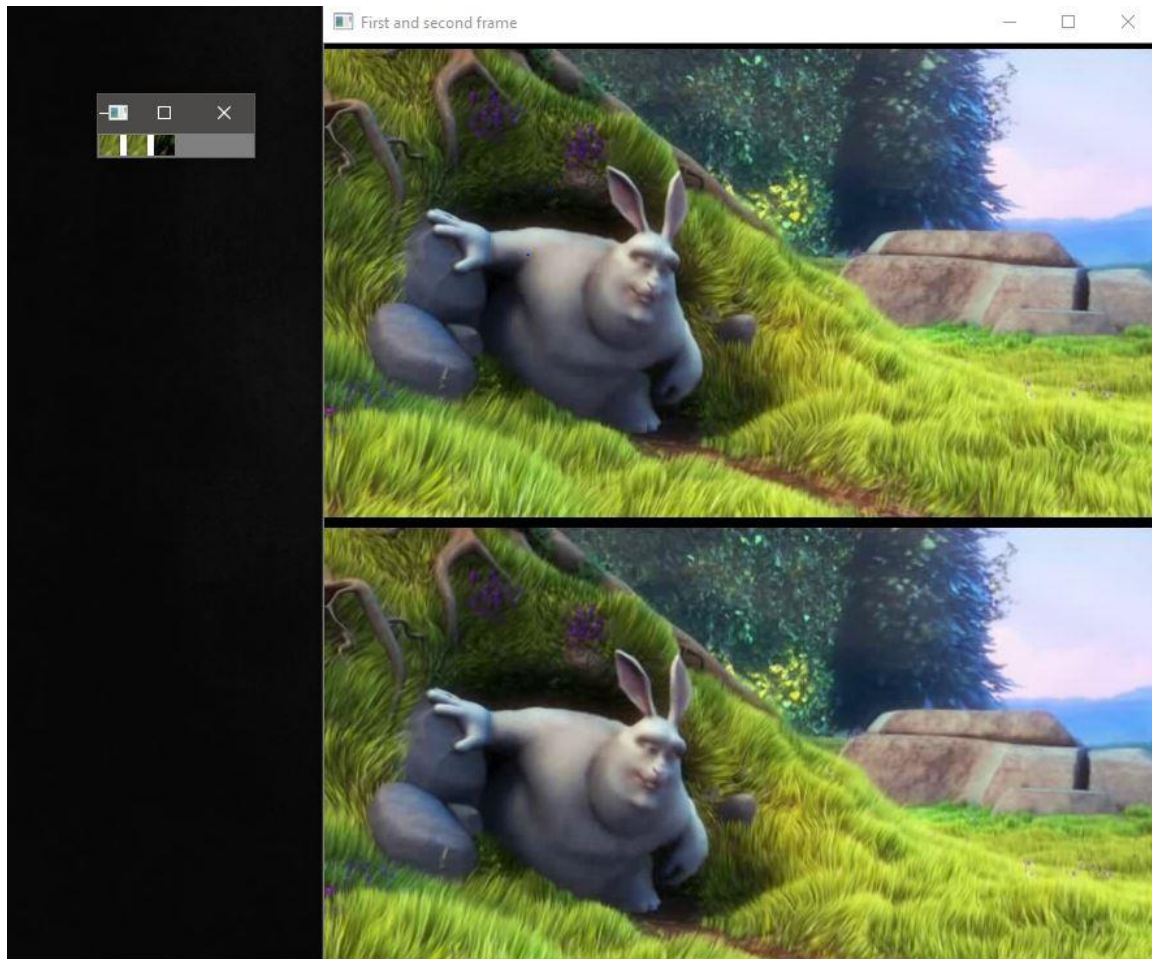
Β' μέρος 8.17

Καλύτερο ταίριασμα

Τρέχουμε το αρχείο `best_match.py`

Για την επίλυση είναι αναγκαίο να μεταβάλλουμε το πλάτος και το ύψος του βίντεο με τέτοιο τρόπο ώστε τα επιθυμητά macroblock να διαιρούνται με το 16. Θα χρησιμοποιήσουμε την μετρική SAD.

Το αρχείο `best_match.py` βρίσκει το καλύτερο ταίριασμα μεταξύ των πλαισίων του βίντεου που εισάγουμε. Εξάγωντας τα macroblocks επαναληπτικά κάθε φορά από το προηγούμενο και το επόμενο πλαίσιο και από την συνάρτηση `get_best_match()`



Εδώ βλέπουμε το καλύτερο ταίριασμα του βίντεο και στο μικρό παραθυράκι όλα τα πλαίσια στόχων, αναφοράς και σφάλματος που προέκυψαν αντίστοιχα. Μόλις βρει το καλύτερο macroblock της πρώτης εικόνας ο αλγόριθμος τερματίζει.

