

Συστήματα Πολυμέσων και Εικονική Πραγματικότητα

Εργασία 2019-2020

Ομάδα Κατανόησης Πολυμέσων

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

1 Εισαγωγικές Παρατηρήσεις

Η παρακάτω εργασία αποτελεί *προαιρετικό* μέρος του μαθήματος Συστήματα Πολυμέσων και Εικονική Πραγματικότητα και η εκτέλεσή της συνεισφέρει 1,3 ή και 4 επιπλέον μονάδες στην τελική βαθμολογία.

Η εργασία **είναι ατομική** και αποτελείται από 3 ενότητες. Η υλοποίηση της εργασίας μπορεί κατ' επιλογή να περιλάβει την πρώτη ενότητα, την πρώτη και τη δεύτερη, κ.ο.κ. συνεισφέροντας τις αντίστοιχες μονάδες για κάθε ενότητα. Δεν μπορεί όμως να εκτελεστεί μία ενότητα χωρίς να έχει ορθά εκτελεστεί η προηγούμενή της.

Η εργασία στοχεύει στην υλοποίηση ενός κωδικοποιητή/αποκωδικοποιητή ακίνητης εικόνας κατά το πρότυπο JPEG (ISO/IEC 10918-1:1994). Συγκεκριμένα, θα υλοποιηθεί η εκδοχή baseline sequential DCT-based που είναι και η πιο απλή. Ωστόσο, θα προσπαθήσουμε να παρεκκλίνουμε το δυνατόν λιγότερα από το πρότυπο.

2 Διάρθρωση και Παραδοτέα

Η εργασία θα περιλαμβάνει τις συναρτήσεις που περιγράφονται στη συνέχεια, υλοποιημένες σε MATLAB. Θα συνοδεύεται υποχρεωτικά από γραπτή αναφορά η οποία θα περιγράφει τον τρόπο χρήσης των προγραμμάτων και θα επιδεικνύει ενδεικτικά αποτελέσματα.

Για τον έλεγχο του κωδικοποιητή/αποκωδικοποιητή που θα κατασκευαστεί θα χρησιμοποιηθούν RGB εικόνες. Η βαθμολογία θα εξαρτηθεί: 1) από την λειτουργικότητα του κώδικα, 2) την ποιότητα των ανακατασκευασμένων εικόνων και 3) την πληρότητα της αναφοράς. Συναρτήσεις οι οποίες δεν λειτουργούν δεν θα βαθμολογούνται. Για τον παραπάνω λόγο, καλό θα ήταν να μην χρησιμοποιήσετε ελληνικούς χαρακτήρες στα σχόλια του κώδικά σας, αντ' αυτού μπορείτε να γράψετε με λατινικούς χαρακτήρες (i.e. greeklish).

3 JPEG Library

Σκοπός του πρώτου παραδοτέου είναι η δημιουργία μιας βιβλιοθήκης συναρτήσεων που υλοποιούν τμήματα του προτύπου. Κάθε συνάρτηση της βιβλιοθήκης συνοδεύεται από την αντίστροφή της που ανακατασκευάζει την είσοδο της πρώτης και που θα αποτελέσει μέρος του αποκωδικοποιητή. Για να εξοικειωθείτε με το πρότυπο συνιστούμε να διαβάσετε πρώτα την ενότητα 4.1 και το παράρτημα Α.

3.1 Προεπεξεργασία

Η αρχική εικόνα περνά αρχικά από ένα στάδιο προεπεξεργασίας με σκοπό τη μετατροπή της από RGB σε YCbCr. Θα πρέπει να υπάρχει δυνατότητα υποδειγματοληψίας των πεδίων της χρωματικότητας. Οι πιθανές μορφές υποδειγματοληψίας είναι 4:4:4, 4:2:2 και 4:2:0. Επομένως θα πρέπει να κατασκευαστεί η συνάρτηση:

```
1 function [imageY, imageCb, imageCr] = convert2ycbcr(imageRGB, subimg
    )
```

όπου,

imageY: Το Y component της εικόνας.

imageCb: Το Cb component της εικόνας.

imageCr: Το Cr component της εικόνας.

imageRGB: Η εικόνα σε RGB.

subimg: Πίνακας 1×3 που καθορίζει την υποδειγματοληψία (π.χ. [4 2 0]).

Η αντίστροφή συνάρτηση της παραπάνω είναι:

```
1 function imageRGB = convert2rgb(imageY, imageCr, imageCb, subimg)
```

όπου τα ορίσματα έχουν γενικά την ίδια σημασία, ωστόσο τώρα θα πρέπει να γίνει υπερδειγματοληψία, προκειμένου η ανακατασκευασμένη εικόνα imageRGB να έχει ίδια διάσταση με την αρχική. Οι διαστάσεις της αρχικής εικόνας θεωρούμε ότι είναι πολλαπλάσια του 8, προκειμένου να μπορεί να χωριστεί σε ακέραιο αριθμό blocks. Σε αντίθετη περίπτωση θα πρέπει να αφαιρούνται οι οριζόντιες γραμμές και στήλες μέχρι να ικανοποιηθεί αυτή η συνθήκη.

3.2 Μετασχηματισμός DCT

Οι συναρτήσεις εδώ είναι σε επίπεδο block. Επομένως, για την εφαρμογή του μετασχηματισμού DCT κατασκευάστε την συνάρτηση:

```
1 function dctBlock = blockDCT(block)
```

καθώς και την αντίστροφή της:

```
1 function block = iBlockDCT(dctBlock)
```

όπου,

dctBlock: Οι DCT coefficients του block.

block: Το block της εισόδου.

Για την υλοποίηση αυτού του βήματος συμβουλευτείτε τις ενότητες A.3.1, A.3.2 του προτύπου.

Για δική σας διευκόλυνση, συνίσταται η χρήση των συναρτήσεων dct2 και idct2 της MATLAB.

3.3 Κβαντισμός

Για τον κβαντισμό των blocks κατασκευάστε την συνάρτηση:

```
1 function qBlock = quantizeJPEG(dctBlock, qTable, qScale)
```

και την αντίστροφή της:

```
1 function dctBlock = dequantizeJPEG(qBlock, qTable, qScale)
```

όπου,

qBlock: Τα σύμβολα κβαντισμού των DCT coefficients του block.

dctBlock: Οι DCT coefficients του block.

qTable: Ο πίνακας κβαντισμού.

qScale: Η κλίμακα κβαντισμού.

Ο πίνακας κβάντισης διαφέρει ανάλογα με τον τύπου του block. Για την υλοποίηση αυτών των συναρτήσεων συμβουλευτείτε την ενότητα A.3.4. Για να μπορούμε να συγκρίνουμε τα αποτελέσματα σας, τα qTable που θα χρησιμοποιήσετε θα είναι οι αντίστοιχοι πίνακες κβαντισμού που υπάρχουν στην ενότητα K.1. Ο πίνακας κβαντισμού που θα εφαρμόζεται σε ένα block θα προκύπτει από το γινόμενο του qScale με το qTable.

3.4 Zig-zag scanning και RLE

Κατασκευάστε τη συνάρτηση υπολογισμού των συμβόλων μήκους διαδρομής για τους κβαντισμένους συντελεστές DCT:

```
1 function runSymbols = runLength(qBlock, DCpred)
```

και την αντίστροφή της:

```
1 function qBlock = irunLength(runSymbols, DCpred)
```

όπου,

runSymbols: Ο πίνακας που περιέχει τα σύμβολα μήκους διαδρομής τα οποία είναι δυάδες της μορφής (precedingZeros, quantSymbol). Ο πίνακας έχει συνεπώς διαστάσεις $R \times 2$, όπου R τα μήκη διαδρομής που εντοπίστηκαν στο συγκεκριμένο block. Ο DC όρος θεωρούμε ότι έχει μηδενικό μήκος διαδρομής.

qBlock: Τα σύμβολα κβαντισμού των DCT coefficients.

DCpred: Η πρόβλεψη για τον DC όρο με βάση το προηγούμενο block.

Τα σύμβολα δεν κωδικοποιούνται με τη σειρά 'γραμμή-στήλη', αλλά χρησιμοποιώντας zig-zag scanning. Σύμφωνα με το πρότυπο, η διαδικασία κωδικοποίησης RLE ακολουθείται μόνο για τους AC συντελεστές κάθε block. Για τους DC συντελεστές κωδικοποιούμε τις διαφορές τους. Για την υλοποίηση αυτών των συναρτήσεων ανατρέξτε στις ενότητες A.3.5 και A.3.6.

3.5 Κωδικοποίηση Huffman

Για την κωδικοποίηση των συμβόλων μήκους διαδρομής το πρότυπο προτείνει είτε κωδικοποίηση Huffman είτε αριθμητική κωδικοποίηση. Στην εργασία, θα χρησιμοποιήσουμε κωδικοποίηση Huffman, η οποία είναι απλούστερη και πιο διαδεδομένη. Κατασκευάστε τη συνάρτηση:

```
1 function huffStream = huffEnc(runSymbols)
```

και την αντίστροφή της:

```
1 function runSymbols = huffDec(huffStream)
```

όπου,

huffStream: stream από bits, που περιέχει την κωδικοποιημένη πληροφορία για ένα block.

runSymbols: Τα σύμβολα μήκους διαδρομής.

