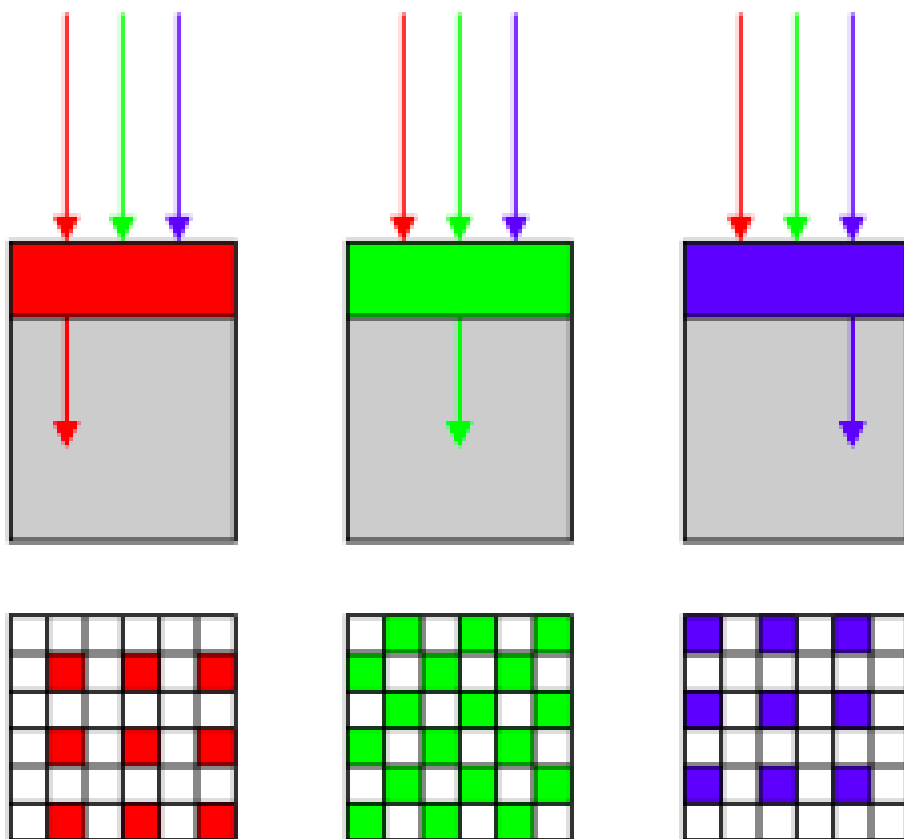


# ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

*ΑΠΟ ΤΟΝ RGB ΑΙΣΘΗΤΗΡΑ ΣΤΗ ΜΝΗΜΗ*



**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:** ΓΕΩΡΓΙΟΣ ΓΚΟΓΚΟΥ

**ΑΕΜ:** 10119

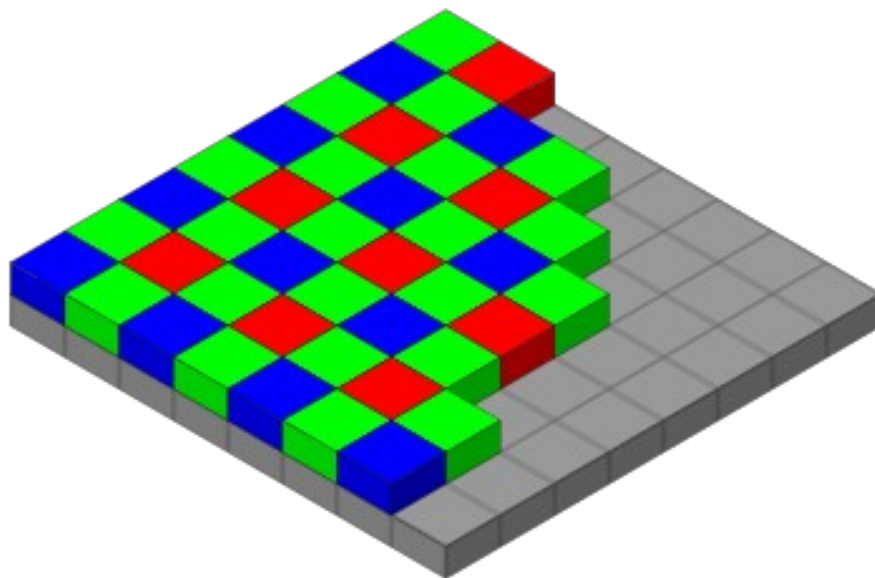
**EMAIL:** ggkogkou@ece.auth.gr

# 1. Εισαγωγή

## 1.1 Θεωρία

Στην εργασία αυτή, ζητείται η μετατροπή RAW εικόνων σε μία sRGB μορφή που είναι κατάλληλη για παρουσίαση σε τυπικές οθόνες, προβολείς κτλ.

Αυτό που συμβαίνει στους αισθητήρες της ψηφιακής κάμερας (π.χ. τύπου CCD ή CMOS) είναι ότι κατά τη φωτογράφιση, απορροφούν το φως που φτάνει στην επιφάνειά τους. Η τοπολογία τους έχει τη μορφή ενός ορθογωνικού πλέγματος, το οποίο αποτελείται από μικρότερα τετράγωνα (pixels). Βέβαια, κάθε pixel δεν είναι αντιπροσωπευτικό για όλα τα χρώματα, και αυτό διότι δεν είναι δυνατόν να γνωρίζουμε τι ποσοστό του προσπίπτοντος φωτός είναι κόκκινο, μπλε και πράσινο. Για αυτό το λόγο τοποθετούμε πάνω από αυτά τα pixels, ένα φίλτρο το οποίο αναλόγως το pixel, αφήνει να περάσει μόνο το επιθυμητό χρώμα (αναλόγως του μήκους κύματος δηλαδή). Φυσικά, η διάταξη αυτή είναι συγκεκριμένη, με πιο δημοφιλή την Bayer. Για παράδειγμα, παρακάτω φαίνεται ένα τέτοιο φίλτρο Bayer:



Στην παραπάνω εικόνα, οι γκρι κύβοι συμβολίζουν τον CMOS αισθητήρα και τα χρωματιστά, το φίλτρο Bayer, το οποίο αφήνει μόνο τη διέλευση του αντίστοιχου χρώματος. Για την παραγωγή της τελικής φωτογραφίας, θα πρέπει, προφανώς, με κάποιο τρόπο να “γεμίσουμε” τα pixel που λείπουν για κάθε χρώμα. Αυτή η διαδικασία καλείται demosaicing και χρησιμοποιούνται τεχνικές παρεμβολής, άλλες πιο απλές και άλλες πιο σύνθετες.

Αξίζει να παρατηρήσουμε ότι ο αριθμός των πράσινων pixel που λαμβάνουμε είναι διπλάσιος από αυτόν των κόκκινων και των μπλε. Αυτό εξηγείται διότι το ανθρώπινο μάτι είναι πιο ευαίσθητο στο πράσινο από ότι στα άλλα δύο, και επειδή η φωτεινότητα εντοπίζεται, κυρίως, στο πράσινο κανάλι.

Η ψηφιακή κάμερα, λοιπόν, αποθηκεύει τα δεδομένα απευθείας σε RAW format αρχεία. Η διαδικασία που πρέπει να ακολουθηθεί, δηλαδή, για την προβολή των εικόνων στον υπολογιστή μας, είναι να διαβάσουμε τα metadata από το RAW αρχείο και έπειτα:

1. Να γραμμικοποιήσουμε την Bayer filtered εικόνα,
2. Να φέρουμε τα Bayer CFA (Color Filter Array) δεδομένα σε ένα επιθυμητό εύρος τιμών, πχ  $[0, 1]$ ,
3. Να εφαρμόσουμε White Balance και να ρυθμίσουμε το Black Level,
4. Να μετατρέψουμε την Bayer CFA εικόνα σε RGB μορφή μέσω της διαδικασίας του demosaicing,
5. Να μετατρέψουμε την τελευταία εικόνα σε όποιο Color Space επιθυμούμε

## 1.2 Κώδικας

Για την υλοποίηση σε κώδικα χρησιμοποιήθηκε MATLAB (2023a). Το τελικό project έχει την εξής μορφή:

```
├── demo.m
├── demosaic_bilinear.m
├── demosaic_nearest_neighbor.m
├── dng2rgb.m
├── readdng.m
├── report.pdf
├── resize_bilinear.m
└── resize_nearest_neighbor.m
```

Περιλαμβάνει τα ζητούμενα αρχεία:

- readdng.m
- dng2rgb.m
- demo.m
- report.pdf

καθώς και βοηθητικά αρχεία με ομώνυμες συναρτήσεις:

- *demosaic\_bilinear.m*: υλοποιείται το demosaicing με bilinear μέθοδο παρεμβολής
- *demosaic\_nearest\_neighbor.m*: υλοποιείται το demosaicing με παρεμβολή nearest neighbor
- *resize\_bilinear.m*: υλοποιείται resizing με bilinear μέθοδο παρεμβολής
- *resize\_nearest\_neighbor.m*: υλοποιείται resizing με nearest neighbor μέθοδο παρεμβολής

Η γενική ιδέα της διαδικασίας που ακολουθήθηκε είναι ότι:

1. Αρχικά, διαβάζουμε όλα τα χρήσιμα metadata της RAW εικόνας με την συνάρτηση readdng()
2. Εφαρμόζουμε White Balancing στην Bayer CFA εικόνα, χρησιμοποιώντας τους συντελεστές που βρήκαμε στα metadata
3. Εφαρμόζουμε demosaicing με Bilinear ή Nearest Neighbor μέθοδο παρεμβολής, και λαμβάνουμε την πρώτη rgb εικόνα που βρίσκεται στο Linear Camera Color Space
4. Εφαρμόζουμε resizing στην εικόνα, ώστε να την φέρουμε στις διαστάσεις MxN που δίνονται ως ορίσματα στην dng2rgb().
5. Μετατρέπουμε την εικόνα σε οποιοδήποτε Color Space επιθυμούμε

Τα βήματα 3-4, ίσως, θα μπορούσαν να γίνουν μαζί αλλά η λογική του resizing σε ήδη demosaiced εικόνα είναι αρκετά πιο απλή καθώς υπάρχουν πολύ λιγότερες υποπεριπτώσεις υπολογισμού των pixel κατά την παρεμβολή (ουσιαστικά μόνο να μη βγει εκτός ορίων πίνακα) και εφόσον κάθε pixel έχει συντελεστή R, G, B είναι πολύ απλό να παρεμβάλλουμε όλα αυτά τα layers μαζί.

Οι συναρτήσεις για demosaicing και resizing έχουν πολυπλοκότητα  $O(n^2)$ .

## 2. Αποτελέσματα

Στο αρχείο demo.m υπάρχει ένα script που εξετάζει διάφορες περιπτώσεις επεξεργασίας της εικόνας με χρήση bilinear και nearest neighbor παρεμβολή.

Πριν παρουσιαστούν τα αποτελέσματα, πρέπει να βρούμε τι bayer CFA διάταξη χρησιμοποιείται. Τρέχοντας τον αλγόριθμο για κάθε ένα από τα Bayer Patterns, χωρίς resizing, λαμβάνουμε τις εξής εικόνες:

RGGB Pattern



BGGR Pattern



GRBG Pattern



Συμπεραίνουμε, λοιπόν, ότι το Bayer CFA φίλτρο στην κάμερα είναι τύπου RGGB. Από εδώ και στο εξής ο αλγόριθμος έχει εκτελεστεί με αυτό το Pattern.

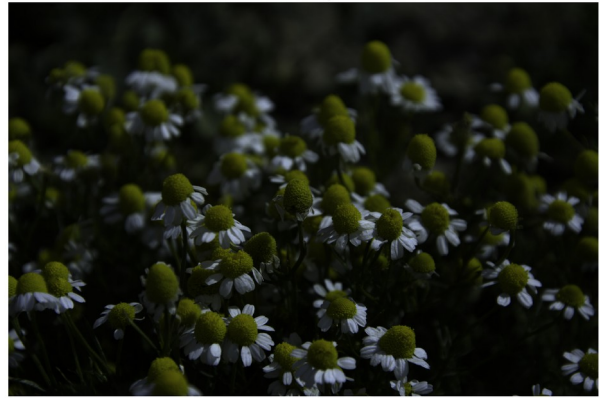
## 2.1 Bilinear Παρεμβολή

Αρχικά, θα παρουσιαστούν οι εικόνες που παράγονται κατά την εκτέλεση του αλγορίθμου για μέγεθος ίσο με το αρχικό της  $M_0 \times N_0$ .