Πρόβλεψη Κίνησης

Αφού διαιρέσουμε τα επιθυμητά πλαίσια σε 16x16 macroblocks θα χρησιμοποιήσουμε την λογαριθμική συνάρτηση `get_motion_vector()` για να προβλέψουμε την κίνηση των αντικειμένων και να έχουμε όσο το δυνατόν πιο γρήγορους και αποτελεσματικούς υπολογισμούς. Κάθε φορά θα υπολογίζουμε την μετρική SAD του τρέχων πλαισίου

αναφοράς αλλά και των διακριτών γειτονικών πλαισίων στόχων ώστε να μεταφερθεί το κέντρο αναζήτησης στο καλύτερο. Η επανάληψη επαναλαμβάνεται κάθε φορά διαιρώντας δια 2 μέχρι να γίνει ίσο με 1 όπου βρίσκεται και το καλύτερο ταίριασμα, το καλύτερο διάνυσμα κίνησης δηλαδή και ο αλγόριθμος τερματίζει.

Μετα την ανακατασκευή των πλαισίων θα γράψουμε το αρχείο σε στο δίσκο σε .avi με την συνάρτηση `cv2.VideoWriter()`

Άσκηση 8.18

Στόχος και υλοποίηση

Στόχος της άσκησης είναι η απόκρυψη κάποιου κινούμενου αντικείμενου από την ροή ενός βίντεο. Η κίνηση του αντικείμενου εντοπίζεται από συνάρτηση η οποία εφαρμόζει τη τεχνική εύρεσης διανυσμάτων κίνησης και ουσιαστικά ανακατασκευάζει το τρέχον πλαίσιο με το καλύτερο γειτονικό του διάνυσμα από τον πίνακα αναφοράς και επιστρέφει νέο κέντρο αναζήτησης .

Το βίντεο μετατράπηκε σε αποχρώσεις του γκρι και η αναζήτηση και η εύρεση του διανύσματος κίνησης λογαριθμικά.

Χρησιμοποιούμε την βιβλιοθήκη `cv2` για να φορτώσουμε το αρχείο ήχου και την καταγραφή σε νέο αρχείο .avi με την μέθοδο `fourcc()` και κωδικοποίηση 'XVID'.

Το πρώτο πλαίσιο είναι πάντα τύπου I.

Θα γίνει η κατάλληλη τροποποίηση του πίνακα με συμπλήρωση των εικονοστοιχείων του τρέχοντως αλλά και του πλαισίου παρασκήνιου που δεν είναι πολλαπλάσιο του 16 με μαύρα εικονοστοιχεία από την συνάρτηση `fix_frame_shape()`.

Ειδικότερα η διαίρεση του ύψους και του πλάτους θα γίνεται υπολογίζοντας τον κοντινότερο αριθμό που είναι διαιρετός με το 16.

Μετρική S.A.D.

Η συνάρτηση `get_sad` υπολογίζει το άθροισμα της απόλυτης τιμής των διαφορών δυο macroblock. Είναι μια γρήγορη μετρική λόγω της απλότητας της και πολύ αποτελεσματική για μια ευρεία αναζήτηση πολλών διαφορετικών μπλοκ. Επίσης είναι εύκολα εφαρμόσιμο καθώς αναλύει κάθε εικονοστοιχείο ξεχωριστά.

Υπολογισμός Διανύσματος Κίνησης

Ενα διάνυσμα κίνησης είναι το βασικό στοιχείο στη διαδικασία εκτίμησης κίνησης. Χρησιμοποιείται για την αναπαράσταση ενός μακρομπλόκ σε μια εικόνα με βάση τη θέση αυτού του macroblock σε μια άλλη εικόνα, που ονομάζεται εικόνα αναφοράς.

Υπολογίζεται από την συνάρτηση `get_motion_vector()` η οποία παίρνει σαν είσοδο το τρέχων πλαίσιο της εικόνας αναφοράς σε πίνακα της μορφής `[i: i+μέγεθος macroblock, j:j+μέγεθος macroblock]`.

Το τρέχων αλλά και το πλαίσιο παρασκηνίου διορθώνεται από την μέθοδο `fix_frame_shape()` η οποία επιστρέφει το πλαίσιο

Η περιοχή αναζήτησης καθορίζεται από την μεταβλητή `κ = macroblock_size` όπου δημιουργεί διακριτούς γειτονικούς χώρους από το τρέχων πλαίσιο ώστε το κέντρο της περιοχής αναζήτησης να μετακινηθεί στην καλύτερη γειτονική θέση.

Η επανάληψη γίνει μέχρι το βήμα να γίνει 1 και να επιστραφεί το διάνυσμα που επιλέχθηκε

Ανακατασκευή πλαισίων

Για να γίνει η ανακατασκευή των πλαισίων που υπολογίστηκαν για αντικατάσταση θα πρέπει να καλύψουμε τα πλαίσια σφαλμάτων ενώνοντας τα 16x16 macroblocks σε πλαίσια βάση των macroblocks στον υπόχωρο x και τις σειρές με τα αντίστοιχα macroblocks στον υπόχωρο y .

Ακολουθούν στιγμιότυπα από ένα την αφαίρεση αντικειμένου σε 16, 8 και 4 macroblocks από ένα βιντεάκι 2 δευτερολέπτων 60 πλαισίων και ανάλυσης 1280x720 μιας μπάλας που πέφτει από ύψος .

