



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ

ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ

ΟΜΑΔΑ ΑΝΑΠΤΥΞΗΣ:

- **ΝΙΚΟΛΑΣ ΠΑΤΕΡΑΣ – Π17172**
- **ΑΝΤΡΕΑΣ ΘΕΟΔΩΡΙΔΗΣ – Π17164**
- **ΒΑΣΙΛΕΙΟΣ ΖΑΡΤΗΛΑΣ ΠΑΠΑΧΑΡΑΛΑΜΠΟΥΣ – Π17168**

Ιούλιος 2020

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	2
ΜΕΡΟΣ Α - ΚΕΦΑΛΑΙΟ 8.....	3
Άσκηση 8.17.....	3
Εκφώνηση.....	3
Υλοποίηση.....	4
Άσκηση 8.18.....	7
Εκφώνηση.....	7
Υλοποίηση.....	8
Αποτέλεσμα.....	10
ΜΕΡΟΣ Β.....	11
Εκφώνηση.....	11
Υλοποίηση.....	13
Αποτέλεσμα.....	15
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ.....	16
ΒΙΒΛΙΟΘΗΚΕΣ/ΕΡΓΑΛΕΙΑ.....	18

ΜΕΡΟΣ Α - ΚΕΦΑΛΑΙΟ 8

ΑΣΚΗΣΗ 8.17

Εκφώνηση

[8.17] 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 ιστότοπου www.cengagc.com. Εκεί θα βρείτε και κώδικα για την ανάγνωση και προβολή εικόνας. Τροποποιήστε 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 πλαίσιων**.

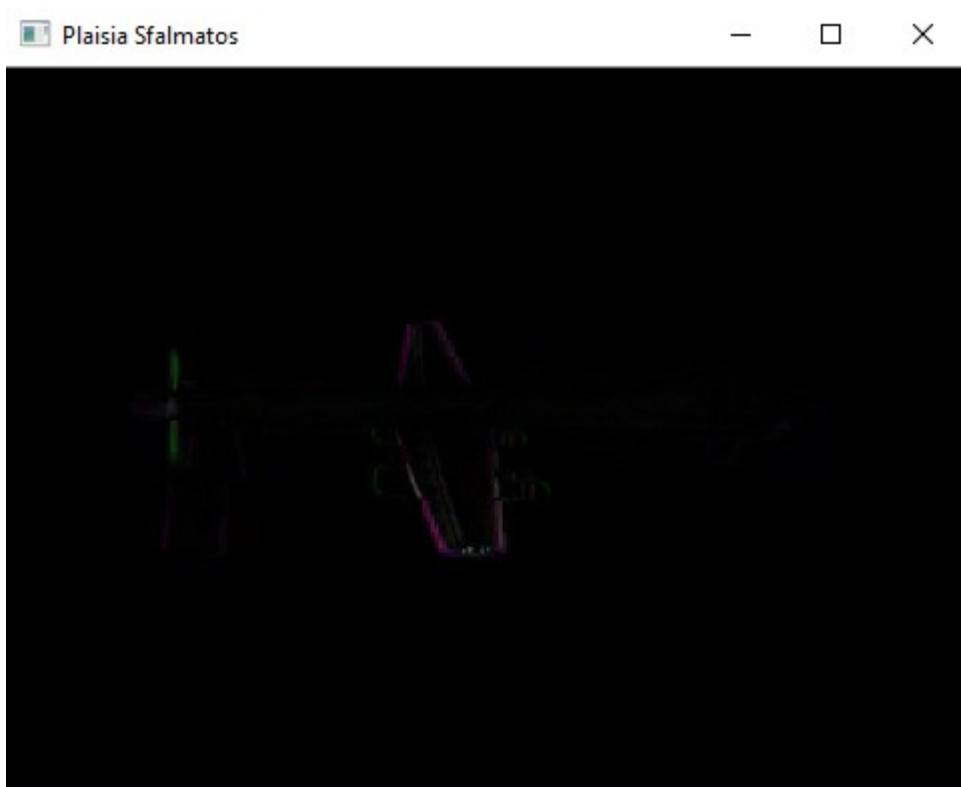
- Στο πρώτο μέρος της άσκησης αυτής, υποθέστε ότι θέλετε να προβλέπετε ολόκληρα **P** πλαίσια και όχι κατά τμήματα. Η πρόβλεψη κάθε ολόκληρου πλαισίου γίνεται με βάση το προηγούμενο πλαίσιο. Υλοποιήστε μία διαδικασία που δέχεται είσοδο δύο πλαίσια, υπολογίζει τη διαφορά τους και επιστρέφει ένα πλαίσιο 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.

