Ψηφιακή Επεξεργασία Εικόνας
-Εργασία 1-

—

Ανακατασκευή τριχρωματικής εικόνας RGB από εικόνα Bayer
Κβαντισμός των τιμών φωτεινότητας ανά κανάλι &
Δυαδική αποθήκευση εικόνας PPM

Μάστορας Ραφαήλ Ευάγγελος
7918

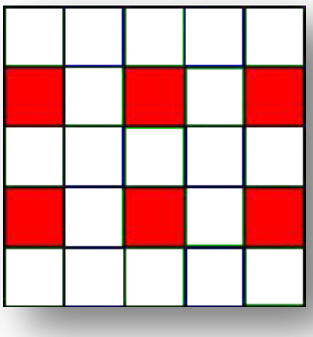
2/4/2017

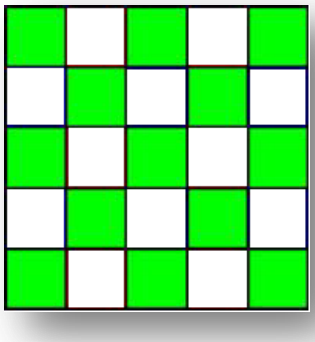
Περιγραφή της εργασίας

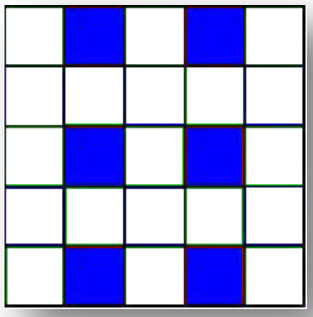
Για την εργασία αυτή μας ζητήθηκε να ανακατασκευάσουμε μια τριχρωματική εικόνα RGB απο δοσμένη εικόνα Bayer, με γνωστή την διάταξη αισθητηρίων της κάμερας. Η ανακατασκευή της εικόνας ζητήθηκε να γίνει με δύο τρόπους παρεμβολής, την μέθοδο του κοντινότερου γείτονα και την μέθοδο της διγραμμικής παρεμβολής. Έπειτα έπρεπε να υλοποιήσουμε κβαντιστή και αποκβαντιστή εικόνας για δοθέν εύρος ζώνης για κάθε χρώμα. Τέλος έπρεπε η κβαντισμένη πλέον εικόνα να μπορεί να αποθηκευτεί σε αρχείο με βάση το πρότυπο PPM.

Ανακατασκευή της εικόνας

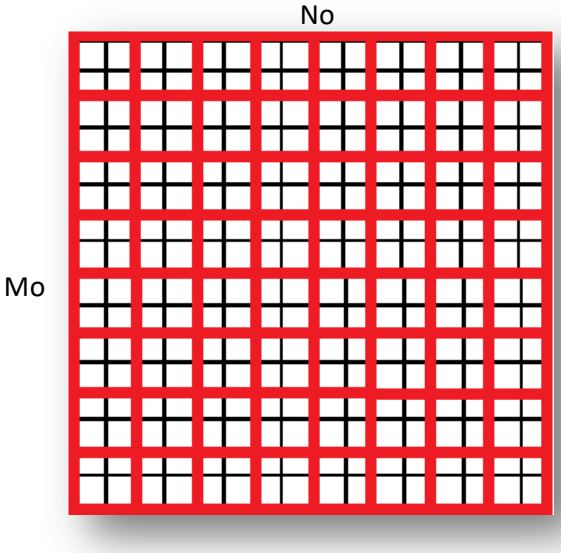
Η ανακατασκευή της εικόνας γίνεται στην συνάρτηση bayer2rgb, η οποία δέχεται σαν ορίσματα τον πίνακα της αρχικής εικόνας, την επιθυμητή ανάλυση της εικόνας που θα εξαχθεί και την μέθοδο παρεμβολής. Αρχικά χρειάζεται να ορίσουμε την διάταξη των αισθητήρων, για τον σκοπό αυτό δημιουργούμε τρείς πίνακες Red,Green και Blue, μεγέθους MoxNo (γραμμές X στήλες), όπου Mo και No η αρχική ανάλυση της δοθέν εικόνας. Σε αυτούς τους πίνακες τοποθετούνται οι τιμές φωτεινότητας της δοθέν εικόνας όπου υπάρχει pixel του ίδιου χρώματος με βάση το Bayer και 0 όταν απουσιάζει η τιμή του. Η μορφή των πινάκων έχει την ακόλοθουθη μορφή:







Έπειτα υπολογίζεται η αναλογία της ανάλυσης της παλιάς εικόνας προς την καινούρια στον κατακόρυφο και οριζόντιο άξονα. Αφού οριστεί ο καινούριος πίνακας της εξόδου με ανάλυση MxNx3, καλείται η συνάρτηση που θα υλοποιήσει την παρεμβολή με την μέθοδο που ζητήθηκε. Και στις δύο συναρτήσεις που θα περιγραφούν παρακάτω, ο πίνακας της αρχικής εικόνας ανατρέχεται με βήμα Mo/M για τις γραμμές και No/N για τις στήλες. Με αυτόν τον τρόπο είναι σαν να έχουμε το grid της καινούριας εικόνας μέσα στο grid της δοθέν εικόνας. Παρακάτω φαίνεται μια οπτικοποίηση αυτού για M=2*Mo και N=2*No. Οι κόκκινες γραμμές ορίζουν τα pixel της αρχικής εικόνας ενώ οι μαύρες γραμμές τα pixel της καινούριας εικόνας. Αντίθετα μια απεικόνιση για M=Mo/2 και N=No/2 οι μαύρες γραμμές θα όριζαν τα pixel της αρχικής εικόνας και οι κόκκινες γραμμές τα pixel της καινούριας.



Παρεμβολή με την μέθοδο του κοντινότερου γείτονα:

Για την μέθοδο αυτή όπως αναφέρθηκε ανατρέχουμε τον πίνακα με βήμα Mo/M για τις γραμμές και No/N για τις στήλες. Οι μεταβλητές indexX και indexY κρατάνε την θέση μας στον καινούριο πίνακα, ενώ η θέση μας στον παλιό πίνακα ορίζεται ως ο μικρότερος κοντινός ακέραιος της τιμής i και j που αυξάνονται με βάση τις παραπάνω αναλογίες και ορίζονται ως bayerI και bayerJ αντίστοιχα. Έπειτα για κάθε pixel της καινούριας εικόνας ελέγχουμε μια γειτονιά 2x2 για κοντινά pixel σε κάθε χρώμα μέσω τον πινάκων Red,Green και Blue, που όπως είπα έχουν στις θέσεις που δόθηκαν, σύμφωνα με το πρότυπο, τις τιμές φωτεινότητας της αρχικής εικόνας και οπουδήποτε αλλού μηδέν . Την πρώτη φορά που βρίσκουμε, για παράδειγμα, ένα κόκκινο pixel σε αυτή την γειτονιά, θέτουμε την ελάχιστη απόσταση ίση με την ευκλείδεια απόσταση του pixel της αρχικής εικόνας που ελέγχουμε με το pixel της καινούριας εικόνας στο οποίο βρισκόμαστε, δηλαδή $distRed = \sqrt{(i - \text{bayerI})\text{^}2 + (\text{j} - \text{bayerJ})\text{^}2}$. Επίσης θέτουμε την τιμή φωτεινότητας της καινούριας εικόνας για το κόκκινο χρώμα ίση με την τιμή φωτεινότητας του κόκκινου pixel που μόλις ελέγξαμε. Έπειτα κάθε φορά που βρίσκουμε ένα καινούριο κόκκινο pixel σε αυτή την γειτονιά ανανεώνουμε την ελάχιστη απόσταση, εφόσον είναι μικρότερη, και εν συνέχεια και την τιμή φωτεινότητας. Η παραπάνω διαδικασία ακολουθείτε και για τα άλλα δύο χρώματα. Όταν ολοκληρωθεί ο καινούριος μας πίνακας έχει τιμή για κάθε pixel ίση με την τιμή του κοντινότερου γείτονα του ίδιου χρώματα του αρχικού πίνακα bayer που μας δόθηκε. Όπως γίνεται αντιληπτό η φράση κοντινότερος γείτονας αναφέρεται στην φυσική απόσταση των pixel μεταξύ της καινούριας και της αρχικής εικόνας.