Για την υλοποίηση αυτών των συναρτήσεων μελετήστε προσεκτικά τις ενότητες F.1.2 και F.2.2 του προτύπου. Για δικιά σας διευκόλυνση δεν θα κατασκευάσετε καινούργιους κώδικες Huffman για κάθε εικόνα, αλλά θα χρησιμοποιήσετε τους πίνακες F.1, F.2, K.3, K.4, K.5 και K.6 του προτύπου. Τους πίνακες αυτούς είναι καλό να τους έχετε διαθέσιμους σαν global μεταβλητές. Υπενθυμίζουμε ότι η διαδικασία κωδικοποίησης διαφέρει ελαφρώς για τους DC και τους AC όρους.

3.6 Demo 1

Δημιουργήστε το αρχείο `demo1.m` στο οποίο θα δείχνετε:

α) Την διαδικασία μετατροπής μίας εικόνας από RGB σε YCbCr και στην συνέχεια πίσω σε RGB χρησιμοποιώντας τις ρουτίνες που ήδη κατασκευάσατε. Για είσοδο χρησιμοποιείτε και τις 2 εικόνες που σας δίνονται σε μορφή `mat`. Για την εικόνα 1 χρησιμοποιείτε υποδειγματοληψία 4:2:2 ενώ για την εικόνα 2 χρησιμοποιείτε 4:4:4.

β) Την διαδικασία μετατροπής μίας εικόνας από RGB σε YCbCr, τον υπολογισμό των κβαντισμένων DCT συντελεστών και στην συνέχεια πίσω σε RGB χρησιμοποιώντας τις ρουτίνες που έχετε κατασκευάσει ήδη. Για την υποδειγματοληψία χρησιμοποιείτε τις τιμές που δόθηκαν στο ερώτημα α) ενώ για τις τιμές του `qScale` χρησιμοποιείτε 0.6 για την εικόνα 1 και 5 για την εικόνα 2.

Σε κάθε περίπτωση, το `demo` θα πρέπει να οπτικοποιεί τα αποτελέσματα (πριν και μετά την ανακατασκευή). Καταγράψτε τα αποτελέσματα σας στην αναφορά.

4 JPEG Intergration

Το δεύτερο παραδοτέο έχει σαν στόχο την ενσωμάτωση των συναρτήσεων του προηγούμενου επιπέδου σε μία γενική συνάρτηση και την εξαγωγή ποσοτικών και ποιοτικών συμπερασμάτων για την συμπίεση που επιτυγχάνεται.

4.1 JPEG Encoder/Decoder

Κατασκευάστε την συνάρτηση:

```
1 function JPEGenc = JPEGencode(img, subimg, qScale)
```

και την αντίστροφή της:

```
1 function imgRec = JPEGdecode(JPEGenc)
```

όπου,

`img`: το όνομα της εικόνας προς επεξεργασία. Θα σας δοθεί σχετική ακολουθία από εικόνες μαζί με την εκφώνηση της εργασίας.

`qScale`: Η κλίμακα κβαντισμού.

`subimg`: Πίνακας 1×3 που καθορίζει την υποδειγματοληψία.

`JPEGenc`: cell από struct της MATLAB. Αν ο αριθμός των block είναι N , τότε ο αριθμός των στοιχείων του cell θα είναι $N + 1$. Το πρώτο στοιχείο του cell θα είναι ένα struct με τα εξής πεδία:

- `qTableL` : ο πίνακας κβαντισμού για την φωτεινότητα.
- `qTableC` : ο πίνακας κβαντισμού για την χρωματικότητα.
- `DCL` : cell με τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των DC συντελεστών για block φωτεινότητας.
- `DCC` : cell με τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των DC συντελεστών για block χρωματικότητας.
- `ACL` : cell με τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των AC συντελεστών για block φωτεινότητας.
- `ACC` : cell με τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των AC συντελεστών για block χρωματικότητας.

Τα υπόλοιπα στοιχεία του cell θα είναι struct που θα περιέχουν τα πεδία :

- `blkType` : ο τύπος του block ("Y", "Cb", "Cr").
- `indHor` : δείκτης που δείνει την οριζόντια θέση του block.
- `indVer` : δείκτης που δείνει την κατακόρυφη θέση του block.
- `huffStream` : το block κωδικοποιημένο. Μια ακολουθία απο bits.