- Στο δεύτερο βήμα θα υλοποιήσετε τεχνική πρόβλεψης κίνησης, η οποία υπολογίζει διανύσματα κίνησης ανά μπλοκ. Κάθε μπλοκ θα έχει το τυπικό **MPEG** μέγεθος **16 x 16**. Υλοποιήστε μια συνάρτηση που δέχεται είσοδο δύο πλαίσια: ένα πλαίσιο αναφοράς, το οποίο θα χρησιμοποιηθεί κατά την αναζήτηση των διανυσμάτων κίνησης, και ένα πλαίσιο στόχο, το οποίο θα προβλεφθεί. Διαιρέστε το πλαίσιο-στόχο σε μακρομπλόκ μεγέθους **16 x 16**. Εάν το πλάτος και ύψος του πλαισίου δεν είναι πολλαπλάσια 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 η διαδικασία ολοκληρωθεί για όλα τα μπλοκ, θα προκύψει ένα πλαίσιο σφαλμάτων. Να γίνει η προβολή όλων των πλαισίων σφαλμάτων. Θα διαπιστώσετε ότι τα πλαίσια

σφαλμάτων εμφανίζουν σημαντικά μικρότερη εντροπία σε σύγκριση με την προηγούμενη περίπτωση, μολονότι απαιτείται περισσότερος χρόνος για τον υπολογισμό τους.

Υλοποίηση

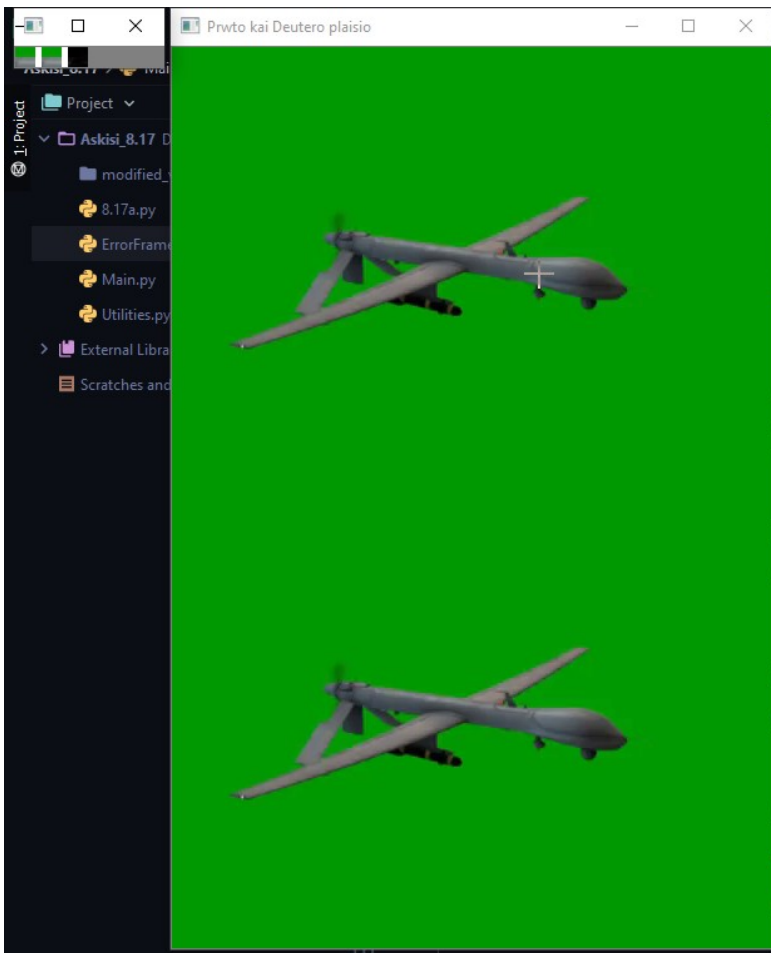
Αρχικά, στην κλάση **Main.py** εκτελούνται οι περισσότερες λειτουργίες αλλά η κλάση **Utilities.py** αποτελεί σκελετός για την τάξη **Main**. Χρησιμοποιούνται οι βιβλιοθήκες **opencv** & **numpy**. Στην τάξη **ErrorFrames** υλοποιούνται τα πλαίσια σφάλματος μεταξύ των πλαισίων. Εφόσον εκτελεστεί η κλάση, η βιβλιοθήκη **opencv** φροντίζει έτσι ώστε να έχουμε ένα **GUI** για την προβολή των βίντεο, όπως φαίνεται και στην **Εικόνα 1** πιο κάτω εμφανίζονται τα πλαίσια σφάλματος και όταν έχουν προβληθεί όλα το πρόγραμμα τερματίζει.



