Image Processing - HW3

Στην παρούσα εργασία καλείστε να γράψετε ένα πρόγραμμα επεξεργασίας εικόνας. Μπορείτε να φανταστείτε μία εικόνα σαν ένα διδιάστατο πίνακα από εικονοστοιχεία (pixels), μεγέθους όσο και το μέγεθος της εικόνας. Για παράδειγμα, μία εικόνα 712x512 pixels αντιστοιχεί σε ένα διδιάστατο πίνακα από pixels 712 στηλών και 512 γραμμών (συνηθίζουμε να αναφέρουμε πρώτα το πλάτος και μετά το ύψος της εικόνας).

Το interface ce325.hw3.Image

Ένα πρόγραμμα επεξεργασίας εικόνας δίνει την δυνατότητα να εφαρμοστούν κάποιες διαδικασίες επεξεργασίας της εικόνας. Η παρούσα εργασία ορίζει το interface ce325.hw3.Image το οποίο ορίζει τις εξής μεθόδους μεταβολής της εικόνας.

1. **`public void grayscale()`**: μετατρέπει την εικόνα σε ασπρόμαυρη.
2. **`public void doublesize()`**: διπλασιάζει το μέγεθος της εικόνας.
3. **`public void halfsize()`**: υποδιπλασιάζει το μέγεθος της εικόνας.
4. **`public void rotateClockwise()`**: περιστρέφει την εικόνα κατά 90 μοίρες δεξιόστροφα.

Η εικόνα τύπου RGB

Στον τύπο εικόνας RGB, κάθε εικονοστοιχείο απεικονίζει τις τιμές φωτεινότητας για τα τρία βασικά χρώματα κόκκινο, πράσινο και μπλε, (γνωστά και ως RGB από τα αρχικά των λέξεων red, green, blue). Θεωρούμε ότι οι τιμές της φωτεινότητας είναι φυσικοί αριθμοί από 0 έως και MAX_ COLOR (συνήθως η τιμή του MAX_ COLOR είναι 255).

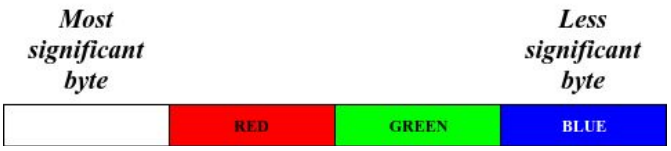
Παραδείγματα RGB τιμών είναι τα εξής:

1. Ένα κόκκινο pixel έχει RGB τιμές 255, 0, 0 (μέγιστη τιμή φωτεινότητας για το κόκκινο και μηδέν για τα υπόλοιπα).
2. Ένα κίτρινο pixel προκύπτει από τις RGB τιμές 255, 255, 0 (το κίτρινο προκύπτει από τη μίξη του κόκκινου με το πράσινο χρώμα)
3. Ένα άσπρο pixel προκύπτει από τις RGB τιμές 255, 255, 255 (μέγιστη φωτεινότητα για όλα τα χρώματα)
4. Ένα μαύρο pixel προκύπτει από τις RGB τιμές 0, 0, 0 (ελάχιστη φωτεινότητα για όλα τα χρώματα)

Μπορείτε να δοκιμάσετε διαφορετικούς χρωματικούς συνδυασμούς εδώ ή με χρήση κάποιου προγράμματος επεξεργασία εικόνας (π.χ. gimp).

Η κλάση ce325.hw3.RGBPixel

Η κλάση RGBPixel περιγράφει ένα pixel που περιέχει την πληροφορία φωτεινότητας για τα τρία βασικά χρώματα κόκκινο, πράσινο και μπλε. Η πληροφορία των τριών χρωμάτων θα πρέπει να αποθηκεύεται σε ένα private πεδίο τύπου int. Παράδειγμα τέτοια αποθήκευσης είναι το παρακάτω σχήμα:



Η κλάση RGBPixel έχει τους εξής κατασκευαστές:

- **`public RGBPixel(short red, short green, short blue)`**: Δημιουργεί ένα pixel με βάση τις τιμές red, green, blue.
- **`public RGBPixel(RGBPixel pixel)`**: Δημιουργεί ένα αντικείμενο που αποτελεί ακριβές αντίγραφο του pixel.