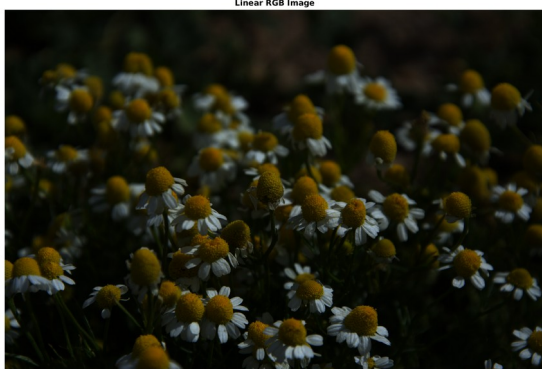
Linear Camera Space



CIE XYZ Space



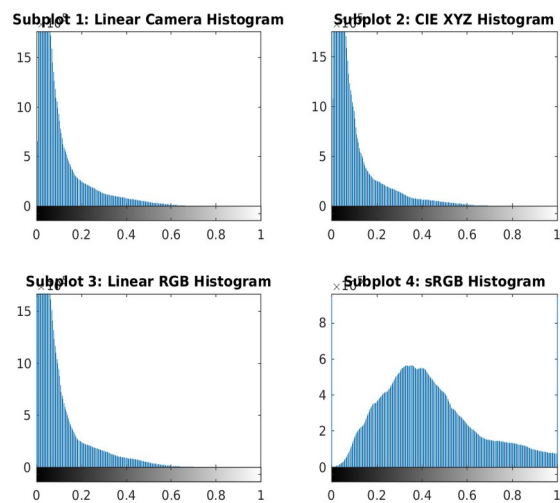
Linear RGB Space



sRGB



Histograms

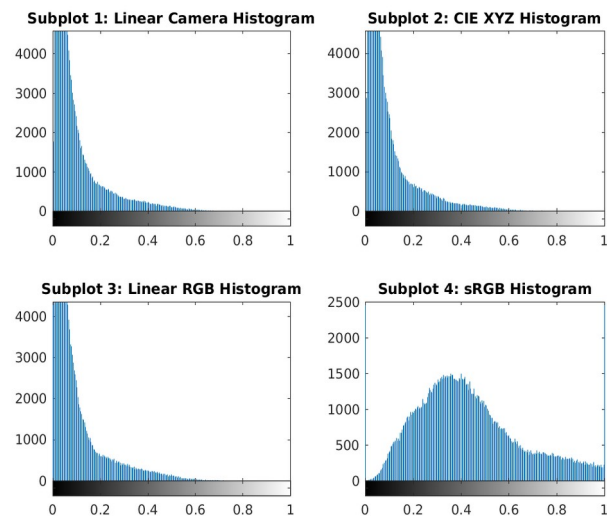


Όπως φαίνεται και από τα ιστογράμματα των εικόνων οι τρεις πρώτες αν και φαίνεται τι απεικονίζουν, τα πάντα είναι πολύ σκοτεινά. Με την εφαρμογή του μη-γραμμικού ΜΣ (gamma correction) στην γραμμική RGB εικόνα, γτιάχνεται η φωτεινότητα και τα χρώματα και το ιστόγραμμα απλώνεται πιο ομαλά σε όλο το εύρος.



