ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

Ψηφιακή Επεξεργασία Εικόνας - 1η εργασία

Γεώργιος Νικολής

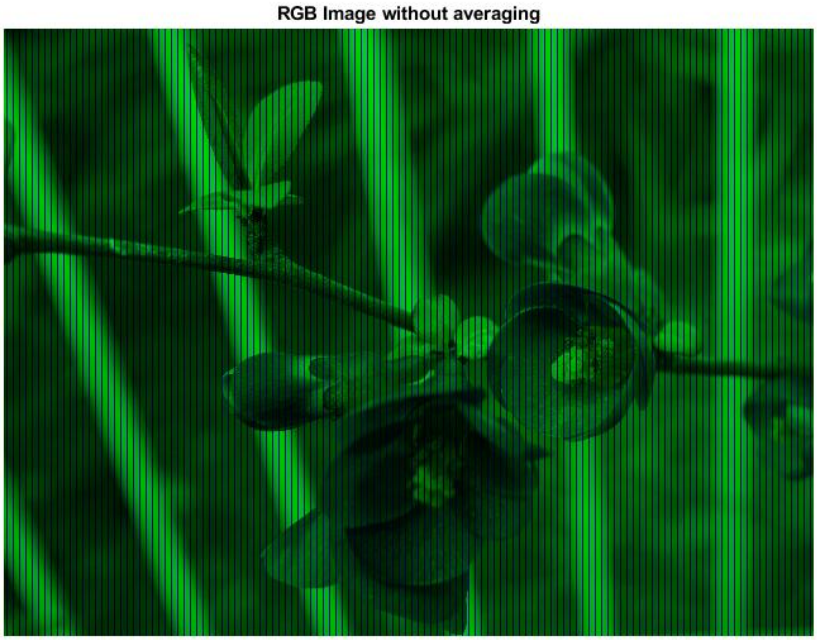
Απρίλιος 2021

Περιεχόμενα

1 Ερώτημα 1

Για το πρώτο ερώτημα αρχικά φορτώνονται τα δεδομένα από το αρχείο της εκφώνησης και στη συνέχεια καλείται η συνάρτηση bayer2rgb ώστε να μετατραπεί η εικόνα σε RGB μορφή.

Μέσα στη συνάρτηση αυτή, αρχικά σπάμε την πληροφορία κάθε χρώματος με βάση το GBRG Sensor Alignment. Μέχρι αυτό το σημείο η εικόνα που έχουμε είναι η παρακάτω.



Στη συνέχεια θα χρησιμοποιήσουμε μια μάσκα που με βάση τα 3χ3 γειτονικά pixel θα γεμίσει το κεντρικό της αντίστοιχα. Το αποτέλεσμα είναι το ακόλουθο.

Υπάρχει μια σαφής βελτίωση στο χρωματισμό της εικόνας, αλλά τα αποτελέσματα δεν είναι ακόμη ικανοποιητικά. Βλέπουμε ότι υπάρχει ένα καρό μοτίβο, το οποίο μας δείχνει ότι τα κενά εικονοστοιχεία δεν έχουν γεμίσει σωστά. Για το λόγο αυτό θα χρησιμοποιήσουμε μια ακόμα μάσκα, η λογική της οποίας πηγάζει από διγραμμική παρεμβολή, που θα δούμε και στα επόμενα ερωτήματα.


```
%Convolvution of colour matrixes with a 3x3 local averaging mask
R=conv2(R, ones(3)./9,'valid');
G=conv2(G, ones(3)./9, 'valid');
B=conv2(B,ones(3)./9,'valid');

%In order to further improve the image
R = conv2(R,[1 2 1; 2 4 2; 1 2 1]/4, 'same');
G = conv2(G, [0 1 0; 1 4 1; 0 1 0]/4, 'same');
B = conv2(B,[1 2 1; 2 4 2; 1 2 1]/4, 'same');
```

Το αποτέλεσμα φαίνεται στην παραπάνω φωτογραφία και πράγματι είναι ικανοποιητικό. Ακόμη, φαίνονται τα 2 βήματα που μας οδηγούν σε αυτό (με μέση τιμή και επιπλέον συνέλιξη)

Αξίζει να παρατηρήσουμε ότι αν κάνουμε αρκετό ζουμ, θα δούμε ότι στα άκρα της η φωτογραφία φαίνεται να έχει ένα μικρό πλάισιο. Αυτό οφείλεται στους όρους valid και same της συνάρτησης conv2 που χρησιμοποιήθηκε, όπου υλοποιούν συνελίξεις μόνο εκεί που υπάρχουν γειτονικά pixel ή βάζουν μηδενικά σε αυτα τα σημεία, αντίστοιχα.

