

# Ψηφιακή Επεξεργασία Εικόνας

## Εργασία 1: Από τον RGB αισθητήρα στη μνήμη

Δαδόπουλος Μιχαήλ ΑΕΜ 9989

Στην παρούσα αναφορά, παρουσιάζεται ο κώδικας για την επεξεργασία εικόνων RAW και τη μετατροπή τους σε RGB μορφή. Η λειτουργία του κώδικα διασφαλίζεται μέσω της χρήσης συναρτήσεων που επιτελούν τις απαραίτητες μετατροπές και παρεμβολές.

### Περιγραφή Κώδικα

Ο κώδικας που ανέπτυξα περιλαμβάνει αρκετές συναρτήσεις με κυριότερες την `dng2rgb` που είναι υπεύθυνη για την μετατροπή της raw image σε RGB χρησιμοποιώντας αρκετές άλλες επιμέρους συναρτήσεις που υλοποίησα και η `readdng.m` που η χρήση της είναι για την ανάγνωση της raw εικόνας και εξαγωγής των σημαντικότερων στοιχείων της. Αυτές οι 2 συναρτήσεις καλούνται στο `script demo.m` όπου εισάγω τις παραμέτρους στις συναρτήσεις αυτές ανάλογα τα αποτελέσματα που θέλουμε και στην συνέχεια εκτυπώνει τα αντίστοιχα αποτελέσματα.

**Σημείωση:** Λόγω κάποιου προβλήματος στον κώδικα που δεν μπόρεσα να βρω στο οποίο όταν καλούσα την `readdng` επέστρεφε την `rawim` μόνο με μηδενικά έκανα αντιγραφή τον κώδικα της συνάρτησης στο `demo.m` όπου και λειτουργεί.

Η κύρια συνάρτηση `dng2rgb` δέχεται ως είσοδο τις εξής παραμέτρους:

- `rawim`: Ο πίνακας  $M0 \times N0$  με τις τιμές των δειγμάτων της εικόνας σε θέσεις που περιγράφονται από το φίλτρο Bayer.
- `XYZ2Cam`: Μεταβλητή που παράγεται από τη συνάρτηση `readdng.m`.
- `wbcoeffs`: Συντελεστές ισορροπίας του λευκού που παράγεται από την `readdng.m`.
- `bayerType`: Υποκατηγορία του Bayer pattern (BGGR, GBRG, GRBG, RGGB).
- `method`: Μέθοδος παρεμβολής ('nearest' ή 'linear').
- `M, N`: Διαστάσεις της εικόνας εξόδου.

Η συνάρτηση `dng2rgb` επιστρέφει τις εικόνες `Csrgb`, `Clinear`, `Cxyz`, `Ccam` διάστασης  $M \times N$  που αντιστοιχούν στις εικόνες `CsRGB`, `Clinear`, `CXYZ`, `Ccam`.

Επίσης στην `dng2rgb` καλούνται με την σειρά οι παρακάτω συναρτήσεις :

1. `scale_grid(bayerim, M, N)`: Συνάρτηση που κατασκευάζει ένα νέο grid συντεταγμένων  $M \times N$  (για κάθε χρώμα), έτσι ώστε καθένα από τα τέσσερα γωνιακά σημεία του να συμπίπτουν με τα αντίστοιχα γωνιακά σημεία του grid της εικόνας εισόδου και να την κάνει scale στην διάσταση  $M \times N$ .