Εικόνα 1

Στην συνέχεια, στην τάξη Main, γίνεται πρόβλεψη της κίνησης και χρησιμοποιείται η μετρική **SAD** δηλαδή στα Αγγλικά Sum of Absolute Difference και γίνεται λογαριθμική αναζήτηση με βάση το προηγούμενο και

επόμενο **macroblock**. Στην **Εικόνα 2** γίνεται σύγκριση μεταξύ δυο πλαισίων, και πάλι το GUI είναι εφικτό με την βοήθεια του **opencv**. Επίσης, στο μικρό παράθυρο προβάλλει τα macroblocks που συγκρίνονται και η διαφορά τους. Το πρόγραμμα τερματίζει όταν ο γίνει η σύγκριση όλων των macroblock του δεύτερου πλαισίου με το καλύτερο του πρώτου.



Εικόνα 2

Γενικά, η βιβλιοθήκη **opencv** της **python** περιέχει πολλές χρήσιμες συναρτήσεις για την ευκολότερη υλοποίηση της

παρόντος προγραμματιστικής άσκησης, για παράδειγμα μπορούμε με μόνο μία λειτουργία ([cv.VideoCapture\(\)/Ορίσματα:1](#)) να φορτώσουμε όλα τα

πλαίσια του βίντεο και αποθηκεύοντας το προηγούμενο εκτός το πρώτο που είναι τύπου **I**. Εκτός από αυτό, η μέθοδος [cv.absdiff\(\)/2](#) μας επιτρέπει να βρούμε την απόλυτη διαφορά μεταξύ δύο πλαισίων. Επομένως, για το δεύτερο υποερώτημα παίρνουμε τα δύο πρώτα πλαίσια του βίντεο και τα χωρίζουμε σε macroblock **16 x 16**. Θα πρέπει να μεταβάλλουμε το πλάτος και ύψος του βίντεο έτσι ώστε να διαιρούνται με το **16**. Μετά γίνεται εφαρμογή της λογαριθμικής αναζήτησης για να βρούμε το επιθυμητό πλαίσιο αναφοράς με βάση την μετρική **SAD**. Αναλυτικότερα, εκτελώντας, nested loops για τα να βρούμε το κατάλληλο macroblock που ταιριάζει καλύτερα από το προηγούμενο, βρίσκουμε και το «padding» που χρησιμοποιείται για την τοποθέτηση περιθωρίων γύρω από το βίντεο. Στο τέλος οι συναρτήσεις [vid.release\(\)/0](#) και [cv.destroyAllWindows\(\)/0](#) κλείνουν το βίντεο και όλα τα ανοιχτά παράθυρα που δημιουργήσαμε αφού η διαδικασία έχει τελειώσει.