Παρεμβολή με την μέθοδο της διγραμμικής παρεμβολής:

Αρχικά για την μέθοδο της διγραμμικής παρεμβολής χρειάζεται να κάνουμε padding τους πίνακες Red,Green και Blue, προκειμένου να αποφύγουμε επιπλέον περιπτώσεις στα όρια του πίνακα. Με το padding εννοώ ότι προσθέτουμε από δύο επιπλέον γραμμές στο πάνω και στο κάτω μέρος του πίνακα καθώς και από δύο επιπλέον στήλες αριστερά και δεξιά του πίνακα. Οι νέες αυτές γραμμές παίρνουν τις τιμές των ορίων του πίνακα, δηλαδή η πρώτη γραμμή παίρνει την πρώτη γραμμή και 2 δεύτερη γραμμή την δεύτερη του παλιού πίνακα, αντίστοιχα και για τις στήλες. Τώρα μέσα στην συνάρτηση, ανατρέχουμε τον πίνακα με βήμα Mo/M για τις γραμμές και No/N για τις στήλες. Για κάθε θέση του νέου πίνακα ελέγχουμε σε τι χρώμα pixel βρισκόμαστε στον παλιό πίνακα καθώς και σε τι χρώμα γραμμή βρισκόμαστε. Συγκεκριμένα αν βρισκόμαστε σε πράσινο pixel υπάρχουν δύο περιπτώσεις, δεξιά και αριστερά μας θα έχουμε είτε μπλε pixel, οπότε βρισκόμαστε σε μπλε γραμμή είτε κόκκινο, οπότε και βρισκόμαστε σε κόκκινη γραμμή. Αφού προσδιορίσουμε το χρώμα της γραμμής και το χρώμα του pixel στο οποίο βρισκόμαστε διακρίνουμε περιπτώσεις. Αν είμαστε σε πράσινο pixel και κόκκινη γραμμή τότε έχουμε πάνω και κάτω μπλε pixel , ενώ τα επόμενα κοντινά μπλε pixel βρίσκονται είτε 2 στήλες αριστερά είτε 2 στήλες δεξιά. Προκειμένου να διαλέξουμε ελέγχουμε την απόσταση απο την δεύτερη στήλη από τα αριστερά και την δεύτερη στήλη από τα δεξιά, και διαλέγουμε εκείνη με την μικρότερη απόσταση. Έτσι έχουμε 4 γειτονικά μπλε pixel για να κάνουμε διγραμμική παρεμβολή. Αρχικά υπολογίζουμε την παρεμβολή στον y άξονα με τους τύπους :

$$I1 = Blue(x1, y1) * (y2 - j)/(y2 - y1) + Blue(x1, y2) * (j - y1)/(y2 - y1)$$
$$I2 = Blue(x2, y1) * (y2 - j)/(y2 - y1) + Blue(x2, y2) * (j - y1)/(y2 - y1)$$

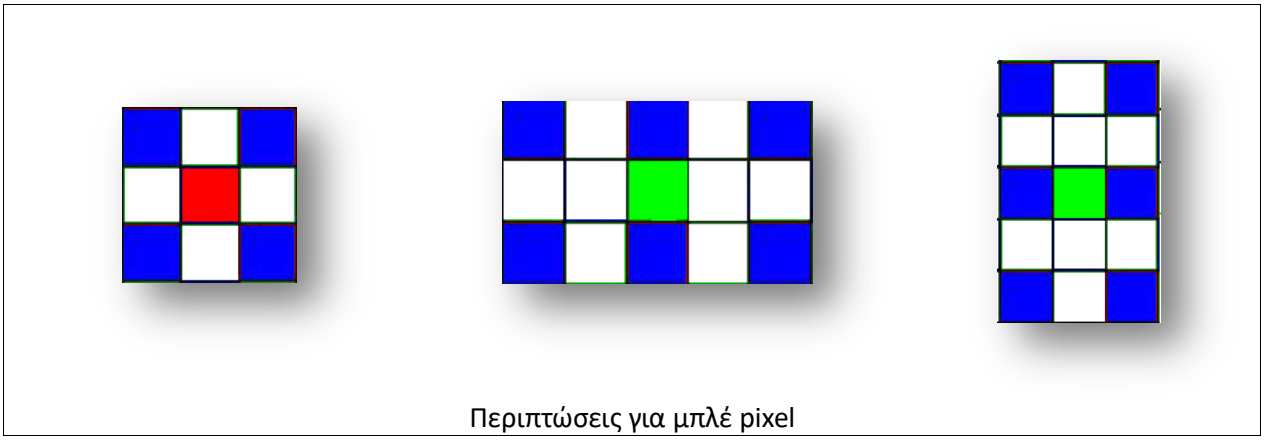
και έπειτα την παρεμβολή στον x άξονα με τον τύπο :