2. `nearest_neighbor_demosaic(rawim, bayertype)`: Συνάρτηση που υλοποιεί τον αλγόριθμο του κοντινότερου γείτονα (Nearest Neighbor) για την αποκατάσταση των χρωμάτων στην εικόνα `rawim`, βάσει του `bayertype`. Μεταμορφώνει την εικόνα της κάμερας στο 3D colorspace της.
3. `bilinear_demosaic(rawim, bayertype)`: Συνάρτηση που υλοποιεί τον αλγόριθμο διγραμμικής παρεμβολής (Bilinear Interpolation) για την αποκατάσταση των χρωμάτων στην εικόνα `rawim`, βάσει του `bayertype`. Μεταμορφώνει την εικόνα της κάμερας στο 3D colorspace της.
4. `apply_white_balance(rawim, wbc coeffs, bayertype)`: Εφαρμόζει white balance στην εικόνα `rawim`, χρησιμοποιώντας τους κατάλληλους διορθωτικούς συντελεστές `wbc coeffs`. Αυτή η συνάρτηση επιστρέφει την εικόνα `Cwb` μετά την εφαρμογή της ισορροπίας. Αυτό συμβαίνει με την πολλαπλασιαστική εφαρμογή κάθε pixel της εισόδου με το αντίστοιχο συντελεστή λευκού ισορροπημένου (`wbc coeffs`) προκειμένου να επιτευχθεί η απόσβεση των αντικραδασμών των χρωμάτων που προκύπτουν από διαφορές στην επίδραση των χρωμάτων στους αισθητήρες της κάμερας. Η συνάρτηση εφαρμόζει μια μάσκα στην εισαγόμενη εικόνα, ανάλογα με τον τύπο Bayer της κάμερας. Αυτή η μάσκα πολλαπλασιάζεται στον αντίστοιχο αισθητήρα χρώματος και επιτελεί το έργο της λευκού ισορροπημένου φίλτρου στον αλγόριθμο επεξεργασίας εικόνων.
5. `apply_color_matrix(C, matrix)`: Εφαρμόζει έναν πίνακα μετασχηματισμού χρωμάτων (`matrix`) στην εικόνα `C`. Αυτή η συνάρτηση είναι γενικής χρήσης και μπορεί να χρησιμοποιηθεί για τη μετατροπή της εικόνας από ένα χρωματικό προφίλ σε ένα άλλο, όπως από `Cwb` σε `Cxyz` (χρησιμοποιώντας τον πίνακα `inv(XYZ2Cam)`) και από `Cxyz` σε `Csrgb` (χρησιμοποιώντας τον πίνακα `XYZ2sRGB`).
6. `gamma_correction(Clinear)`: Εφαρμόζει τη μη γραμμική διόρθωση στην εικόνα `Clinear`, μετατρέποντας την γραμμική απόκριση των φωτοδιοδών σε μια απόκριση που ανταποκρίνεται καλύτερα στην αντίληψη της φωτεινότητας από τον ανθρώπινο οπτικό αισθητήρα. Αυτή η συνάρτηση επιστρέφει την εικόνα `Csrgb` με τη μη γραμμική διόρθωση εφαρμοσμένη.
7. `rgb2gray(rgb_image)`: που μετατρέπει μια RGB εικόνα σε grayscale χρησιμοποιώντας τα βάρη των καναλιών RGB. Σύμφωνα με την επίσημη τεκμηρίωση του MATLAB (<https://www.mathworks.com/help/matlab/ref/rgb2gray.html>), η μετατροπή RGB σε grayscale τιμές γίνεται με τον ακόλουθο τρόπο:  $0.2989 * R + 0.5870 * G + 0.1140 * B$ . Αυτά είναι τα βάρη που χρησιμοποιούνται επίσης από τη συνάρτηση `rgb2ntsc` του Image Processing Toolbox.