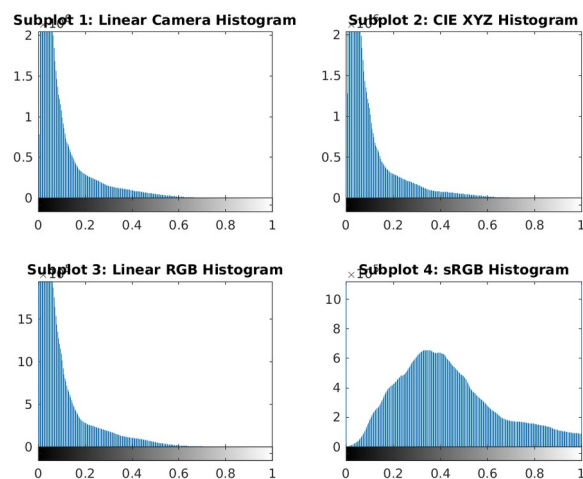
Στη συνέχεια, εξετάζεται η περίπτωση όπου στην `dnv2rgb()` οι είσοδοι  $M$ ,  $N$  είναι μικρότερες από τα  $M_0, N_0$ . Αυτό οδηγεί την εικόνα σε σμίκρυνση, επομένως, στόχος είναι ο αλγόριθμος να υποδειγματοληπτεί την αρχική εικόνα, και η επιλεγμένη μέθοδος παρεμβολής να εξομαλύνει τις διαφορές. Η εικόνα 250x280 είναι η εξής:

**sRGB Image**

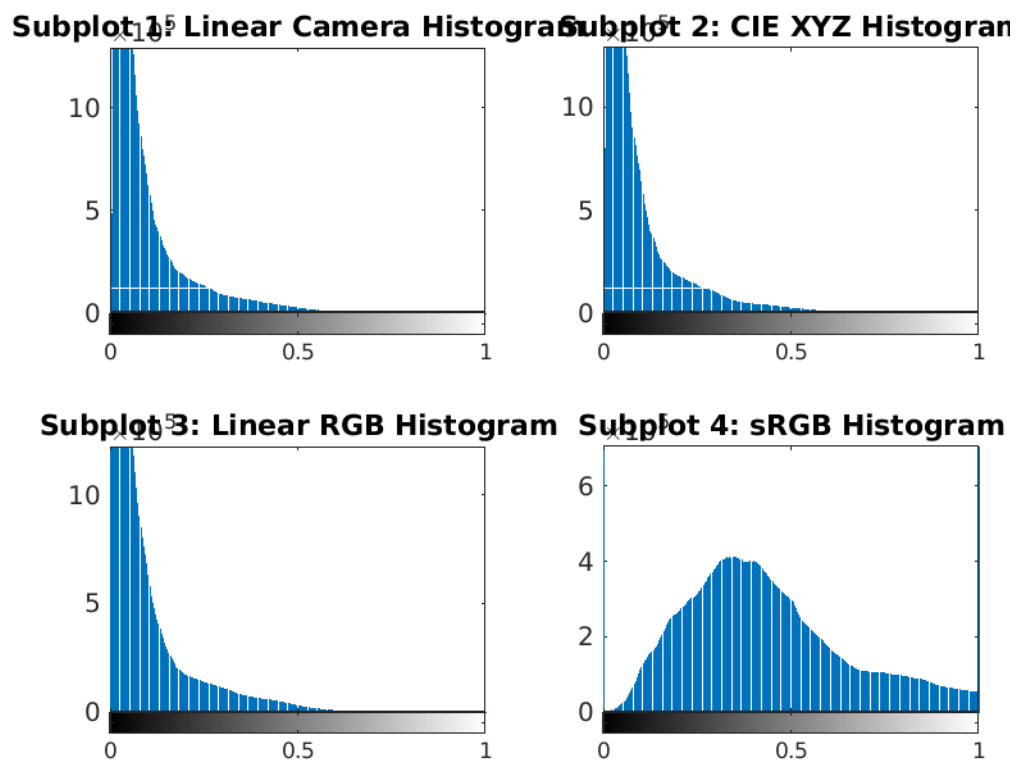


Όπως είναι εύκολα αντιληπτό μπορεί να φανεί η απώλεια πληροφορίας καθώς τα pixels είναι πιο “ορατά” και οι άκρες των αντικειμένων, π.χ. των λουλουδιών, φαίνονται παραποιημένες.