$$I = I1 * (x2 - i)/(x2 - x1) + I2 * (i - x1)/(x2 - x1)$$

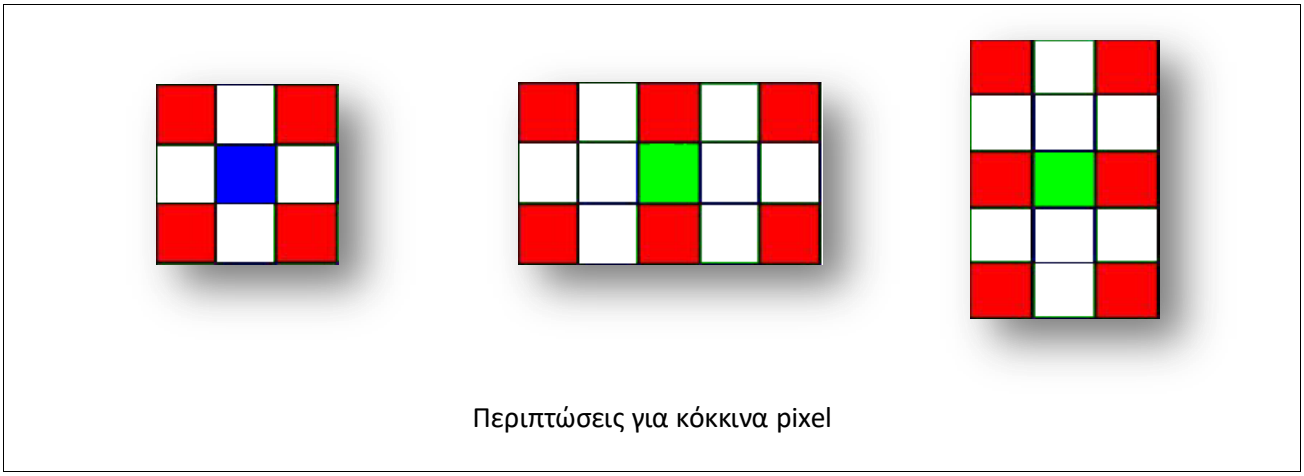
Όπου (x1,y1) η θέση του πάνω αριστερά pixel στον παλιό πίνακα, (x1,y2) η θέση του πάνω δεξιά, (x2,y1) η θέση του κάτω αριστερά και (x2,y2) η θέση του κάτω δεξιά.

Τέλος η τιμή I που βρήκαμε αποθηκεύεται για το μπλε χρώμα στη θέση που εξετάζουμε του καινούριο πίνακα. Βέβαια το χρώμα του pixel μέσα στο οποίο βρισκόμαστε δεν χρειάζεται να βρεθεί με παρεμβολή , αλλά αντιθέτως το παίρνουμε κατευθείαν από την τιμή φωτεινότητας του pixel αυτού στον αρχικό πίνακα.

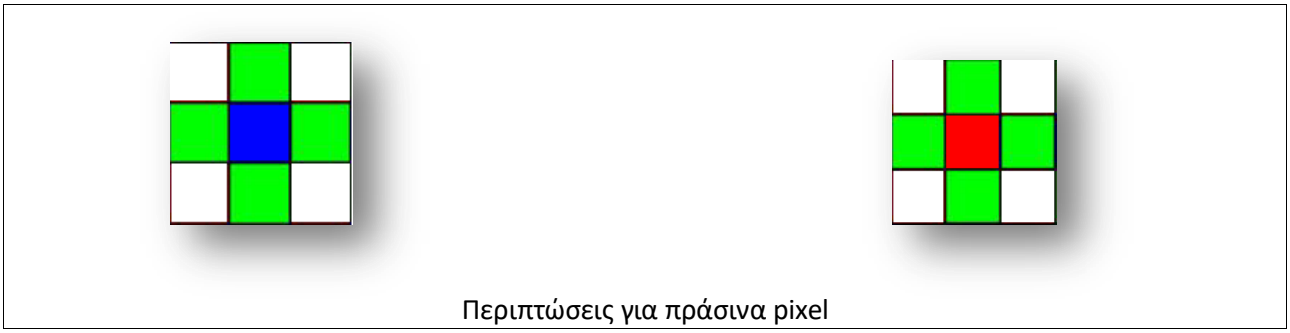
Αντίστοιχη διαδικασία ακολουθείται για όλα τα χρώματα και για τις υπόλοιπες περιπτώσεις που φαίνονται παρακάτω.