- **`public RGBPixel(YUVPixel pixel)`:** Δημιουργεί ένα RGBPixel από ένα YUVPixel (επεξηγείται παρακάτω).

και τις **public** μεθόδους:

- **`short getRed()`**: επιστρέφει την τιμή του κόκκινου χρώματος.
- **`short getGreen()`**: επιστρέφει την τιμή του πράσινου χρώματος.
- **`short getBlue()`**: επιστρέφει την τιμή του μπλε χρώματος.
- **`void setRed(short red)`**: θέτει την τιμή του κόκκινου χρώματος.
- **`void setGreen(short green)`**: θέτει την τιμή του πράσινου χρώματος.
- **`void setBlue(short blue)`**: θέτει την τιμή του μπλε χρώματος.
- **int getRGB():** Επιστρέφει έναν ακέραιο που περιέχει τα 3 RGB χρώματα, όπως περιγράφεται παραπάνω.
- **void setRGB(int value):** Θέτει τις τιμές των τριών χρωμάτων με βάση τον ακέραιο value. Η ακέραια μεταβλητή value περιέχει τα 3 RGB χρώματα.
- **final void setRGB(short red, short green, short blue):** Θέτει τις τιμές των τριών χρωμάτων με βάση τις τιμές των μεταβλητών **red, green, blue**.
- String toString(): Επιστρέφει ένα αλφαριθμητικό στην μορφή "**`(R,G,B)`**", όπου **R** η τιμή του κόκκινου, **G** η τιμή του πράσινού και **B** η τιμή του μπλε χρώματος.

Η κλάση ce325.hw3.RGBImage

Η κλάση RGBImage υλοποιεί το interface ce325.hw3.Image. Τα εικονοστοιχεία της εικόνας αποτελούνται από αντικείμενα της κλάσης RGBPixel. Η κλάση θα πρέπει να διαθέτει τους εξής κατασκευαστές.

1. **`public RGBImage(int width, int height, int colordepth)`**: Δημιουργεί μία RGB εικόνα με διαστάσεις πλάτος **`width`** και ύψος **`height`** και μέγιστη τιμή φωτεινότητας **`colordepth`**.
2. **`public RGBImage(RGBImage copyImg)`**: Δημιουργεί μία εικόνα RGB από μία άλλη εικόνα RGB. Η νέα εικόνα είναι αντίγραφο της αρχικής (copy constructor).
3. **`public RGBImage(YUVImage YUVImg)`**: Δημιουργεί μία εικόνα RGB από μία εικόνα YUV. Η εικόνα YUV θα εξηγηθεί στη συνέχεια.

Εκτός από τους παραπάνω κατασκευαστές η κλάση RGB θα πρέπει να διαθέτει τις παρακάτω μεθόδους

- **int getWidth():** Επιστρέφει έναν ακέραιο που αντιστοιχεί στην τιμή του πλάτους της εικόνας.
- **int getHeight():** Επιστρέφει έναν ακέραιο που αντιστοιχεί στην τιμή του ύψους της εικόνας.
- **int getColorDepth():** Επιστρέφει έναν ακέραιο που αντιστοιχεί στην τιμή του βάθους χρώματος της εικόνας.
- **RGBPixel getPixel(int row, int col):** επιστρέφει το αντικείμενο τύπου Pixel που αντιστοιχεί στη γραμμή row, col του πίνακα της εικόνας.
- **void setPixel(int row, int col, pixeRGBPixell):** θέτει το αντικείμενο pixel στη γραμμή row, col του πίνακα της εικόνας.

Τέλος η κλάση διαθέτει την public ακέραια σταθερά MAX_COLORDEPTH με τιμή 255.

Επιπλέον, πρέπει να υλοποιεί τις μεθόδους που περιγράφονται από το interface ce325.hw3.Image.