ΑΣΚΗΣΗ 8.18

Εκφώνηση

[8.18] 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 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 μακρομπλόκ.

- 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, η, 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 μηδενικά (ή σχεδόν μηδενικά) διανύσματα κίνησης, γιατί η κάμερα ήταν στατική. Για τα μπλοκ των αντικειμένων σε κίνηση, 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 υποθέσετε ότι η κάμερα δεν κινείται, ότι 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. Πώς μπορείτε να αξιοποιήσετε αυτό το γεγονός, πλέον της χρήσης αντιστάθμισης κίνησης σε επίπεδο μακρομπλόκ;

Υλοποίηση

Καταρχάς, με την ίδια δομή τάξεων έγινε και αυτή η προγραμματιστική άσκηση. Χρησιμοποιούνται οι βιβλιοθήκες **numpy** & **opencv**. Επίσης, χρησιμοποιήθηκε η τεχνική ιεραρχικής αναζήτησης. Θα μπορούσε να χρησιμοποιηθεί και η τεχνική λογαριθμικής αναζήτησης αλλά θα έπαιρνε πολύ περισσότερο χρόνο για κάθε πλαίσιο λόγω της ανάλυσης των βίντεο. Ξεκινάμε διαβάζοντας ένα-ένα τα πλαίσια του βίντεο ελέγχοντας στην αρχή αν η εικόνα μπορεί να διαιρεθεί σε μακρομπλόκ μεγέθους **16 x 16** και αν όχι γεμίζουμε τις υπολειπόμενες θέσεις του πίνακα τιμών του πλαισίου με μηδενικά, δηλαδή μαύρα **pixel** για να μπορεί να διαιρεθεί. Στη συνέχεια μετατρέπουμε τα πλαίσια σε ασπρόμαυρα για ευκολότερες πράξεις και αρχικοποιούμε τις αρχικές τους μορφές των εικόνων ως πρώτο επίπεδο της ιεραρχικής μας ανάλυσης. Το πρώτο πλαίσιο του βίντεο όπου θα είναι και το παρασκήνιο μας, αποθηκεύεται ως η πρώτη εικόνα.

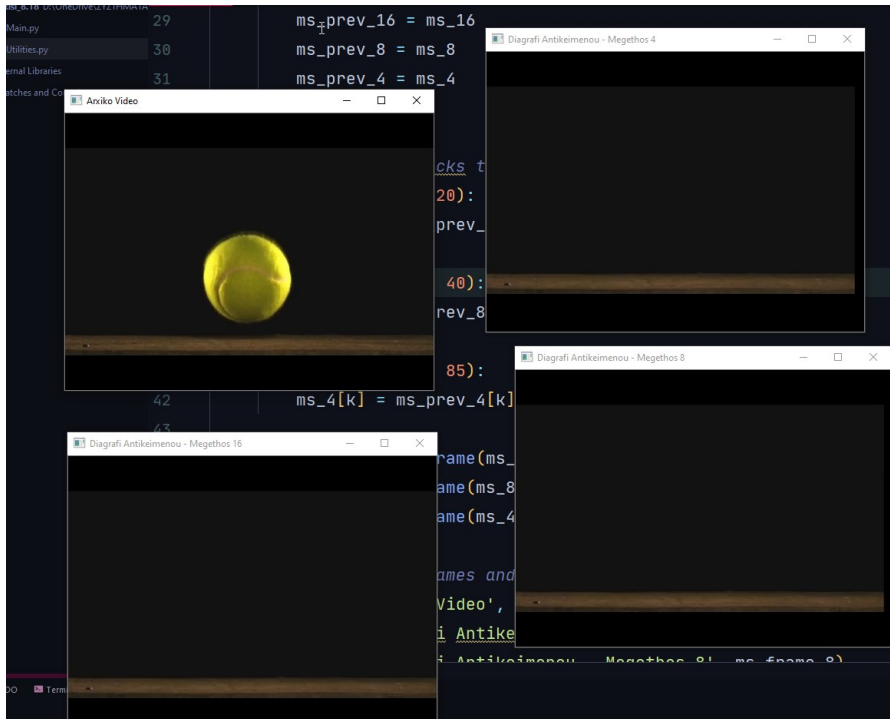
Συνεχίζουμε καλώντας τη συνάρτηση που κάνει ανάλυση των εικόνων σε επίπεδα. Δηλαδή η συνάρτηση, για κάθε δύο διαδοχικά πλαίσια, τα αναλύει σε επίπεδα οπού σε κάθε επίπεδο μειώνουμε στο μισό την ανάλυση, το μέγεθος των macroblock αλλά και την παράμετρο *m* που καθορίζει την περιοχή αναζήτησης. Ξεκινάμε με την εικόνα στη πλήρες μορφή της και μετά την χωρίζουμε σε macroblock **16 x 16** και την αποθηκεύουμε ως δεύτερο επίπεδο όπως ζητείται και μετά σε **8 x 8** κ.ο.κ μέχρι να φτάσουμε στο **4 x 4** που είναι το μικρότερο επίπεδο, και το υψηλότερο αντίστοιχα στην ιεραρχική αναζήτηση. Με αυτή την τεχνική επιτυγχάνουμε