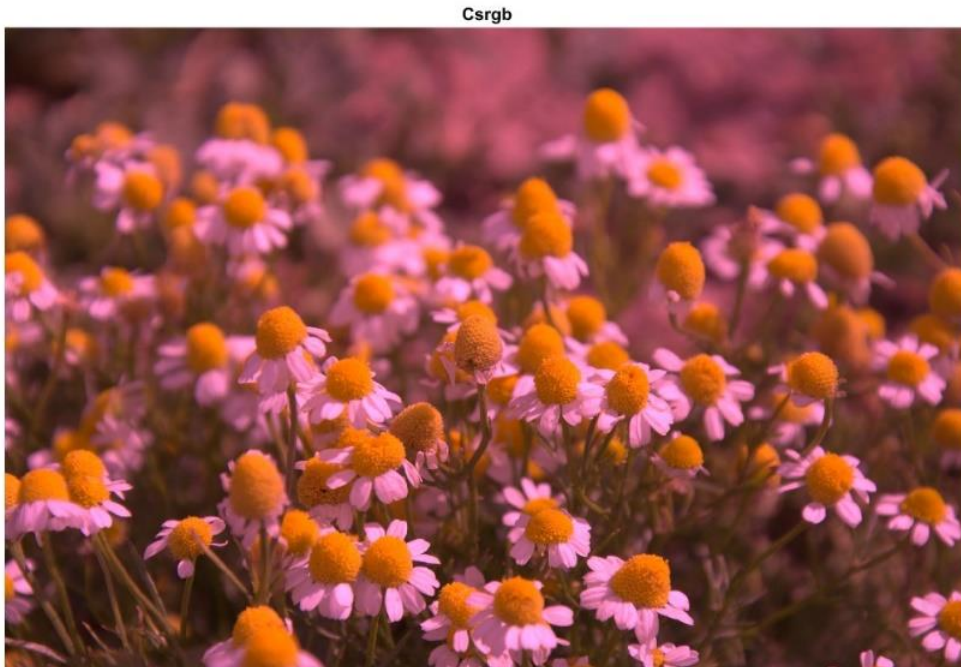
*Στην `dng2rgb` καλούνται όλες αυτές οι συναρτήσεις με την εξής σειρά:*

Πρώτα εφαρμόζεται η `apply_white_balance` στην εικόνα εισόδου με τα κατάλληλα ορίσματα τους διορθωτικούς συντελεστές και τον τύπο `bayer` της εικόνας. Μετά εφαρμόζεται η `scale_grid` για να κάνουμε `scale` την εικόνα στις διαστάσεις εισόδου. Στην υλοποίηση μου το αποτέλεσμα είναι καλό για μεγαλύτερου μεγέθους εικόνες αλλά όχι για μικρότερου, αλλά λόγω χρόνου δεν μπόρεσα να το διορθώσω. Έπειτα ανάλογα τον τύπο `bayer` της εικόνας εφαρμόζονται μία από τις 2 συναρτήσεις, `nearest_neighbor_demosaic` και `bilinear_demosaic` που μεταμορφώνουν την εικόνα της κάμερας στο 3D colorspace της. Στην συνέχεια εκτελείτε 2 φορές η συνάρτηση `apply_color_matrix` για την εξαγωγή των εικόνων `Cxyz` `Clinear` μέσω πολλαπλασιασμού με τους

πίνακες XYZ2sRGB και τον αντίστροφο του XYZ2Cam.Επίσης αργότερα κανονικοποίησα τον αντίστροφο του XYZ2Cam για να βελτιωθούν τα αποτελέσματα(βλέπε εικόνα 1 πριν την κανονικοποίηση).Επίσης μετά την εφαρμογή των συναρτήσεων έκανα scale τις εικόνες στο εύρος τιμών 0-1 που αυτό δεν είχε κάποια διαφορά στο αποτέλεσμα αλλά το έκανα για σιγουριά ότι παραμένουν σε αυτό το εύρος. Έπειτα εφαρμόζω διόρθωση φωτεινότητας όπως είχε και στο paper χρησιμοποιώντας την rgb2gray και βρίσκοντας τη μέση φωτεινότητα της εικόνας, στη συνέχεια την κλιμακώνω έτσι ώστε η μέση φωτεινότητα να είναι κάποια πιο λογική τιμή, κλιμακώνω την εικόνα έτσι ώστε η μέση φωτεινότητα να είναι το 1/4 της μέγιστης. Στην αρχική υλοποίηση δεν εφαρμόζα διόρθωση φωτεινότητας αλλά όταν προσπαθούσα να βελτιώσω τα αποτελέσματα τότε την υλοποίησα με ελαφρώς καλύτερα αποτελέσματα. Αργότερα δοκίμασα και μία άλλη τεχνική διόρθωσης για κάθε κανάλι κλιμακώνοντας την εικόνα ανάλογα με την μέση τιμή των πίξελ κάθε χρώματος. Το αποτέλεσμα ήταν ελαφρώς καλύτερο (εικόνα 4).Ωστόσο δεν έφτασα ποτέ στο τέλειο αποτέλεσμα των παραδειγμάτων. Στην συνέχεια εφαρμόζω και μη γραμμική διόρθωση στην εικόνα με την συνάρτηση gamma\_correction και έχω την τελική rgb εικόνα.