Μετατροπή της εικόνας σε ασπρόμαυρη (μέθοδος grayscale)

Μια ασπρόμαυρη (grayscale) εικόνα είναι μία εικόνα της οποίας οι τρείς RGB τιμές κάθε εικονοστοιχείου έχουν την ίδια τιμή. Για παράδειγμα οι RGB τιμές 0 0 0 αντιπροσωπεύουν το μαύρο χρώμα, οι τιμές 255 255 255 το λευκό και οι τιμές 128 128 128 τη συγκεκριμένη απόχρωση του γκρι.

Γράψτε μία μέθοδο η οποία μετατρέπει τα pixels της εικόνας σε *grayscale* με βάση την παρακάτω φόρμουλα

```
Gray = Red * 0.3 + Green * 0.59 + Blue * 0.11
```

Διπλασιασμός του μεγέθους της εικόνας (μέθοδος doublesize)

Γράψτε μία μέθοδο η οποία διπλασιάζει το μέγεθος της εικόνας. Για τον διπλασιασμό της εικόνας απαιτείται η φωτεινότητα του εικονοστοιχείου στη τυχαία θέση **row, col** να αντιγράφεται στις θέσεις

- `2*row,   2*col,`
- `2*row+1, 2*col,`
- `2*row,   2*col+1`
- `2*row+1, 2*col+1`

Υποδιπλασιασμός του μεγέθους της εικόνας (μέθοδος halfsize)

Γράψτε μία μέθοδο η οποία υποδιπλασιάζει το μέγεθος της εικόνας. Για τον υποδιπλασιασμό της εικόνας απαιτείται η φωτεινότητα του εικονοστοιχείου στη τυχαία θέση **row, col** να προκύψει από τον μέσο όρο της φωτεινότητας των εικονοστοιχείων στις θέσεις

- `2*row,   2*col,`
- `2*row+1, 2*col,`
- `2*row,   2*col+1`
- `2*row+1, 2*col+1`

Περιστροφή δεξιόστροφα κατά 90º (μέθοδος rotateClockwise)

Γράψτε μία μέθοδο η οποία περιστρέφει την εικόνα κατά 90º δεξιόστροφα.

Το format PPM για εικόνες τύπου RGB

Στην παρούσα εργασία θα χρησιμοποιήσετε για διάβασμα από αρχείο και αποθήκευση σε αρχείο έγχρωμες εικόνες που είναι αποθηκευμένες **σε μορφή κειμένου** και έχουν κατάληξη **`.PPM`** (PPM format). Η μορφή ενός αρχείου PPM έχει ως εξής:

1. Ξεκινάει με το αλφαριθμητικό **`P3`**.
2. Ακολουθεί ένας ακέραιος που αντιστοιχεί στο πλάτος της εικόνας (σε εικονοστοιχεία).
3. Ακολουθεί ένας ακέραιος που αντιστοιχεί στο ύψος της εικόνας (σε εικονοστοιχεία).
4. Ακολουθεί ένας ακέραιος που αντιστοιχεί στη μέγιστη τιμή φωτεινότητας της εικόνας.

Τα σημεία 1-4 αποτελούν την κεφαλίδα του αρχείου.

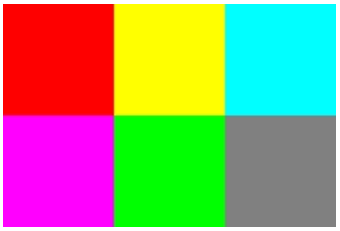
5. Στη συνέχεια για κάθε εικονοστοιχείο εμφανίζονται 3 ακέραιοι για τα τρία RGB χρώματα κόκκινο, πράσινο και μπλε με τη σειρά. Το αρχείο ξεκινά με την πληροφορία του επάνω αριστερού εικονοστοιχείου (θεωρούμε ως πρώτη σειρά εικονοστοιχείων, την κορυφαία σειρά), έπεται το αμέσως δεξιότερο εικονοστοιχείο, μέχρι να φτάσουμε στο δεξιότερο στοιχείο της κορυφαίας σειράς. Στη συνέχεια το αρχείο συνεχίζει από τον αριστερότερο στοιχείο της επόμενης σειράς κ.ο.κ.