πολύ γρήγορες πράξεις για την εύρεση των διανυσμάτων κίνησης και την αναζήτηση του καλύτερου ταιριαστού macroblock, καθώς τα macroblocks στο υψηλότερο επίπεδο έχουν μικρότερο μέγεθος και η περιοχή αναζήτησης είναι μικρότερη. Στη συνέχεια, γίνεται μια πλήρης αναζήτηση στο υψηλότερο επίπεδο και

βρίσκονται τα διανύσματα κίνησης, όπου τα υπολογίζουμε βρίσκοντας τη διαφορά ανάμεσα σε δυο macroblock στις αντίστοιχες θέσεις του κάθε macroblock και έπειτα για κάθε ένα από αυτά ελέγχουμε αν το ποσοστό των μηδενικών στο μακρομπλόκ διαφορών, είναι μεγαλύτερο του 90% δηλαδή δεν υπήρχε αξιοσήμαντη κίνηση. Αντιθέτως αν υπήρχε κίνηση αποθηκεύουμε τη θέση του μακρομπλόκ για να τη ξαναχρησιμοποιήσουμε στο κατώτερο επίπεδο.

Έπειτα, προχωράμε στο κατώτερο επίπεδο βελτιώνοντας στην ουσία τα μπλοκ και τα διανύσματα κίνησης που υπήρχε κίνηση στο υψηλότερο επίπεδο και τα ελέγχουμε ξανά στο χαμηλότερο επίπεδο αυτή τη φορά. Έχοντας υπολογίσει τα διανύσματα κίνησης για κάθε μακρομπλόκ αντικαθιστούμε σε κάθε macroblock στη θέση που εντοπίζεται κίνηση το ήδη υπάρχον μπλοκ του πλαισίου αναφοράς με το αντίστοιχο μακρομπλόκ του παρασκηνίου στην αντίστοιχη θέση. Τέλος καλούμε τη συνάρτηση ανακατασκευής της εικόνας που παίρνει όλα τα μακρομπλόκ **16 x 16** και τα επαναφέρει στο αρχικό μέγεθος «ενώνοντάς» τα, δίνοντας μας την πλήρης εικόνα στην ανάλυση που θέλουμε έχοντας πιά τα ολοκληρωμένα πλαίσια με την «εξαφάνιση» του αντικειμένου, όπου και το γράφουμε στο τελικό αρχείο βίντεο.

