

# Ψηφιακή Επεξεργασία Εικόνας

## Εργασία 1

Όνομα: Πορλού Χάιδω

AEM: 9372

---

### 1. Φίλτρο Bayer

Στο πρώτο κομμάτι της εργασίας καλούμαστε, χρησιμοποιώντας ένα δωσμένο πίνακα, να κάνουμε μετατροπή μιας αρχικής (bayer) εικόνας σε μία κλασσική rgb.

Αρχικά, όπως φαίνεται και στον κωδικά μου, χωρίζω τον πίνακα σε 3 ξεχωριστούς, έναν για το κάθε χρώμα, πίνακες, αφήνοντας κενές τις θέσεις που δεν αντιστοιχούν σε ένα εκ των τριων χρωμάτων κάθε φορά.

Οι κενές αυτές θέσεις, στη συνέχεια, γεμίζουν με averaging των διπλανών μετρήσεων των αισθητήρων τους, το οποίο γίνεται με τη χρήση **kernels**. Τα kernels αυτά παρουσιάζονται παρακάτω, και αυτά είναι που στην τελική συνελίσσονται με τον κάθε πίνακα (r, g, b). Οι τιμές των kernel υπολογίστηκαν με μελέτη του δωσμένου grid gbrg.

G	B	G	B
R	G	R	G
G	B	G	B
R	G	R	G

$$K_G = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$K_B = K_R = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Τέλος, μετά τη συνέλιξη, ενώνω τους 3 πίνακες χρωμάτων ώστε να έχω την τελική rgb εικόνα, και «κόβω» τις άκρες ώστε η τελική να έχει ακριβώς την ίδια διάσταση (960x1280). Η εικόνα που δημιουργείται φαίνεται να παρουσιάζει τα σωστά χρώματα, γίνεται δηλαδή όντως **averaging**, και είναι η παρακάτω:



---

## 2. Υποδειγματοληψία Εικόνας

Στο δεύτερο κομμάτι της εργασίας καλούμαστε να κάνουμε υποδειγματοληψίας μίας εικόνας με δύο διαφορετικές μεθόδους, την μέθοδο του *nearest neighbor*, κατά την οποία η τελική εικόνα έχει έναν συγκεκριμένο αριθμό πίξελ της αρχικής, και την μέθοδο της *bilinear interpolation* (γραμμική παρεμβολή σε 2 διαστάσεις).

Όσον αφορά τη μέθοδο του **nearest neighbor**, την υλοποίησα χρησιμοποιώντας *scaling* και στη συνέχεια *stroγγυλοποίηση*, ώστε να επιλεγεί το πίξελ που κάθε φορά θα συμμετέχει στην τελική εικόνα. Στην ουσία, δηλαδή, εντοπίζω μέσω του κώδικα, ποιό είναι το κοντινότερο στην επιλεγμένη κάθε φορά θέση πίξελ, και αυτό προσθέτω στην εικόνα/πίνακα εξόδου (για κάθε ένα απο τα τρία χρώματα).

Η δεύτερη μέθοδος, αυτή της **bilinear interpolation**, θεωρητικά θα έχει καλύτερα αποτελέσματα λόγω της ακρίβειας της. Αφορά ουσιαστικά μια γραμμική παρεμβολή σε 2 διαστάσεις, η οποία στον κωδικά μου υπολογίζεται με τον εξής τρόπο:

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}),$$

$$f(x, y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}).$$

$$f(x, y) \approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2)$$

Όπου οι θέσεις  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_1, y_2)$  και  $(x_2, y_1)$  αφορούν τα 4 κοντινότερα πίξελ στην θέση που θέλουμε να τοποθετήσουμε το πίξελ της νέας εικόνας. Αντίστοιχα, τα  $f(Q_{ij})$  είναι οι τιμές της φωτεινότητας των πίξελ αυτών. Κάνοντας **διγραμμική παρεμβολή** με τον παραπάνω τρόπο, βρίσκουμε την τιμή του πίξελ στην θέση  $x, y$  και επαναλαμβάνουμε τη διαδικασία για όλο τον πίνακα.

Στον κώδικα αναλύονται και ειδικές περιπτώσεις όπου ισχύει  $x_1 = x_2$ ,  $y_1 = y_2$  ή και τα δύο, και υπολογίζονται αναλόγως (ώστε να μην προκύπτει πουθενά διαίρεση με το 0 λόγω του παραπάνω τύπου).

Οι εικόνες που προκύπτουν για:

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

παρουσιάζονται παρακάτω.