Έπειτα, θα ελεγχθεί η περίπτωση όπου τα  $M$ ,  $N$  είναι μεγαλύτερα από τα  $M_0, N_0$ . Εδώ, αναμένουμε η αρχική εικόνα να δειγματοληπτηθεί και τα κενά που θα προκύψουν στην νέα εικόνα μεγαλύτερων διαστάσεων να καλυφθούν με μέσω της παρεμβολής. Το αποτέλεσμα για διαστάσεις εισόδου 4500x6500 είναι:



Φυσικά, δεν φαίνεται κάποια διαφορά στην εικόνα αφού γίνεται αυτόματα ξανά resize για να χωρέσει στην αναφορά, παρ'όλα αυτά η λειτουργικότητα του αλγορίθμου επιδεικνύεται καλύτερα στην επόμενη περίπτωση όπου  $M = 4400, N = 4000$ .



sRGB Image



Παρατηρούμε, ότι η εικόνα έχει γίνει πετυχημένα resized και η bilinear μέθοδος παρεμβολής έχει πολύ καλή επίδοση, καθώς δεν παρατηρούνται περίεργα φαινόμενα.

Το αρνητικό της bilinear παρεμβολής, βέβαια, έγκειται στο ότι είναι υπολογιστικά πιο ακριβή από την Nearest Neighbor, αλλά όπως θα φανεί έχει και καλύτερα αποτελέσματα.

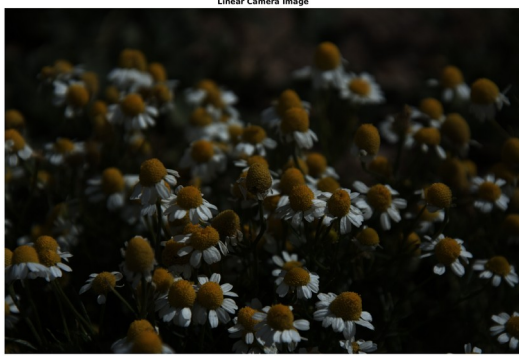
Πρόκειται, δηλαδή, για ένα trade- off ταχύτητας-ποιότητας.



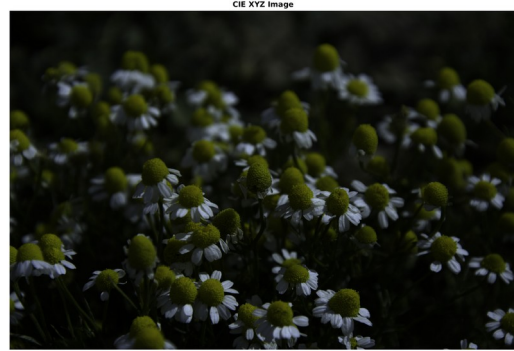
## 2.2 Nearest Neighbor Παρεμβολή

Όπως και προηγουμένως, παρουσιάζονται πρώτα οι εικόνες που παράγονται κατά την εκτέλεση του αλγορίθμου για μέγεθος ίσο με το αρχικό της  $M_0 \times N_0$ .

