

# Ψηφιακή Επεξεργασία Εικόνας

## -Εργασία 1-

### Bayer, downsampling, quantization & PPM representation

Ομάδα Κατανόησης Πολυμέσων  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ  
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Άνοιξη 2021

## Εισαγωγικά

Στην πρώτη εργασία του μαθήματος θα υλοποιήσετε τα ακόλουθα

1. Ανακατασκευή τριχρωματικής εικόνας RGB από εικόνα Bayer
2. Υποδειγματοληψία της RGB εικόνας
3. Κβαντισμός των τιμών φωτεινότητας ανά κανάλι
4. Αποθήκευση των κβαντισμένων τιμών σε διαδική μορφή

Για τους σκοπούς της εργασίας θεωρείστε ότι έχετε τις πραγματικές τιμές φωτεινότητας στο διάστημα  $[0, 1]$  τις οποίες κατέγραψε η φωτογραφική μηχανή. Οι τιμές φωτεινότητας που κατέγραψε η φωτογραφική μηχανή σας δίνονται στο αρχείο `march.mat`.

## 1 Φίλτρο Bayer

Σε πολλούς έγχρωμους αισθητήρες CCD ή CMOS η διάταξη των τριών χρωματιστών (r, g, b) κυττάρων ακολουθεί τη δομή της εικόνας που είναι γνωστή ως φίλτρο Bayer. Η μετατροπή σε RGB pixels γίνεται στη συνέχεια με συνέλιξη (convolution).

Να κατασκευάσετε συνάρτηση `bayer2rgb`

```
1 function xrgb = bayer2rgb(xb)
```

που δέχεται σαν είσοδο τον  $M_0 \times N_0$  πίνακα `xb` με τις τιμές των δειγμάτων μιας εικόνας σε θέσεις που περιγράφονται από το φίλτρο Bayer (Εικόνα 1) και επιστρέφει εικόνα διάστασης  $(M_0, N_0)$  σε μορφή RGB (δηλαδή  $M_0 \times N_0 \times 3$ ). Στο εσωτερικό της, η συνάρτησή σας θα πρέπει να πραγματοποιεί συνέλιξεις με σκοπό να γεμίσει τα κενά pixel κάθε καναλιού της RGB εικόνας `xrgb` με τον μέσο όρο (averaging) των γειτονικών τους (όπως αυτά προέρχονται από την αρχική, κατά bayer, εικόνα `xb`).

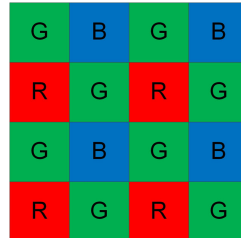
Για ευκολία και ταχύτητα, μπορείτε να χρησιμοποιείτε τις εντολές `convn` και `conv2` της MATLAB. Επιπλέον, δεν είναι υποχρεωτικό να χρησιμοποιείτε τις ίδιες μάσκες (kernels) για όλα τα χρωματικά κανάλια.

## 1.1 Demo 1

Υλοποιήστε το MATLAB script `demo1.m` στο οποίο να δείχνετε:

1. Τα διαφορετικά kernels που χρησιμοποιείτε για συνέλιξη
2. Την έξοδο της συνάρτησης σας με είσοδο την εικόνα που σας δίνεται στο αρχείο `march.mat`

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας. Πετυχαίνετε όντως averaging;



Σχήμα 1: Μέρος μίας δομής Bayer. Αξίζει να παρατηρηθεί πως ο αριθμός των πράσινων αισθητήρων είναι διπλάσιος από αυτούς των μπλε και κόκκινων.

## 2 Υποδειγματοληψία εικόνας

Σε αυτή την ενότητα θα υλοποιήσετε μία μέθοδο η οποία θα υποδειγματοληπτεί την εικόνα εισόδου. Υλοποιήστε την συνάρτηση `myresize`

```
1 function xrgbres = myresize(xrgb, N, M, method)
```

που δέχεται σαν είσοδο την RGB εικόνα `xrgb` και την υποδειγματοληπτεί στην εικόνα `xrgbres` με διαστάσεις  $N \times M$ . Επιπλέον, όταν η μεταβλητή `method` έχει την τιμή 'nearest' τότε κατά την παρεμβολή θα πρέπει να χρησιμοποιείται η μέθοδος του κοντινότερου γείτονα. Όταν η ίδια μεταβλητή δέχεται την τιμή 'linear', τότε θα πρέπει να χρησιμοποιήσετε διγραμμική παρεμβολή (bilinear interpolation).

### 2.1 Demo 2

Υλοποιείτε το MATLAB script `demo2.m` στο οποίο θα πρέπει να δείξετε την έξοδο της συνάρτησής σας για:

1.  $N = 240$  και  $M = 320$  και `method = 'nearest'`
2.  $N = 200$  και  $M = 300$  και `method = 'linear'`

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας.

## 3 Κβαντισμός και αποκβαντισμός

Σε αυτή την ενότητα θα υλοποιήσετε ένα σύστημα κβαντιστή και αποκβαντιστή. Αρχικά υλοποιήστε τη συνάρτηση

```
1 function q = myquant(x, w)
```

η οποία είναι ένας ομοιόμορφος συμμετρικός κβαντιστής (όχι νεκρής ζώνης), με πλάτος ζώνης κβαντισμού  $w$ . Υιοθετήστε τη σύμβαση ότι τα σύμβολα κβαντισμού είναι ακέραιοι, οι οποίοι ανατίθενται με τις ακόλουθες συμβάσεις (αντίστοιχα με το πρόσημο της τιμής προς κβάντιση)

- Για θετική τιμή εισόδου ανατίθεται το σύμβολο  $i$  το οποίο είναι ο αύξων αριθμός (AA) της ζώνης κβαντισμού. Η αρίθμηση ξεκινά από το 0, το οποίο είναι ο AA της ζώνης κβαντισμού που είναι ξεκινά από το 0 (δηλαδή της ζώνης  $[0, w]$ ), και αυξάνει διαδοχικά για τις ζώνες που βρίσκονται πιο κοντά στο  $+\infty$ .
- Για αρνητική τιμή εισόδου ανατίθεται το σύμβολο  $-i$ , όπου  $i$  είναι ο AA της ζώνης κβαντισμού. Η αρίθμηση ξεκινά από το 1, το οποίο είναι το AA της ζώνης κβαντισμού που είναι πιο κοντά στο 0, και αυξάνει διαδοχικά για τις ζώνες που βρίσκονται πιο κοντά στο  $-\infty$ .

Υπενθυμίζεται ότι το εύρος τιμών της φωτεινότητας είναι στο  $[0, 1]$ .

Στη συνέχεια κατασκευάστε την συνάρτηση

```
1 function x = mydequant(q, w)
```

η οποία υλοποιεί τον αντίστοιχο αποκβαντιστή.

Οι παραπάνω δύο συναρτήσεις θα πρέπει να λειτουργούν σωστά είτε η είσοδος  $x$  είναι βαθμωτή, είτε διάνυσμα στήλης, είτε διάνυσμα γραμμής, είτε διδιάστατος πίνακας (όχι cell). Επίσης, αποφύγετε την χρήση `for` για να μην επιβραδύνετε την υλοποίησή σας. Μπορείτε να χρησιμοποιήσετε κλειστούς τύπους για την υλοποίησή σας.

Τέλος, κατασκευάστε τις συναρτήσεις

```
1 function q = imagequant(x, w1, w2, w3)
```

και

```
1 function x = imagedequant(q, w1, w2, w3)
```

οι οποίες κβαντίζουν και αποκβαντίζουν αντίστοιχα μία τριχρωματική εικόνα  $x$  χρησιμοποιώντας της συναρτήσεις `myquant` και `mydequant` που υλοποιήσατε με πλάτος ζώνης κβαντισμού  $w1$ ,  $w2$  και  $w3$  για τα τρία κανάλια αντίστοιχα. Στην αναφορά, παρουσιάστε την εικόνα που σας δίνεται χρησιμοποιώντας 3 bits ανά χρώμα.

### 3.1 Demo 3

Υλοποιήστε το MATLAB script `demo3.m` στο οποίο να δείχνετε την λειτουργία του κβαντιστή/αποκβαντιστή χρησιμοποιώντας 3 bits ανά χρώμα.

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας.

## 4 Πρότυπο PPM

Στο τελευταίο μέρος θα πρέπει να υλοποιήσετε έναν κωδικοποιητή σύμφωνα με το πρότυπο PPM<sup>1</sup>. Η περιγραφή του προτύπου συνοψίζεται στα εννέα σημεία της ιστοσελίδας. Συγκεκριμένα, το αρχείο περιλαμβάνει ένα header σε μορφή ASCII χαρακτήρων, και το κυρίως κομμάτι το οποίο περιέχει σε διαδική μορφή (binary) τις τιμές φωτεινότητας κάθε pixel. Ένα παράδειγμα φαίνεται στον πίνακα 1.

Το πρώτο τμήμα του αρχείου περιλαμβάνει bytes τα οποία, αν αναγνωσθούν ως ASCII χαρακτήρες, αποτελούν τέσσερις παραμέτρους ακολουθούμενες από έναν χαρακτήρα κενού ή αλλαγή γραμμής. Οι παράμετροι είναι η σταθερή "λέξη" P6, ο αριθμός pixels για το πλάτος της εικόνας, ο αριθμός pixels για το ύψος της εικόνας, και ο αριθμός των στάθμεων κβαντισμού  $K$ . Η παράμετρος  $K$  είναι ακέραιος στο διάστημα  $[0, 65536]$ .