Στην demo.m καλώ πρώτα την συνάρτηση readimg.m και μετά με τα κατάλληλα ορίσματα που επιστρέφει και διαλέγοντας διαστάσεις για την εικόνα εξόδου καλώ την dng2rgb.m. Έπειτα εκτυπώνω όλες τις παραγόμενες εικόνες και τα ιστογράμματα τους.

Αποτέλεσμα για  $M=M_0$ ,  $N=N_0$  και linear method(ίδια αποτελέσματα και για nearest method)χωρίς κανονικοποίηση του αντίστροφου XYZ2Cam και χωρίς διόρθωση φωτεινότητας.



Αποτελέσμα για  $M=M_0$  ,  $N=N_0$  και linear method(ίδια αποτελέσματα και για nearest method) με κανονικοποίηση του αντίστροφου XYZ2Cam αλλά χωρίς διόρθωση φωτεινότητας.

Csrgb



**Αποτελέσματα για  $M=M_0$  ,  $N=N_0$  και linear method(ίδια αποτελέσματα και για nearest method) με κανονικοποίηση του αντίστροφου XYZ2Cam και με διόρθωση φωτεινότητας.**



## Εικόνα Crgb:

Csrgb



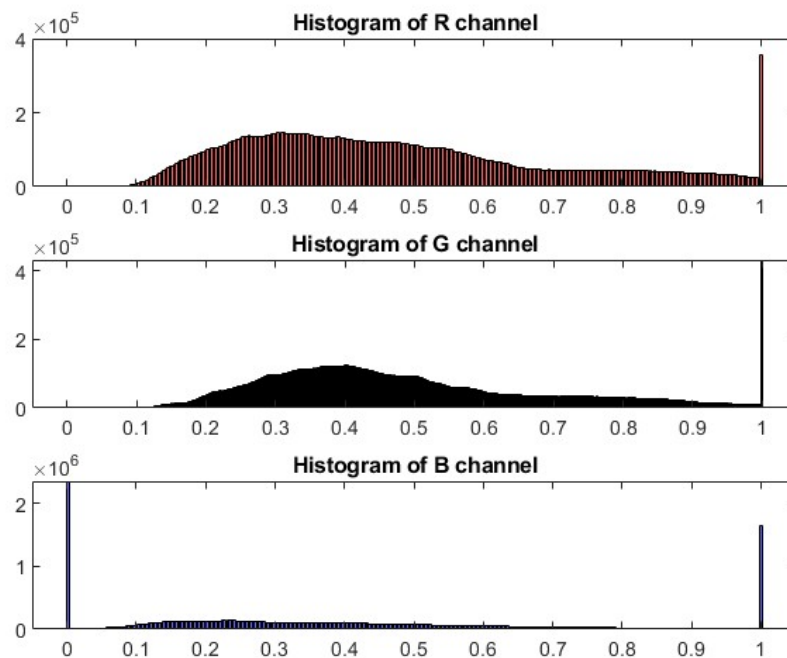
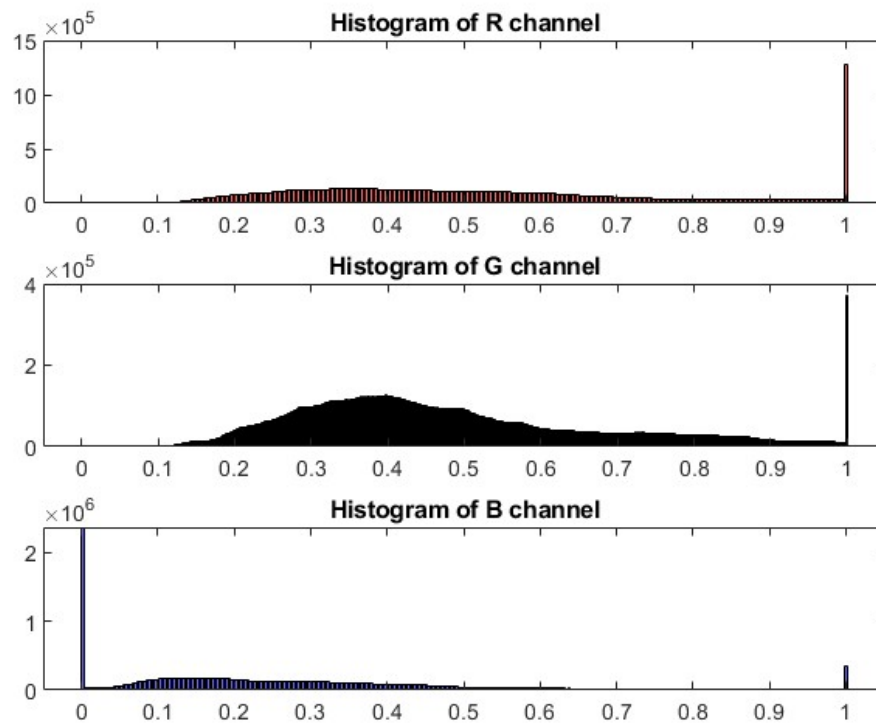
Εικόνα 3, με διορθωση φωτεινότητας όπως στο paper