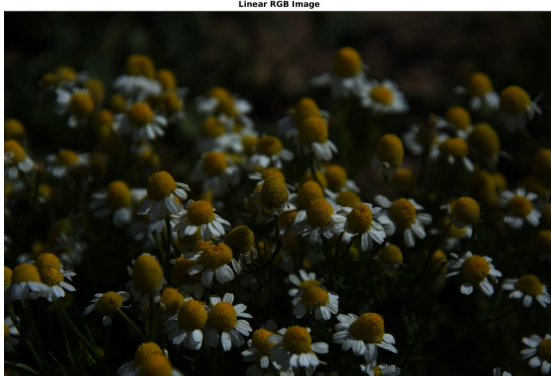
Linear Camera Space



CIE XYZ Space



Linear RGB Space

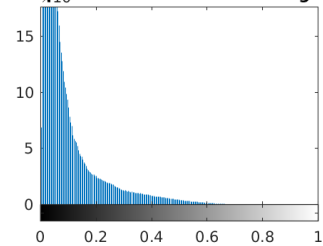


sRGB

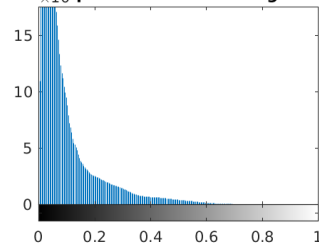


Histograms

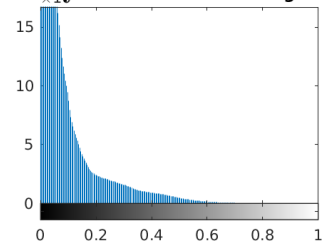
Subplot 1: Linear Camera Histogram



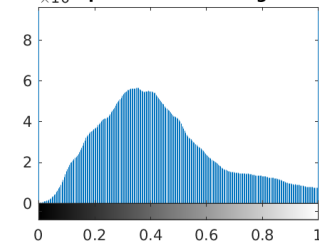
Subplot 2: CIE XYZ Histogram



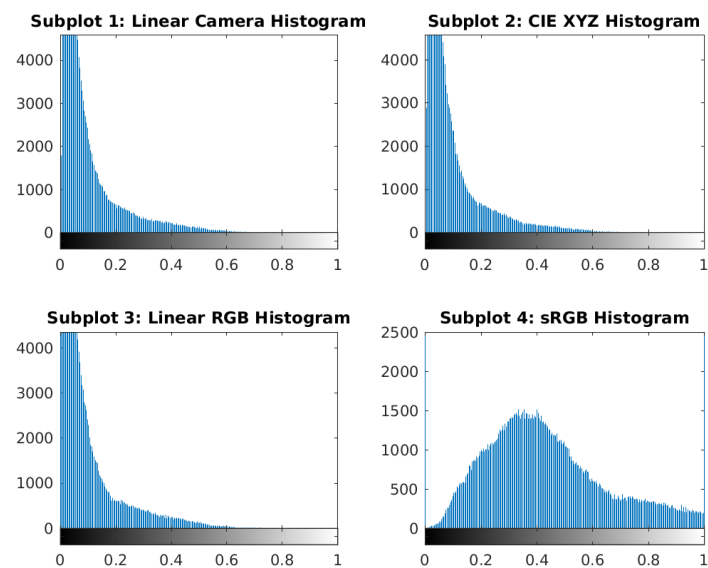
Subplot 3: Linear RGB Histogram



Subplot 4: sRGB Histogram



Τα αποτελέσματα, όπως φαίνεται, είναι ικανοποιητικά. Ενδιαφέρον όμως έχει η περίπτωση της σμίκρυνσης της εικόνας.

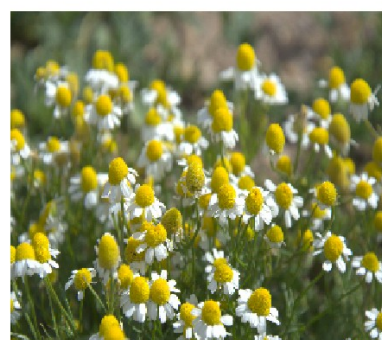


Για σύγκριση της bilinear και της nearest neighbor παρατίθενται και μαζί:

Bilinear



Nearest Neighbor



Επαληθεύεται ότι η bilinear πετυχαίνει ένα καλύτερο smoothing και αντιμετωπίζει καλύτερα την απώλεια πληροφορίας στην οποία υπόκειται. Η nearest neighbor από την άλλη είναι, όντως, αρκετά πιο γρήγορη.

Τέλος, παρουσιάζεται η εικόνα όπου  $M=4400, N=4000$ , δηλαδή το ύψος είναι μεγαλύτερο από της original εικόνας αλλά το πλάτος είναι αρκετά μικρότερο.



Αυτό που παρατηρούμε και εδώ είναι ότι εάν γίνει λίγο zoom στην παραπάνω εικόνα, μπορούν να ξεχωρίσουν τα pixels. Αυτό γίνεται, διότι, η μέθοδος αυτή αντιγράφει τις τιμές των κοντινότερων pixels, άρα ουσιαστικά μεγαλώνει το pixel.

Αν και από απόσταση η εικόνα φαίνεται τέλεια, στην πραγματικότητα η ποιότητα είναι χειρότερη από αυτή που επιτεύχθηκε με την bilinear παρεμβολή καθώς με λίγο zoom τα pixels ξεχωρίζουν.

Εάν, ζητούμενο είναι η ταχύτητα και όχι ιδιαίτερα η πολύ ακριβής απεικόνιση, τότε η μέθοδος αυτή είναι εξαιρετική. Αλλιώς, εάν η λεπτομερής απεικόνιση είναι το κύριο μέλημα, τότε συστήνεται η bilinear.