Το υπόλοιπο αρχείο περιλαμβάνει τα σύμβολα κβαντισμού (για την ακρίβεια, τις ακέραιες τιμές στις οποίες έχουν αντιστοιχηθεί τα σύμβολα) των τιμών φωτεινότητας κάθε pixel. Σύμφωνα με την σύμβαση

<sup>1</sup><http://netpbm.sourceforge.net/doc/ppm.html>

της προηγούμενης ενότητας, προκύπτει ότι  $w = 1/K$ , όπου  $w$  είναι το πλάτος της ζώνης κβαντισμού, 1 είναι η τιμή της μέγιστης φωτεινότητας (χωρίς να είναι υποχρεωτικό να εμφανίζεται τέτοια τιμή σε κάθε εικόνα), και  $K$  είναι ο αριθμός των στάθμεων κβαντισμού. Οι ακέραιοι που αντιστοιχούν στα σύμβολα κβαντισμού αποθηκεύονται σε διαδική (raw binary) μορφή, δηλαδή τα bits που θα χρησιμοποιηθούν για την αποθήκευση αντιστοιχούν στη μορφή του μη-προσημασμένου ακεραίου στο διαδικό σύστημα. Αν  $K < 256$  χρησιμοποιείται ένα byte για κάθε ακέραιο, διαφορετικά χρησιμοποιούνται δύο bytes (σε big endian).

Οι τιμές αποθηκεύονται διαδοχικά για τα τρία χρώματα με σειρά κόκκινο, πράσινο και μπλε. Επίσης, οι τριάδες δίνονται ανά γραμμή. Συνεπώς τα πρώτα 3 ή 6 bytes<sup>2</sup> είναι οι τιμές φωτεινότητας RGB για το pixel (1, 1), τα επόμενα 3 ή 6 είναι για το pixel (1, 2), κλπ. Αφού τελειώσει η πρώτη γραμμή pixels της εικόνας, ακολουθούν τα bytes για τη δεύτερη γραμμή, μετά για την τρίτη, κλπ.

Υλοποιήστε τη συνάρτηση

```
1 function saveasppm(x, filename, K)
```

η οποία δέχεται ως είσοδο μία κβαντισμένη εικόνα RGB  $x$  και την αποθηκεύει με το όνομα `filename`. Η παράμετρος  $K = K$  και είναι η μέγιστη τιμή φωτεινότητας που θα μπορούσε να έχει κάποιο pixel, όπως περιγράφηκε παραπάνω. Αν έχετε υλοποιήσει σωστά το πρότυπο, η εντολή `imshow(filename)` θα πρέπει να μπορεί να “διαβάσει” και να εμφανίσει σωστά την εικόνα που αποθηκεύσατε σε μορφή ppm.

Πίνακας 1: Παράδειγμα κωδικοποιημένης εικόνας  $1920 \times 1080$  pixels και 256 σταθμών κβαντισμού κατά PPM σε text viewer.

```
P6 1920 1080 255
\250\62\74\56\206\177\176\104\200\161\27\27\52\62 ...
```

#### 4.1 Demo 4

Υλοποιήστε το MATLAB script `demo4.m` στο οποίο θα παρουσιάζετε *ολόκληρο* το pipeline της εργασίας. Συγκεκριμένα, το demo script

1. Αρχικά μετατρέπει την εικόνα που σας δίνεται στο αρχείο `march.mat` σε RGB
2. Στην συνέχεια, πραγματοποιεί την υποδειγματοληψία της RGB εικόνας από τις αρχικές διαστάσεις της, σε  $N = 150$  και  $N = 200$  και `method = 'linear'`
3. Μετά, κβαντίζει την υποδειγματοληπτημένη RGB εικόνα χρησιμοποιώντας 3 bits ανά χρώμα
4. Τέλος, κωδικοποιεί και θα αποθηκεύει την εικόνα σύμφωνα με το πρότυπο PPM και `filename = 'dip_is_fun.ppm'`

Παρουσιάστε/σχολιάστε τα αποτελέσματα στην αναφορά σας.

### Για 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 κλίσης συναρτήσεων). Επίσης, ο κώδικας θα πρέπει να εκτελείται

<sup>2</sup>Τα 3 ή 6 bytes σχετίζονται με το αν χρησιμοποιείται ένα byte ή δύο για την αναπαράσταση ενός χρώματος, ανάλογα με το αν η μέγιστη τιμή φωτεινότητας είναι μεγαλύτερη από 255.

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 της ομάδας.
- 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.