2 Ερώτημα 2

Για το δεύτερο ερώτημα υλοποιήθηκαν 2 υποδειγματοληψίες της εικόνας με τα ορίσματα που ορίζει η εκφώνηση, στο script με όνομα demo2. Για το λόγο αυτό δημιουργήθηκε η μέθοδος myresize, η οποία υλοποιεί υποδειγματοληψία με διγραμμική παρεμβολή αλλά και με τη μέθοδο του κοντινότερου γείτονα.

Στη μέθοδο του κοντινότερου γείτονα, η διαδικασία που ακολουθούμε είναι αρκετά απλή. Απλώς υπολογίζουμε του δείκτες του πίνακα χρωμάτων που θα κρατήσουμε και τους αντιγράφουμε στη νέα εικόνα, που φαίνεται παρακάτω. Οι δείκτες αυτοί προκύπτουν από τους λόγους γραμμών και στηλών του αρχικού και του τελικού μεγέθους της εικόνας. Η διαδικασία αυτή γίνεται για όλη την εικόνα και για κάθε χρώμα ξεχωριστά.

Πράγματι, το νέο μέγεθος της εικόνας είναι το επιθυμητό, ενώ με μια πρώτη ματιά δε φαίνεται κάποια οπτική αλλαγή στην πληροφορία της εικόνας.

Στη μέθοδο της διγραμμικής παρεμβολής, ακολουθούμε διαφορετική τακτική. Αφού και πάλι υπολογίσουμε τους λόγους γραμμών και στηλών, θα σπάσουμε την εικόνα σε μικρά τετραγωνάκια (a,b,c,d) μέσα στα οποιά θα υπολογίσουμε το Pixel που θα κρατήσουμε, ώστε να το αντιγράψουμε στη μικρότερη εικόνα. Ο υπολογισμός του τελικού εικονοστοιχείου που επιλέγεται φαίνεται στη φωτογραφία . Για τα άκρα, αφού δεν μπορούμε να δημιουργήσουμε σίγουρα αυτά τα τετραγωνάκια, κάνουμε την παραδοχή να τα γεμίσουμε με τις τιμές των ακριβώς προηγούμενων, έτσι ώστε να μην αλλοιωθεί οπτικά το αποτέλεσμα. Η διαδικασία αυτή γίνεται για όλη την εικόνα και για κάθε χρώμα.

```
%If we consider a,b,c,d the above square then we
%calculate the interpolated pixel that we will keep

red = a_red*(1 - row_weight)*(1 - column_weight)...
      + b_red*row_weight*(1 - column_weight)+ ...
      c_red*column_weight*(1 - row_weight) + ...
      d_red*row_weight*column_weight;
green =a_green*(1 - row_weight)*(1 - column_weight)...
      +b_green*row_weight*(1 - column_weight)+ ...
      c_green*column_weight*(1 - row_weight)+ ...
      d_green*row_weight*column_weight;
blue =a_blue*(1 - row_weight)*(1 - column_weight)...
      + b_blue*row_weight*(1 - column_weight)+ ...
      c_blue*column_weight*(1 - row_weight)+ ...
      d_blue*row_weight*column_weight;
```

200x300 Image with linear method

Οι δύο μέθοδοι έχουν ικανοποιητική απόδοση, δηλαδή μεταβάλλουν επιτυχημένα το μέγεθος των εικόνων, στο μέγεθος που ζητείται. Σε ότι έχει να κάνει με την ποιότητα της εικόνας, φαίνεται ότι διατηρείται στα ίδια πλαίσια, ίσως λίγο υποβαθμισμένη εάν επιλέξουμε περίεργες αναλογίες μήκους - πλάτους ή πολύ μεγάλο μέγεθος.

3 Ερώτημα 3

Για το ερώτημα αυτό υλοποιήθηκαν οι παρακάτω συναρτήσεις: myquant, myydequant, imagequant, imagedequant. Ακολουθεί μια σύντομη περιγραφή τους.