2

Σε σχετικά μικρές εικόνες όπως οι παραπάνω δε μπορούμε να παρατηρήσουμε πολλές διαφορές. Είναι αρκετά ξεκάθαρο όμως ότι η μέθοδος της διγραμμικής παρεμβολής έχει καλύτερα αποτελέσματα στις ακμές και στις απότομες χρωματικές αλλαγές, όπως στα σημεία που τελειώνει το λουλούδι και αρχίζει το background.

---

### **3. Κβαντισμός και αποκβαντισμός**

Στο 3<sup>ο</sup> κομμάτι της εργασίας καλούμαστε να δημιουργήσουμε έναν ομοιόμορφο κβαντιστή (και έναν αποκβαντιστή), ο οποίος θα λειτουργεί για τις εικόνες μας αλλά και για βαθμωτή είσοδο, καθώς και για διάνυσμα γραμμής ή στήλης.

Αξίζει να σημειωθεί ότι στην περίπτωση μας ο πίνακας/εικόνα παίρνει μόνο θετικές τιμές στο διάστημα  $[0, 1]$ , αλλά ο κβαντιστής, μιας και ομοιόμορφος, δέχεται και το διάστημα  $[-1, 0]$ .

Ο κβαντιστής που δημιουργήσα δέχεται, πέρα από την εικόνα εισόδου, και το μέγεθος της ζώσης κβαντισμού, η οποία καθορίζεται από τον αριθμό των bits που θέλουμε να χρησιμοποιήσουμε για κάθε σύμβολο του κβαντιστή (π.χ. για 3 bits ανα σύμβολο, το μέγεθος ζώσης κβαντισμού είναι  $w = \frac{1}{2^3} = 1/8 = 0.125$ ).

Τα σύμβολα εξόδου του κβαντιστή είναι ακέραιοι θετικοί αριθμοί που ξεκινούν από το 0 για θετικές τιμές εισόδου και ακέραιοι αρνητικοί αριθμοί που ξεκινούν από το -1 για αρνητικές τιμές εισόδου. Συμπληρωματικά, πρέπει να σχολιαστεί ότι ο κβαντιστής είναι mid-rise.

Ο αποκβαντιστής, από την άλλη πλευρά, δέχεται την κβαντισμένη εικόνα και δίνει τις αντίστοιχες τιμές στο νέο πίνακα, δηλαδή στην εικόνα εξόδου. Πρέπει, φυσικά, να έχει δοθεί και ίδιο μέγεθος ζώνης κβαντισμού ώστε να λειτουργήσει ο αποκβαντισμός.

Τέλος, οι συναρτήσεις `imagequant()` και `imagequant()` χρησιμοποιούν τους ήδη υλοποιημένους κβαντιστή και αποκβαντιστή, και επιπροσθέτως δίνουν διαφορετικό μέγεθος ζώνης κβαντισμού για κάθε χρώμα. Το αποτέλεσμα κβαντισμού και αποκβαντισμού της αρχικής μας εικόνας με 3 bits/σύμβολο για κάθε χρώμα παρουσιάζεται παρακάτω:



Αξίζει να σημειωθεί ότι με 3 bits/χρώμα χάνεται μεγάλο μέρος της πληροφορίας. Το αποτέλεσμα θα ήταν πολύ πιο πιστό στην αρχική εικόνα αν είχαμε περισσότερες ζώνες κβαντισμού.



#### 4. Πρότυπο PPM

Στο τελευταίο κομμάτι της εργασίας καλούμαστε να υλοποιήσουμε έναν κωδικοποιητή σύμφωνα με το πρότυπο PPM. Ο κωδικοποιητής αυτός δέχεται σαν είσοδο την κβαντισμένη εικόνα προηγούμενου ερωτήματος, καθώς και τον αριθμό ζωνών κβάντισης, αλλά και το όνομα του αρχείου που θέλουμε να δημιουργήσουμε.

Ο κώδικας μου δημιουργεί το αρχείο αυτό, το οποίο αποτελείται από ένα header με τις απαραίτητες πληροφορίες, καθώς και από τα στοιχεία του πίνακα εικόνας/εισόδου σε raw binary μορφή, για την οποία χρησιμοποιείται η συνάρτηση `fwrite()`. Το αρχείο εξόδου μπορεί να διαβασθεί με οποιοδήποτε PPM viewer είναι διαθέσιμο.

Τα αποτελέσματα/εικόνες εξόδου του demo 4, που αποτελεί ένα pipeline της εργασίας, παρουσιάζονται παρακάτω.



*Εικόνα 150x200 με bilinear downsampling*



*Εικόνα μετά τον κβαντισμό 3 bits/χρώμα και μετατροπή σε ppm*