Περιπτώσεις για μπλέ pixel



Περιπτώσεις για κόκκινα pixel

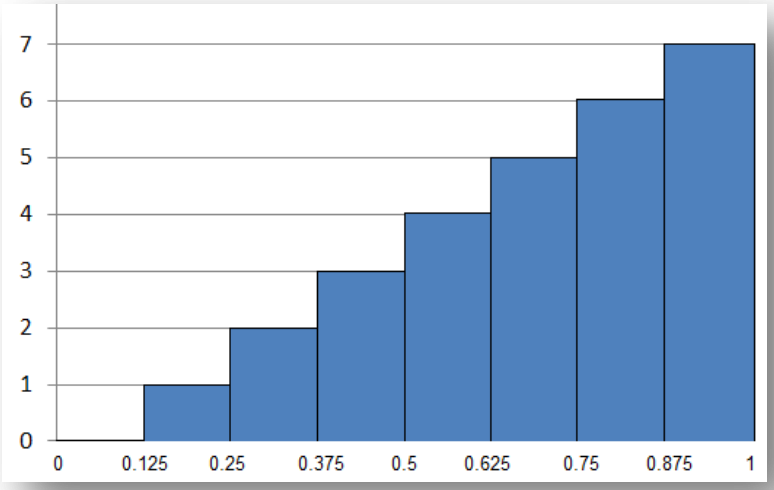


Περιπτώσεις για πράσινα pixel

Ο λόγος που για την διγραμμική παρεμβολή πήρα περιπτώσεις αντί να ανατρέχω σε μια γειτονιά προκειμένου να βρω τα γειτονικά pixel είναι ότι αυτή η γειτονιά θα έπρεπε να είναι 5x5 και ο χρόνος που έκανε το πρόγραμμα για να τρέξει για όλα τα σημεία του πίνακα ήταν πολύ μεγάλος. Αντιθέτως με αυτήν την υλοποίηση ο αλγόριθμος τρέχει πολύ πιο γρήγορα.

Κβάντιση και αποκβάντιση της εικόνας

Μετά την ανακατασκευή της αρχικής Bayer εικόνας σε τριχρωματική RGB εικόνα, ζητήθηκε να δημιουργήσουμε κβαντιστή και αποκβαντιστή που δουλεύει για διαφορετικό πλάτος ζώνης κβαντισμού σε κάθε κανάλι χρώματος. Η λειτουργία του κβαντιστή για 3 bit φαίνεται στο παρακάτω διάγραμμα.



Χωρίζει δηλαδή το διάστημα [0,1] σε 8 ζώνες και κάθε μία ζώνη την αντιστοιχίζει σε μία τιμή στο διάστημα [0,7]. Προκειμένου να υλοποιήσουμε τον κβαντιστή αυτόν το μόνο που χρειάζεται να κάνουμε είναι να διαιρέσουμε όλες τις τιμές των pixel της εικόνας μας με το πλάτος της ζώνης κβάντισης και να πάρουμε τον μικρότερο κοντινό ακέραιο. Επίσης χρειάζεται να πρσέξουμε την περίπτωση που η φωτεινότητα ενός pixel είναι 1, γιατί τότε θα χρειαζόμασταν ένα επιπλέον bit για να αναπαραστήσουμε τον αριθμό αυτό. Δηλαδή έστω για τον παραπάνω κβαντιστή των 3 bit, χωρίζουμε το διάστημα [0,1] σε 8 ζώνες, κάθε ζώνη θα έχει 0.125 πλάτος. Τότε για τιμές φωτεινότητας 1, το σύμβολο του κβαντιστή θα γινόταν 1/0.125=8 και θα χρειαζόμασταν 4 bit για να το αναπαραστήσουμε καθώς το $(111)_2$=7. Για τον λόγο αυτό προσθέτουμε μια απειροελάχιστη ποσότητα στον παρονομαστή κατά την διαίρεση. Η συνάρτηση myquant(x,q) υλοιποιεί τα παραπάνω. Ο χρήστης του προγράμματος το μόνο που επιλέγει είναι τον αριθμό των bit που θα κωδικοποιηθεί κάθε χρώμα. Έπειτα υπολογίζεται ο μέγιστος αριθμός που αναπαριστάται με αυτόν τον αριθμό απο bits, υπολογίζουμε τον αριθμό των σταθμών ως μέγιστος αριθμός+1, λόγο της μηδενικής στάθμης, και μετά υπολογίζουμε τον πίνακα w=1./(αριθμός σταθμών). Τέλος καλείται η imagequant(x,w1,w2,w3) η οποία καλεί την myquant τρεις φορές , μία για κάθε χρώμα.

Ο αποκβαντιστής από την άλλη κάνει την ακριβώς αντίθετη λειτουργία. Παίρνει δηλαδή τις τιμές του διαστήματος [0,7] και σαν έξοδο δίνει τιμές στο διάστημα [0,1]. Για την λειτουργία αυτή η συνάρτηση mydequant(q,w) πολλαπλασιάζει κάθε pixel του πίνακα q με το πλάτος στάθμης κβαντισμού w και προσθέτει την τιμή w/2 καθώς ζητήθηκε η αποκβάντιση να γίνεται στο μισό πλάτος της στάθμης κβαντισμού. Αντίστοιχα η imagedequant(q,w1,w2,w3) καλεί την mydequant τρεις φορές, μια για κάθε χρώμα.

Ο κβαντιστή και ο αποκβαντιστής δουλεύουν για οποιαδήποτε τιμή απο -∞ εώς +∞ και για όσα bit επιθυμούμε.

Αποθήκευση της εικόνας σε δυαδικό αρχείο

Τέλος ζητήθηκε να δημιουργηθεί συνάρτηση που θα αποθήκευε την εικόνα με βάση το πρότυπο PPM. Προκειμένου να το πετύχω αυτό δημιούργησα την συνάρτηση saveasppm(x, filename, K), η οποία λαμβάνει σαν ορίσματα τον κβαντισμένο πίνακα x, το όνομα του αρχείου που θα γίνει η αποθήκευση και την μέγιστη φωτεινότητα , που στην περίπτωση μας είναι ο μέγιστος αριθμός που μπορεί να αναπαρασταθεί με τον αριθμό των bit που επιλέχθηκαν. Μέσα στην συνάρτηση αυτή γίνεται έλεγχος αν η μέγιστη φωτεινότητα ξεπερνάει το 255 ώστε να αλλάξει η κωδικοποίηση των byte σε Big Endian και οι αριθμοί να αποθηκεύονται με 16 bit. Για την αποθήκευση του αρχείου χρειάζεται αρχικά να γράψουμε στην πρώτη γραμμή : P6 N M K και έπειτα σε επόμενη γραμμή να ακολουθήσουν όλες οι τιμές φωτεινότητας με σειρά κόκκινο , πράσινο και μπλέ ανα στήλη για όλες τις γραμμές. Η αποθήκευση των τιμών αυτών έγινε με την συνάρτηση fwrite() και προκειμένου να αποφευχθούν εμφωλευμένες for, που θα καθυστερούσαν πολύ την διαδικάσία, δημιούργησα έναν πίνακα μιας σειράς που περιέχει όλες τις τιμές φωτεινότητας ανα στήλη για κάθε χρώμα με την σειρά RGBRGBRGB....

Σε αυτό το σημείο να αναφέρω πως προκειμένου το αποτέλεσμα της αποθηκευμένης εικόνας να είναι σωστό , χρειάζεται να έχει επιλέξει ο χρήστης ίδιο αριθμό bit κβάντισης για κάθε χρώμα, καθώς στο παρόν πρόγραμμα η saveasppm() παίρνει σαν όρισμα τον πίνακα που εξάγεται απο την imagedequant() και το πρότυπο PPM αποκβαντίζει την εικόνα για ένα ενιαίο πλάτος ζώνης κβαντισμού.

Οπτικά αποτελέσματα

Image produced by *nearest* method

960

1280

Παρεμβολή με την μέθοδο κοντινότερου γείτονα

Image produced by *linear* method

960

1280

Παρεμβολή με την μέθοδο διγραμμικής παρεμβολής

Image produced after dequantatition with RGB = [3 3 3] bits

960

1280

Η εικόνα μετά την αποκβάντιση της για 3 bit σε κάθε χρώμα

Image produced after saving as "*march.ppm*"

960

1280

Μετά την αποθήκευση κβαντισμένης εικόνας με 3 bit για κάθε χρώμα

Image produced after dequantatition with RGB = [8 8 8] bits

960

1280

Η εικόνα μετά την αποκβάντιση της για 8 bit σε κάθε χρώμα

Η εικόνα για διαφορετικές διαστάσεις απο τις αρχικές με μέθοδο παρεμβολής κοντινότερου γείτονα

Η εικόνα για διαφορετικές διαστάσεις απο τις αρχικές με διγραμμική παρεμβολή