`imgRec`: η ανακατασκευασμένη εικόνα.

4.2 Καταγραφή αποτελεσμάτων

Εφαρμόστε τις παραπάνω συναρτήσεις στις εικόνες που σας δόθηκαν, για διάφορες τιμές του `qScale`. Στην αναφορά σας συμπεριλάβετε τις τιμές (0.1, 0.3, 0.6, 1, 2, 5, 10). Τι παρατηρείτε για την ποιότητα της εικόνας; Κατασκευάστε αντίστοιχα διαγράμματα με την τιμή του MSE (Mean Squared Error). Σε κάθε πείραμα μετρήστε τον αριθμό των bits της κωδικοποιημένης εικόνας. Ποια η σχέση του αριθμού των bits με το MSE; Τροποποιήστε τους πίνακες κβαντισμού, ώστε να μηδενιστούν οι 20, 40, 50, 60 και 63 πλέον υψίσυχνι όροι των `dctBlock` για `qScale = 1`. Τι παραμορφώσεις εισάγονται και σε ποιες εικόνες; Μπορείτε να δώσετε κάποια ερμηνεία;

4.3 Demo 2

Δημιουργήστε το αρχείο `demo2.m` στο οποίο θα υπολογίζετε:

- Την εντροπία στο spatial domain (i.e. RGB).
- Την εντροπία των κβαντισμένων συντελεστών DCT.

γ) Την εντροπία χρησιμοποιώντας τα μήκη διαδρομής.

Επαναλάβετε τα παραπάνω και για τις 2 εικόνες που σας δίνονται.

Καταγράψτε τα αποτελέσματα σας στην αναφορά. Οι τιμές των `subimg` και `qScale` για τις 2 εικόνες θα είναι ίδιες με αυτές που δώσατε στο Demo 1.

5 JPEG Syntax

Το τρίτο παραδοτέο αποσκοπεί στην κατασκευή ενός `bitstream` με την κωδικοποιημένη εικόνα καθώς και στην αποκωδικοποίηση του. Το αρχείο που θα προκύπτει με την αποθήκευση του `bitstream` σε `binary` μορφή θα πρέπει να είναι αποκωδικοποιήσιμο από οποιοδήποτε πρόγραμμα επεξεργασίας εικόνας.

5.1 JPEG Syntax Encoder/Decoder

Κατασκευάστε την συνάρτηση:

```
1 function JPEGencStream = JPEGencodeStream(img, subimg, qScale)
```

και την αντίστροφή της:

```
1 function imgCmp = JPEGdecodeStream(JPEGencStream)
```

όπου τα ορίσματα είναι τα ίδια με παραπάνω με την εξαίρεση του `JPEGencStream` που είναι ένα `bitstream` με την κωδικοποιημένη εικόνα. Για την υλοποίηση αυτού του τμήματος της εργασίας θα πρέπει να συμβουλευτείτε το παράρτημα Β του προτύπου. Ποιος είναι ο λόγος συμπίεσης για τις διάφορες τιμές του `qScale`;

Σχετικά με την υποβολή της εργασίας

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Η αναφορά θα πρέπει να επιδεικνύει την ορθή λειτουργία του κώδικά σας στις εικόνες που σας δίνονται.

Ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλήσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει τα σωστά αποτελέσματα για *οποιαδήποτε* είσοδο πληροί τις υποθέσεις της εκφώνησης, και όχι μόνο για τις εικόνες που σας δίνονται.

Απαραίτητες προϋποθέσεις για την βαθμολόγηση της εργασίας σας είναι ο κώδικας να εκτελείται χωρίς σφάλμα, καθώς και να τηρούνται τα ακόλουθα :

- Υποβάλετε ένα και μόνο αρχείο, τύπου zip.
- Το όνομα του αρχείου πρέπει να είναι AEM.zip, όπου AEM είναι τα τέσσερα ψηφία του Α.Ε.Μ. του φοιτητή.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα MATLAB (συναρτήσεις και demos) και το αρχείο report.pdf το οποίο θα είναι η αναφορά της εργασίας. Κάθε συνάρτηση θα πρέπει να είναι ένα ξεχωριστό αρχείο .m με όνομα ίδιο με αυτό της συνάρτησης που υλοποιεί.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου PDF, και να έχει όνομα report.pdf.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου UTF-8, και να έχουν κατάληξη m.
- Το αρχείο τύπου zip που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Μην υποβάλετε τις εικόνες που σας δίνονται για πειραματισμό.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ “Thumbs.db”, “.DS_Store”, “.directory”.
- Για την ονομασία των αρχείων που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ “”, “\$”, “%” κλπ.