Αποτέλεσμα



Εικόνα 3

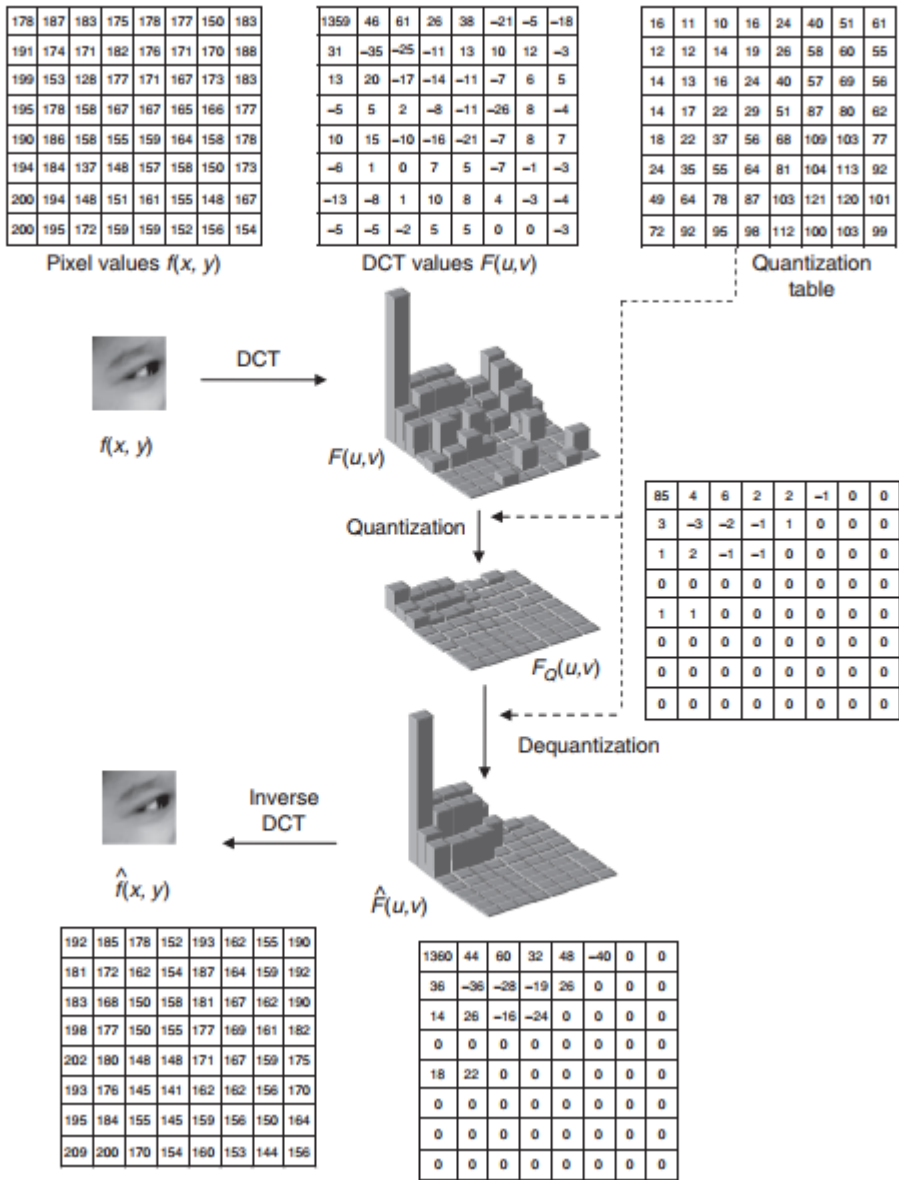
ΜΕΡΟΣ Β

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.

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.



Εικόνα Βιβλίου

Υλοποίηση

Αρχικά, δημιουργούμε μια μήτρα A , δηλαδή:

$$A =$$

Για κάθε A_i το i θα είναι η γραμμή που θα επιλέγεται τυχαία (η πεντάδα). Η σκέψη είναι να επιλέγουμε τυχαία ένα αριθμό $1 \leq N \leq 3$ και να τον εισάγουμε σε μια μήτρα B . Πρώτα, δημιουργούμε μια μηδενική μήτρα B (104×200) για να μπορέσουμε να εισάγουμε τυχαία τις πεντάδες μέσα στην μήτρα B . Επιλέγεται ένα τυχαίο στοιχείο από κάθε πεντάδα και αντικαθιστάτε με τον αριθμό 5.

