Άσκηση 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 λειτουγεί ο αλγόριθμος απομάκρυνσης αντικειμενών για διάφορα μεγέθη μακρομπλόκ.

Περιγραφή συστήματος/ψευδοκώδικας

Για την υλοποίηση και την επίλυση της άσκησης χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python 3. Για να μπορεί να γίνει η εκτέλεση του προγράμματος θα πρέπει να υπάρχουν εγκατεστημένες στο σύστημα οι εξής βιβλιοθήκες: numpy & cv

**Ψευδοκώδικας:**

Συνάρτηση blocks(πλήθος\_γραμμών, πλήθος\_στηλών, πίνακας\_blocks):

Για k μέχρι το πλήθος\_γραμμών:

| Για j μέχρι πλήθος\_στηλών:

| block = πίνακας [ k:k+πλήθος\_γραμμών, c:c+πλήθος\_στηλών]

μετατροπή το block σε πίνακα\_block

*Επιστροφή πίνακα\_block*

Συνάρτηση προσθήκη\_μαύρων\_εικονοστοιχείων(εικόνα):

Ύψος\_εικόνας

Πλάτος\_εικόνας

προσθήκη\_μαύρων\_εικονοστοιχείων = μετατροπή\_σε\_πίνακα(μαύρο\_pixel() \* (((πλάτος-1 // 16+1)

\* 16)- πλάτος))

Για i μέχρι μέγεθος\_εικόνας

| γραμμή = μετατροπή\_σε\_πίνακα(i, προσθήκη\_μαύρων\_εικονοστοιχείων,

| άξονας\_γραμμών)

| προσωρινή\_εικόνα = γραμμή

προσθήκη\_μαύρης\_γραμμής = μετατροπή\_σε\_πίνακα(μαύρο\_pixel() \*

(((πλάτος-1 // 16+1) \* 16)- πλάτος))

Για j μέχρι (((πλάτος-1 // 16+1) \* 16)- πλάτος)):

προσωρινή\_εικόνα = προσθηκη\_μαύρης\_γραμμής

*Επιστροφή μετατροπή\_σε\_πίνακα(προσωινή\_εικόνα)*

Συνάρτηση μαύρο\_εικονστοιχείο():

*Επιστροφή [0]*

Συνάρτηση ιεραρχική\_αναζήτηση(πίνακα):

πίνακα1 = μετατροπή\_σε\_πίνακα()

x, y, = μορφή\_πίνακα /\* να έχουμε τις συντεταγμένες σε μορφή πίνακα\*/

*πίνακα2* = [] /\*αρχικοποιήση του πίνακα2 σαν λίστα\*/

Για i από 0 μέχρι x με\_βήμα 2:

| Για j από 0 μέχρι y με\_βήμα 2:

| | *πίνακα2* = πίνακα1([x], [y]) /\*ο πίνακα2 είναι σε μορφή λίστας \*/

*πίνακα2* = μετατροπή\_σε\_πίνακα()

*πίνακα2* = σχηματισμός(x/2, y/2) /\* για να αποκτήσει τις διαστάσεις που θέλουμε \*/

*Επιστροφή πίνακα2*

Συνάρτηση εύρεση\_κίνησης(πίνακα):

πίνακα1 = μετατροπή\_σε\_πίνακα()

x, y = μορφή\_πίνακα

πλήθος\_μηδενικών\_στοιχείων = x \* y – μη\_μηδενικά\_στοιχεία(πίνακα1)

Αν πλήθος\_μηδενικών\_στοιχείων >= 0.9 \*x \*y:

Επέστρεψε 0

Αλλιώς:

Επέστρεψε 1

Συνάρτηση εύρεση\_διαφορών(εικόνα1, εικόνα2):

«πηγή» = μετατροπή\_σε\_πίνακα()

«στόχος» = μετατροπή\_σε\_πίνακα()

Για i μέχρι 2:

| ιεραρχική\_εικόνα1 = ιεραρχική\_εικόνα1 + [ιεραρχική\_αναζήτηση(«πηγή»)]

| «πηγή» = ιεραρχική\_αναζήτηση(«πηγή»)

| ιεραρχική\_εικόνα2 = ιεραρχική\_εικόνα2 + [ιεραρχική\_αναζήτηση(«στόχος»)]

| «στόχος» = ιεραρχική\_αναζήτηση(«στόχος»)

Συνάρτηση ανακατασκευή\_εικόνας(x, y, εικόνα2):