- **16 macroblocks**



8 macroblocks



Χρόνος ανακατασκευής 60 πλαισίων του βίντεο 1579s

- 4 macroblocks



Χρόνος ανακατασκευής πλαισίων 2149s

Συμπεράσματα

Βλέποντας τα αποτελέσματα από το εξαγόμενο βίντεο ενός αμαξιού που κινείται σε ευθεία κατεύθυνση παρατηρούμε ότι υπάρχουν αρκετές παρενέργειες στο βίντεο και απότομη αλλαγή εικονοστοιχείων στο παρασκήνιο ειδικά προς το τέλος που η κάμερα είναι λιγότερο σταθερή.

Το ίδιο συμβαίνει και με την μπάλα καθώς πέφτει με αρκετή ταχύτητα. Το συγκεκριμένο βίντεο είναι με πιο σταθερή κάμερα. Παρατηρούμε ότι μειώνοντας το μέγεθος των macroblock και αυξάνοντας την ανάλυση πετυχαίνουμε καλύτερα αποτελέσματα τα οποία στο παρών παραδείγμα δεν είναι ευδιάκριτα. Επίσης η διαδικασία έχει μεγαλύτερο χρόνο εκτέλεσης.

Για να αντιμετωπίσουμε οποιαδήποτε παρενέργεια θα ήταν ίσως μια πιο έξυπνη αντικατάσταση macroblock από το ίδιο πλαίσιο και όχι προηγούμενου παίρνωντας μέρος από το ίδιο στιγμιότυπο εφόσον το αντικείμενο δεν καλύπτει όλο το παρασκήνιο.

Επίσης ο συνδυασμός ίδιων και προηγούμενων πλαισίων κάνει τον φωτισμό και άλλα χαρακτηριστικά του παρασκήνιου να μην είναι διακριτά σε ένα διάστημα πλαισίων της κίνησης του αντικειμένου.

Συναρτήσεις

Το αρχείο `functions.py` περιέχει χρήσιμες συναρτήσεις που χρησιμοποιούνται στον κώδικα των ασκήσεων.

Frame_to_macroblocks(frame, window=16)

Η συνάρτηση δέχεται ένα πλαίσιο και επιστρέφει πίνακα που περιέχει όλα τα macroblock. Η προεπιλεγμένη τιμή είναι 16.

- Τα macroblocks δημιουργούνται μεταβάλλοντας το πλάτος και το ύψος τους ώστε να διαιρείται σε 16x16 macroblocks.
- Εάν το πλαίσιο έχει μεγαλύτερη διάσταση από το αρχικό, συμπληρώνεται με μαύρα εικονοστοιχεία.
- Διατρέχουμε όλες τις γραμμές και στήλες του πλαισίου ανάλογα με το μέγεθος του macroblock
- Αποθηκεύουμε το macroblock στην εκάστοτε γραμμή και στήλη του πίνακα

Get_sad(previous_macroblock, next_macroblock) & get_sad_color()

Η συνάρτηση δέχεται 2 macroblock ενός πλαισίου αναφοράς (`previous_macroblock`) και ενός πλαισίου στόχου (`next_macroblock`). Επιστρέφει την μετρική τιμή SAD για αυτά τα δύο μακρομπλόκ. Το μέγεθος υπολογίζεται από το macroblock του πλαισίου αναφοράς.

Διατρέχουμε ανα εικονοστοιχεί τους πίνακες και προσθέτουμε την απολύτερη διαφορά ανα χρωματική συνιστώσα.

Επιστρέφουμε το αποτέλεσμα διαφορών.

Η `get_sad_color()` καλύπτει και τις 3 αποχρώσεις RGB (δεν υλοποιήθηκε στην εξαγωγή των αντικειμένων)

get_best_match(previous_motion, next_row, next_col, next_motion, k=16)

Λογαριθμική αναζήτηση για ένα macroblock που βρίσκεται στην θέση (`next_row`, `next_col`) βάση των macroblock του πλαισίου αναφοράς.

Τα βήματα αντιστοίχισης:

- Μας ενδιαφέρουν οι 8 γειτονικές διακριτές θέσεις γύρω από το macroblock του πλαισίου στόχου.
- Υπολογίζουμε τις μετρικές SAD τιμές τους.
- Νέο κέντρο αναζήτησης στο γειτονικό macroblock με το μικρότερο SAD(colored)

- Επαναλαμβάνουμε για $k = 16$ μειώνοντας $/2$ μέχρι να γίνει 1 και επιστρέφουμε το macroblock που επιλέχθηκε.

def get_motion_vector(macroblock, return_frame, coords, macroblock_size):

Η ίδια λογαριθμική αναζήτηση με την best_match η οποία λειτουργεί αναδρομικά ώστε να υπολογίσει την πρόβλεψη κίνησης και επιστρέφει το νέο κέντρο αναζήτησης. Χρησιμοποιείται για την εξαγωγή του αντικείμενου από το βίντεο.

Η συνάρτηση 'κοιτάζει' τις γειτονικές διακριτές θέσεις

Το Β' μέρος βρίσκεται σε pdf απο notebook στον αντίστοιχο φάκελο.

Βιβλιογραφία

<https://www.pexels.com/videos/>

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