Csrgb

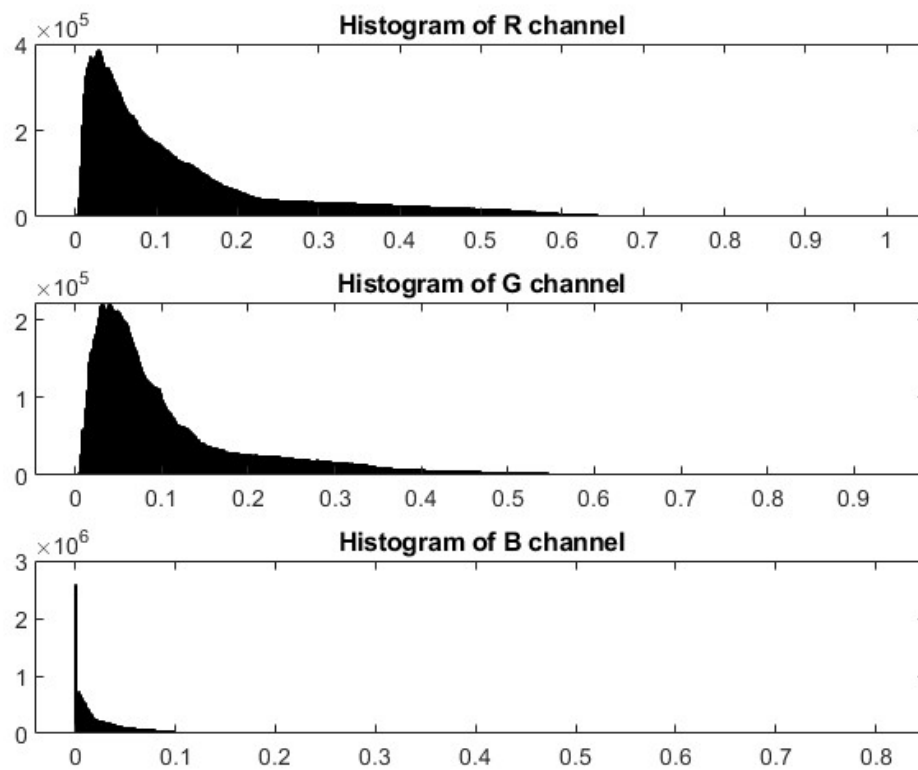
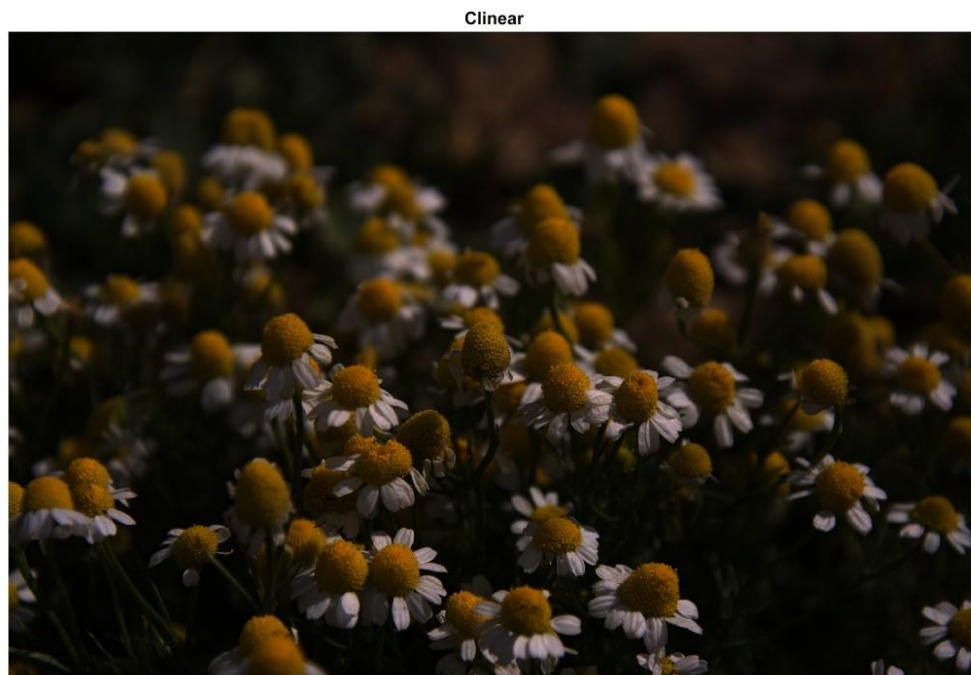