ρ =1

Για i μέχρι x:

| έξοδος = μετατροπή\_σε\_πίνακα(εικόνα2 [i \*y]) /\* για να πάρουμε τις διαστάσεις

| της κάθε εικόνας\*/

| Για j μέχρι (y-1):

| | έξοδος = ένωσε(έξοδο, εικόνα2[ρ], άξονα\_στηλών)

| | ρ +=1

| ρ +=1

| Αν i == 0:

| ανακατασκευασμένη\_εικόνα = έξοδο

| Αλλιώς:

| ανακατασκευασμένη \_εικόνας = ένωσε((εμφάνιση\_εικονας, έξοδο), άξονας\_γραμμών)

*Επιστροφή ανακατασκευασμένη \_εικόνας*

Συνάρτηση αρχικοποίησης(ς):

ιεραρχική\_εικόνα1 = [εικόνα[0]] /\* λίστα από εικονα[0]\*/

ιεραρχική\_εικόνα2 = [εικόνα[ς]] /\* λίστα από εικόνα [ς]\*/

εύρεση\_διαφορών ([εικόνα[0], εικόνα[ς]) /\* εύρεση διαφορών 2 εικόνων, ρίχνοντας την

ανάλυση \*/

εικόνα1 = blocks(4, 4, ιεραρχική\_εικόνα[2]) /\* διαίρεση της εικόνας σε 4x4 blocks\*/

εικόνα2 = blocks(4, 4, ιεραρχική\_εικόνα[2])

blocks\_move = [] /\*θα αποθηκευτούν τα blocks στα οποία βρέθηκε κίνηση \*/

Για i μέχρι το μέγεθος(εικόνα1):

| Αν εύρεση\_κίνησης (εικόνα1[i]-εικόνα2[i]):

| | blocks\_move = blocks\_move + [i] /\* αν βρεθεί κίνηση την καταγράφει\*/

*Επιστροφή εικόνα1, εικόνα2, ιεραρχική\_εικόνα1, ιεραρχική\_εικόνα2, blocks\_move*

Συνάρτηση κίνηση\_blocks(blocks\_move, ιεραρχική\_εικόνα1, ιεραρχική\_εικόνα2):

size\_block = [8, 16]

Για k μέχρι 2:

| /\* διαίρεση της μικρής εικόνας σε blocks μεγέθους size\_block \* size\_block \*/

| εικόνα1 = blocks(size\_block [k], size\_block [k], ιεραρχική\_εικονα1[size\_block [1- k])

| εικόνα2 = blocks(size\_block [k], size\_block [k], ιεραρχική\_εικονα2[size\_block [1- k])

| blocks\_χωρίς\_κίνηση=[]

| Για i μέχρι το μέγεθος(blocks\_move):/\*ελέγχουμε μόνο τα blocks στα οποία βρήκαμε κίνηση\*/

| | Αν εύρεση\_κίνησης (εικόνα1[blocks\_move[i]] – εικονα2[blocks\_move[i]]):

| | | Συνέχισε

| | Αλλιώς:

| | | blocks\_χωρίς\_κίνηση = blocks\_χωρίς\_κίνηση [i] /\* αν δεν υπάρχει κίνηση

| | | βάζουμε το block στην λίστα\*/

| blocks\_move =[αν κάποιο block\_χωρίς\_κίνηση δεν είναι στη λίστα blocks\_moves,αφαίρεσε το]

| /\* ο κώδικας για την συγκεκριμένη εντολή βρίσκετε στη γραμμή 128\*/

*Επιστροφή εικόνα1, εικόνα2, blocks\_move*

Συνάρτηση main (εικόνες):

Αν (εικόνες[0] ή εικόνες[1]) % 16 !=0 /\* αν οι εικόνες δεν διαιρούνται ακριβώς σε block 16x16

| προσθέτουμε μαύρα εικονοστοιχεία\*/

| Για i μέχρι το μέγεθος(εικόνες):

| εικόνες[i] = πρόσθεση\_μαύρων\_εικονοστοιχείων(εικόνες[i])

εικόνες = μετατροπή\_σε\_πίνακα(εικόνες)

Για ς μέχρι το μέγεθος(εικόνες):

| /\* αρχικποιήση των πινάκων με τις διαστάσεις, από την συνάρτηση αρχικοποίησης\*/

| εικόνα1, εικόνα2, ιεραρχική\_εικόνα1, ιεραρχική\_εικόνα2, blocks\_move = αρχικοποίηση(ς)

| /\* υπολογισμός των block που θα αντικατασταθούν, από την συνάρτηση κίνηση\_blocks\*/

| εικόνα1, εικόνα2, blocks\_move = κίνηση\_blocks(blocks\_move, ιεραρχική\_εικόνα1,

| ιεραρχική\_εικόνα2)

| Για i μέχρι το μέγεθος(blocks\_move):

| | εικόνα2[blocks\_move[i]] = εικόνα1[blocks\_move[i]] /\*αντικατάσταση κινούμενων block

| | με παρασκήνιο \*/

| Διαίρεση για μπλοκ μεγέθους 16\*16

| εικόνες[ς] = ανακατασκευή\_εικόνας(x, y, εικονα2)

*Επιστροφή εικόνες*

Αρχικοποιήσεις (γραμμές 157-166)

Φόρτωση Βίντεο

Αποθήκευση του πρώτου frame που έχει μόνο το παρασκήνιο

Καταγραφή πλήθους frames

Δημιουργία νέου επεξεργασμένου βίντεο, χωρίς το κινούμενο αντικείμενο.

**ΥΛΟΠΟΙΗΣΗ**

Φορτώνουμε το βίντεο και το διαιρούμε σε μπλοκς για να μπορέσουμε να εργαστούμε, με την συνάρτηση blocks. Στην περίπτωση όπου δεν διαιρούνται ακριβώς, βάζουμε με μαύρο χρώμα pixels για να καλυψουμε τη διαφορά, χρησιμοποιώντας την συνάρτηση add\_black\_pixels. Για να βρούμε τα μπλοκ ταιριάσματος χρησιμοποιήσαμε ιεραρχική αναζήτηση, με την συνάρτηση hierarchical\_search, όπου δέχεται ως παράμετρο ένα πίνακα κάθε φορά. Στην συνάρτηση find\_differences βρίσκουμε τις διαφορές που υπάρχουν μεταξύ των 2 εικόνων, source & target (η source αντιπροσωπεύει την αρχική εικόνα ενώ η target την εικόνα στην οποία θα γίνει η αναζήτηση και το ταίριασμα), καλείται δύο φορές η συνάρτηση hierarchical\_search, μια για την κάθε εικόνα (πίνακα). Για να ελέγχξουμε την κίνηση που υπάρχει θα πρέπει να βρούμε τα μπλοκς τα οποία δεν έχουν μηδενικές τιμές. Αυτή η διαδικασία γίνεται καλώντας την συνάρτηση find\_motion όπου δέχεται ως είσοδο ένα πίνακα και ελέγχει το πλήθος των μηδενικών από την διαφορά της επόμενης μείον της πρώτης εικόνας. Αν υπάρχει κίνηση, δηλαδή δεν υπάρχουν μηδενικά, τότε προσθέτουμε το στοιχείο στην λίστα blocks\_move. Με την συνάρτηση move\_in\_blocks, όπου έχει ως παραμέτρους τη λίστα blocks\_move και τις εικόνες από την ιεραρχική αναζήτηση, ελέγχουμε ποια μπλοκς έχουν κίνηση και πια όχι. Τα μπλοκς στα οποία δεν βρέθηκε κίνηση τα αποθηκεύουμε στην λίστα blocks\_no\_move και την επιστρέφουμε. Για την αναδημιουργία της εικόνας στην οποία βρέθηκε η κίνηση, υλοποιήσαμε την συνάρτηση image\_recreate όπου δέχεται ως παραμέτρους τις διαστάσεις και την νέα εικόνα στην οποία έχει αφαιρεθεί το αντικείμενο. Η συνάρτηση επιστρέφει την νέα εικόνα (showim στον κώδικα).

Τέλος το πρόγραμμα ξεκινάει από την main, καλώντας τις συναρτήσεις. Παρακάτω ακολουθούν κάποιες αναγκαίες αρχικοποιήσεις, καθώς και αναπαραγωγή – δημιουργία του νέου επεξεργασμένου βίντεο, μετά την αφαίρεση του κίνουμε αντικειμένου.

**ΑΠΟΤΕΛΕΣΜΑΤΑ**

Στο φάκελο στον οποίο υπάρχει το εκτέλεσιμο αρχείο με τον κώδικα, μετά το τέλος της εκτέλεσής του δημιουργείται ένα νέο αρχείο με όνομα edit\_video.avi, το οποίο είναι και το αποτέλεσμα της εκτέλεσης του προγράμματος.

**ΣΥΜΠΕΡΑΣΜΑΤΑ**

**ΒΙΟΒΛΙΟΓΡΑΦΙΑ**