Η μήτρα B έχει συμπληρωθεί με στοιχεία κάθε γραμμής τυχαία από τον πίνακα A αλλά φυσικά κάθε φορά όταν έχουμε μια πεντάδα, τότε κάποιο νούμερο θα αντικατασταθεί με 5 όπως περιγράφετε πιο πάνω και όπως φαίνεται πιο κάτω στην **Εικόνα 4**.

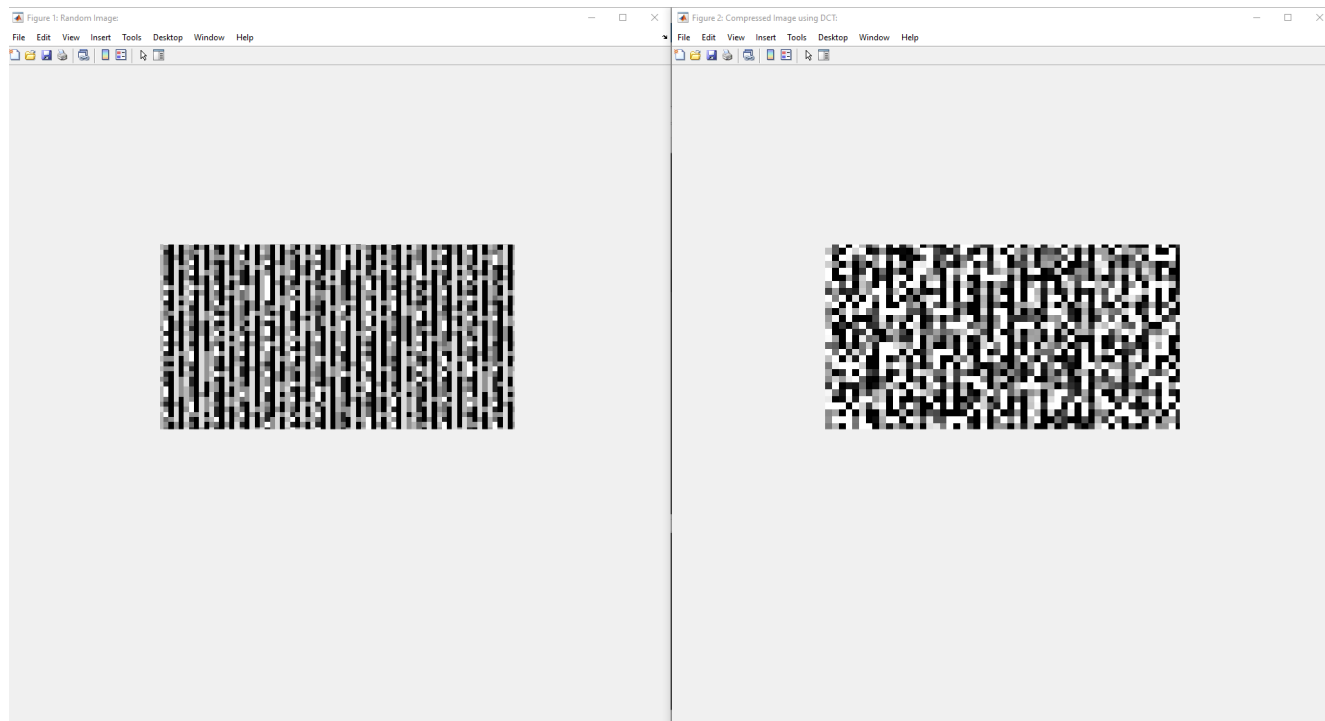
Εικόνα 4

Έπειτα, αφού δημιουργήσουμε την συνθετική εικόνα, θα εφαρμόσουμε την συμπίεση **DCT** (Discrete Cosine Transform). Μετατρέπουμε την εικόνα σε double precision με την εντολή `im2double()/1` και με την εντολή `dctmatrix()/1` παίρνουμε την μήτρα **DCT** (8 x 8). Εφαρμόζουμε **DCT** στα macroblocks 8 x 8. Αρχικοποιούμε την **μήτρα Κβάντισης**,

`q_mtx =`

Στην συνέχεια εφαρμόζουμε Κβάντιση διαιρώντας την μήτρα Κβάντισης με τα macroblocks 8 x 8. Επομένως, βρίσκουμε την αντίστροφη της μήτρας **DCT**, δηλαδή πολλαπλασιάζουμε την μήτρα **q_mtx** με τα macroblocks 8 x 8.