Εικόνα 4, με διαφορετική προσέγγιση διόρθωσης φωτεινότητας

Και τα ιστογράμματα τους αντίστοιχα με την ίδια σειρά:

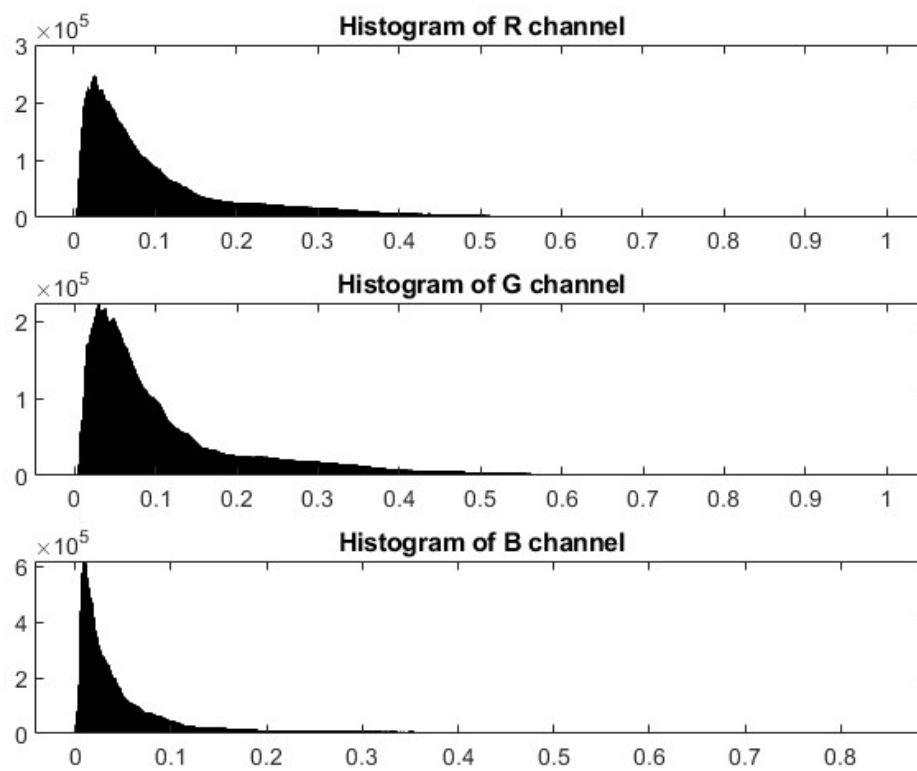


## Εικόνα Clinear και το ιστόγραμμα της:



## Εικόνα Cxyz και το ιστόγραμμα της:

Cxyz





Εικόνα Ccam και το ιστόγραμμα της:

