Συστήματα Πολυμέσων Εργασία
Απλοποιημένος codec JPEG

Α. Ντελόπουλος

2023-2024

1 Εισαγωγικές Παρατηρήσεις

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

Η εργασία είναι ατομική και αποτελείται απο 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 εκτελεστεί ορθά 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 Python. Θα συνοδεύεται υποχρεωτικά από γραπτή αναφορά η οποία θα περιγράφει τον τρόπο χρήσης των προγραμμάτων και θα επιδυκνύει ενδεικτικά αποτελέσματα.

Για τον έλεγχο του κωδικοποιητή/αποκωδικοποιητή που θα κατασκευαστει θα χρησιμοποιηθούν RGB εικόνες. Η βαθμολογία θα εξαρτηθει: 1) από τη λειτουργικότητα του κώδικα, 2) την ποιότητα των ανακατασκευασμένων εικόνων και 3) την πληρότητα της αναφορας. Συναρτήσεις οι οποίες δεν λειτουργούν δε θα βαθμολογούνται. Για τον παραπάνω λόγο, καλό θα ήταν να ΜΗΝ χρησιμοποιείτε 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 από ένα στάδιο προεπεξεργασίας με σκοπό τη μετατροπή της από RGB σε YCrCb. 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 υλοποιηθεί η συνάρτηση:

```
imageY, imageCr, imageCb = convert2ycrcb(imageRGB, subimg)
```

όπου,

- `imageY`: το Y component της εικόνας.
- `imageCr`: το Cr component της εικόνας.
- `imageCb`: το Cb component της εικόνας.
- `imageRGB`: Η εικόνα RGB.
- `subimg`: Πίνακας 1×3 που ρυθμίζει την υποδειγματοληψία (π.χ. [4, 2, 0].

η αντίστροφη συνάρτηση της παραπάνω είναι η:

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

όπου τα ορίσματα έχουν γενικά την ίδια σημασία, ωστόσο τώρα θα πρέπει να γίνει υπερδειγματοληψία, προκειμένου η ανακατασκευασμένη εκόνα `imageRGB` να έχει την 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 οριακές γραμμές και στήλες μέχρι να ικανοποιηθεί αυτή η συνθήκη.

3.2 Μετασχηματισμός DCT

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

```
dctBlock = blockDCT(block)
```

καθώς και η αντιστροφή της:

```
block = iBlockDCT(dctBlock)
```

όπου,

- dctBlock: Οι DCT coefficients του block.
- block: Τα block της εισόδου.

Για την υλοποίηση του βήματος συμβουλευτείτε τις ενότητες Α.3.1, Α.3.2 του προτύπου. Για τη δική σας διευκόλυνση, συνίσταται η χρήση έτοιμων γρήγορων υλοποιήσεων ευθύ και αντίστροφου DCT της βιβλιοθήκης opencv ή scipy.

3.3 Κβαντισμός

Για τον κβαντισμό των block υλοποιήστε τη συνάρτηση:

```
qblock = quantizeJPEG(dctBlock, qTable, qScale)
```

και την αντίστροφή της:

```
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 κβαντισμού διαφέρει ανάλογα με τον τύπο του block. Για την υλοποίηση 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 προκύπτει από το γινόμενο του qScale με το `qTable`.

3.4 Zig-zag scanning και RLE

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

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

και την αντίστροφή της:

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

όπου,

- `runLength`: Ο πίνακας που περιέχει τα σύμβολα μήκους διαδρομής, τα οποία είναι δυάδες της μόρφής `(precedingZeros, quantSymbol)`. Ο πίνακας έχει συνεπώε 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 όρο DC 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 συντελεστές AC 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:

```
huffStream = huffEnc(runSymbols)
```

και την αντίστροφή της:

```
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 F2.2 του προτύπου. Για δική σας διευκόλυνση δεν θα κατασκευάσετε νέους κώδικες Huffman για κάθε εικόνα, αλλα θα χρησιμοποιήσετε τους πίνακες F.1, F.2, K.3, K.4, K.5 και K.6 του προτύπου. Υπενθυμίζουμε ότι η διαδικασία κωδικοποίησης διαφέρει ελαφρώς για τους DC και τους AC όρους.

3.6 Demo 1

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

1. Την διαδικασία μετατροπής μίας εικόνας από RGB σε YCrCb και στη συνέχεια πίσω σε RGB χρησιμοποιώντας τις συναρτήσεις που υλοποιήσατε. Για είσοδο χρησιμοποιείστε και τις 2 εικόνες που σας δίνονται. Για την εικόνα 1 χρησιμοποιείστε υποδειγματοληψία 4:2:2, ενώ για την εικόνα 2 χρησιμοποιείστε υποδειγματοληψία 4:4:4.

2. Τη διαδικασία μετατροπής μίας εικόνας από RGB σε YCrCb, τον υπολογισμό των κβαντισμένων DCT συντελεστών, και στη συνέχεια πίσω σε RGB χρησιμοποιώντας τις συναρτήσεις που υλοποιήσατε. Για την υποδειγματοληψία χρησιμοποιείστε τις τιμές που δόθηκαν στο ερώτημα 1, ενώ για τις τιμές του `qScale` χρησιμοποιήστε 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). Για την οπτικοποίηση των αποτελεσμάτων μπορείτε να χρησιμοποιήσετε τη βιβλιοθλήκη `matplotlib`. Καταγράψτε τα αποτελέσματά σας στην αναφορά.

3.7 JPEG Integration

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 τη συμπίεση που επιτυγχάνεται.

3.8 JPEG Encoder/Decoder

Κατασκευάστε τη συνάρτηση:

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

και την αντίστροφή της:

```
imgRec = JPEGdecode(JPEGenc)
```

όπου,

- `img`: Η εικόνα 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`: `tuple` από κλάσεις που θα πρέπει να υλοποιήσετε. Αν ο αριθμός των block είναι N, τότε ο αριθμός των στοιχείων του `JPEGenc` θα είναι $N + 1$. Για το πρώτο στοιχείο, κατασκευάστε μία κλάση με τα παρακάτω attributes:
  - `qTableL`: ο πίνακας κβαντισμού για τη φωτεινότητα.
  - `qTableC`: ο πίνακας κβαντισμού για τη χρωματικότητα.
  - `DCL`: τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των DC συντελεστών για block φωτεινότητας.
  - `DCC`: τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των DC συντελεστών για block χρωματικότητας.
  - `ACL`: τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των AC συντελεστών για block φωτεινότητας.
  - `ACC`: τα στοιχεία που χρησιμοποιήθηκαν για την κωδικοποίηση των AC συντελεστών για block χρωματικότητας.

  Τα υπόλοιπα στοιχεία του `tuple` θα είναι αντικείμενα κλάσης με τα παρακάτω attributes:
  - `blkType`: Ο τύπος του block (`"Y", "Cr", "Cb"`).
  - `indHor`: δείκτης που δίνει την οριζόντια θέση του block.
  - `indVer`: δείκτης που δίνει την κάθετη θέση του block.
  - `huffStream`: το κωδικοποιημένο block. Μία ακολουθία από bits.
  - `imgeRec`: η ανακατασκευασμένη εικόνα.

3.9 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 τα 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; Τροποποιείστε τους πίνακες κβαντισμού, ώστε να μηδενιστούν οι 20, 40, 50, 60 και 63 πλέον υψίσυχνοι όροι των `dctBlock` για `qScale` = 1. Τι παραμορφώσεις εισάγονται και σε ποιες εικόνες; Μπορείτε να δώσετε κάποια ερμηνεία;

3.10 Demo 2

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

1. Την εντροπία στο spatial domain (i.e. RGB).
2. Την εντροπία των κβαντισμένων συντελεστών DCT.
3. Την εντροπία χρησιμοποιώντας τα μήκη διαδρομής.

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

4 JPEG Syntax

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

4.1 JPEG Syntax Encoder/Decoder

Κατασκευάστε τη συνάρτηση:

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

και την αντίστροφή της:

```
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 `qScale`;

5 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 επιτελέι (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλήσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται και να υπολογίζει 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 τηρούνται τα ακόλουθα:

- 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 Python (συναρτήσεις και demos) και το αρχείο `report.pdf` το οποίο θα είναι η ανφορά της εργασίας.
- Η αναφορά πρέπει να είναι 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 `.py`.
- Το αρχείο τύπου `.zip` που θα υποβάλετε δεν πρέπει να περιέχει κανέναν φάκελο.
- Να υποβάλετε και τις εικόνες που σας δίνονται για τον πειραματισμό.
- Μην υποβάλετε αρχεία που δε χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ.`"Thumbs.db"`, `".DS\_Store"`.
- Για την ονομασία των αρχείων χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ. `"¨"`, `"\$"` κλπ.