Αποτέλεσμα



Εικόνα 5

Στην **Εικόνα 5** φαίνεται η πρώτη συνθετική εικόνα που δημιουργήσαμε και στο δεύτερο παράθυρο είναι η συμπιεσμένη εικόνα χρησιμοποιώντας τον αλγόριθμο συμπίεσης **DCT**.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΠΗΓΕΣ

Τα clips που χρησιμοποιήθηκαν αποτελούν μέρη των πιο κάτω βίντεο:

B1. <https://pixabay.com/videos/drone-air-3d-model-green-screen-14623/>

B2. <https://www.youtube.com/watch?v=1yT0hxplVBg>

Π1. ΣΥΣΤΗΜΑΤΑ ΠΟΛΥΜΕΣΩΝ - ΑΛΓΟΡΙΘΜΟΙ, ΠΡΟΤΥΠΑ ΚΑΙ ΕΦΑΡΜΟΓΕΣ [2012]: Parag Havaladar & Gerard Medioni,
Επιμέλεια: Επίκουρος Καθηγητής Άγγελος Πικράκης.

Π2. MULTIMEDIA SYSTEMS [2004]: John Buford.

Π3. MULTIMEDIA SYSTEMS [2004]: Klara Nahrstedt & Ralf Steinmetz.

Π4. MULTIMEDIA SYSTEMS AND APPLICATIONS [1995]: R.A. Earnshaw & J.A. Vince.

Π5. https://en.wikipedia.org/wiki/Motion_compensation

Π6. <https://www.youtube.com/watch?v=7QeiKjmxduq>

Π7. <https://www.sciencedirect.com/topics/computer-science/motion-compensation>

Π8. <https://www.immersivelimit.com/tutorials/composing-images-with-python-for-synthetic-datasets>

Π9. <https://towardsdatascience.com/create-a-synthetic-image-dataset-the-what-the-why-and-the-how-f820e6b6f718>

Π10. <https://stackoverflow.com/questions/35703208/how-can-i-make-a-synthetic-image-in-matlab>

- Π11.** <https://www.mathworks.com/matlabcentral/answers/459169-how-to-divide-gray-scale-image-into-8x8-blocks-and-access-each-block-separately-to-apply-dct2-on-the>
- Π12.** <https://www.mathworks.com/matlabcentral/answers/375555-how-to-convert-matrix-into-image>

- Π13.** <https://www.mathworks.com/matlabcentral/answers/217004-need-help-on-my-dct-and-quantization-code-for-image-compression>
- Π14.** https://www.researchgate.net/publication/272676573_Image_Compression_using_Discrete_Cosine_Transform_mDCT_using_MATLAB_blockproc_Function
- Π15.** <https://www.mathworks.com/matlabcentral/fileexchange/48695-dct-image-compression>
- Π16.** <https://www.mathworks.com/help/images/discrete-cosine-transform.html>

BIBΛΙΟΘΗΚΕΣ/ΕΡΓΑΛΕΙΑ

- **Python** v.3.8.2
- **Matlab** v.R2019b
- **Image Processing Toolbox**
- **NumPy Library** v.1.19 (<https://numpy.org/>)
- **OpenCV Library** v.4.3.0.36 (<https://pypi.org/project/opencv-python/>)
- **PyCharm Professional** v.2020.1.1 (IDE)
- **VEGAS PRO** v.17 (Για την επεξεργασία των βίντεο)