Η myquant δέχεται ένα διάνυσμα και το κβαντίζει όπως ορίζει η εκφώνηση για τα πλαίσια της εργασίας. Ουσιαστικά χρησιμοποιείται από την imagequant η οποία την καλεί από μια φορά για κάθε χρώμα και δημιουργεί τελικά την κβαντισμένη εικόνα. Αντίστοιχα η myydequant υλοποιεί την αντίστροφη διαδικασία, για έναν πίνακα. Η imagedequant τη χρησιμοποιεί για να αποκβαντίσει την πληροφφορία για κάθε χρώμα. Στο demo3 θα κβαντίσουμε την εικόνα της εκφώνησης και στη συνέχεια θα την αποκβαντίσουμε. Δεδομένου ότι θέλουμε 3 bits για κάθε χρώμα, θα περάσουμε το όρισμα w=8 (στους υπολογισμούς της myquant χρησιμποιηείται το 1/w) με τη λογική ότι θα έχουμε

$$(2^3)^3 = 512$$

συνολικά χρώματα (9-bit RGB). Το ίδιο κάνουμε και για να την αποκβαντίσουμε.

Στην κβαντισμένη εικόνα είναι φανερή η αλλοίωση της ποιότητάς της. Τα χρώματα πλέον είναι πιο φτωχά και οι μεταβάσεις ανάμεσά τους πιο έντονες. Ωστόσο, το περιεχόμενο της εικόνας είναι ακόμα ευδιάκριτο και σε καμία περίπτωση τα αποτελέσματα του κβαντισμού δεν είναι καταστροφικά για την απεικόνηση του λουλουδιού. Στην αποκβαντισμένη εικόνα της επόμενης σελίδας, βλέπουμε ότι δεν έχουμε ανακτήσει πλήρως την αρχική. Αυτό είναι εν μέρη λογικό, καθώς κατά τον κβαντισμό ομαδοποιήσαμε κατάλληλα την αναλογική πληροφορία, κάτι που σημαίνει ότι υπάρχουν χρωματικές τιμές που πετάξαμε. Στην αντίστροφη διαδικασία δεν είναι εφικτό να ανακτήσουμε την πλήρη αυτή αναλογική πληροφορία από την ψηφιακή. Το καλύτερο που μπορούμε να κάνουμε είναι να την αναδημιουργήσουμε έτσι ώστε να την προσωμοιάσουμε, όπως φαίνεται στο αποτέλεσμα. Αξίζει να σημειωθεί, ότι στην αποκβαντισμένη εικόνα έχει γίνει min - max feature scaling τόσο για να είμαστε μέσα στο διάστημα [0,1] όσο και για να είναι πιο ικανοποιητικό οπτικά το αποτέλεσμα.

4 Ερώτημα 4

Η μέθοδος saveasppm αποθηκεύει σε αρχείο ppm την εικόνα που δέχεται σαν όρισμα, όπως ορίζει η εκφώνηση. Ακόμη, στο demo4 παρουσιάζεται το pipeline ολόκληρης της εργασίας, δηλαδή της μετατροπής σε rgb, της υποδειγματοληψίας, του κβαντισμού και τέλος, της αποθήκευσης με το αντίστοιχο όνομα. Με την εκτέλεση αυτού του script θα εμφανιστούν κιόλας, όλες αυτές οι εικόνες που δημιουργήσαμε. Ακολουθούν οι συγκεκριμένες φωτογραφίες.

RGB Image

150x200 Resized Image with linear method

Quantized Image

Saved image

5 Δοκιμές με άλλες εικόνες

Για να επιβεβαιωθεί η ορθή λειτουργία όλων των παραπάνω συναρτήσεων, πραγματοποιήθηκε σύγκριση των αποτελεσμάτων της εικόνας της εκφώνησης με άλλες. Επειδή δεν βρέθηκε κάποια στη .mat μορφή, έγινε η αντίστροφη διαδικασία. Δηλαδή διαβάσαμε μια jpg εικόνα, την αποκβαντίσαμε (ώστε να την περάσουμε στις συναρτήσεις μας), της αλλάξαμε μέγεθος, την κβαντίσαμε ξανά και τελικά την αποθηκεύσαμε. Αξίζει να σημειωθεί, ότι εδώ ο αποκβαντισμός φαίνεται ότι είναι σωστός (μετατρέπονται σωστά οι τιμές χωρίς οπτικές αλλαγές), δεδομένου ότι η αρχική εικόνα δεν είναι αποτέλεσμα υποδειγματολειψίας.

Resized Image with linear method

Quantized Image

Saved image "happyeaster.ppm"