Για παράδειγμα, εάν έχουμε μία εικόνα μεγέθους **25x25** εικονοστοιχείων στο αρχείο θα έχουμε αποθηκευμένους **25x25x3** ακεραίους που αντιστοιχούν στην πληροφορία των εικονοστοιχείων της εικόνας. Είναι προφανές ότι η τιμή κάθε ακεραίου δεν πρέπει να υπερβαίνει τη μέγιστη τιμή φωτεινότητας που ορίστηκε στην αρχή του αρχείου (στο σημείο 4).

Οποιοδήποτε αλφαριθμητικό εντός του αρχείου διαχωρίζεται από την υπόλοιπη πληροφορία με έναν ή περισσότερους κενούς χαρακτήρες, χαρακτήρες **tab** ή χαρακτήρες αλλαγής γραμμής ή συνδυασμούς των παραπάνω (whitespace χαρακτήρες).

Παρακάτω δίνεται ένα παράδειγμα εικόνας PPM μεγέθους **2x3** και δίπλα η εικόνα που αντιστοιχεί σε αυτή σε μεγέθυνση. Το παρακάτω περιεχόμενο αντιστοιχεί στο αρχείο κειμένου **test.ppm** που σας παρέχεται έτοιμο για τον έλεγχο του προγράμματος σας.

```
P3
3 2
255
255 0 0  255 255 0  0 255 255
255 0 255  0 255 0  128 128 128
```



Ένα αρχείο PPM μπορείτε να το δείτε σε Linux από οποιοδήποτε πρόγραμμα προβολής ή επεξεργασίας εικόνων (π.χ. gwenview, gimp). Σε Windows μπορείτε να το δείτε μέσω gimp ή μέσω του προγράμματος OpenSeeIt (δεν απαιτεί εγκατάσταση). Εναλλακτικά, μπορείτε να βλέπετε τις φωτογραφίες και online εδώ.

Η κλάση ce325.hw3.UnsupportedFileFormatException

Η κλάση **UnsupportedFileFormatException** είναι απόγονος της κλάσης **java.lang.Exception**. Ένα exception αυτού του τύπου δημιουργείται όταν επιχειρούμε να διαβάσουμε ένα αρχείο εικόνας το οποίο είναι διαφορετικού τύπου από αυτό που επιχειρούμε να διαβάσουμε. Η κλάση διαθέτει τους εξής δύο κατασκευαστές:

- public UnsupportedFileFormatException()
- public UnsupportedFileFormatException(String msg)

Η κλάση ce325.hw3.PPMImage

Η κλάση PPMImage κληρονομεί την κλάση RGBImage. Έχει τους παρακάτω κατασκευαστές

- **`public PPMImage(java.io.File file)`:** Δημιουργεί ένα αντικείμενο της κλάσης λαμβάνοντας ως είσοδο το περιεχόμενο του αρχείου **`file`**. Στην περίπτωση που το αρχείο file δεν υπάρχει ή υπάρχει αλλά δεν μπορεί να το διαβάσει παράγει ένα **java.io.FileNotFoundException**, ενώ στην περίπτωση που το αρχείο file δεν είναι τύπου **PPM** τότε παράγει ένα **ce325.hw3.UnsupportedFileFormatException**.
- **public PPMImage(RGBImage img):** Κατασκευάζει ένα PPMImage από ένα RGBImage.

- **public PPMImage(YUVImage img):** Κατασκευάζει ένα PPMImage από ένα YUVImage.

Σημείωση: Μπορείτε να θεωρήσετε ότι όλες οι εικόνες που θα διαβάζετε θα έχουν μέγιστη τιμή φωτεινότητας 255.

Η κλάση έχει επιπλέον τις παρακάτω **public** μεθόδους:

- **`toString()`**: επιστρέφει ένα String που περιέχει τα περιεχόμενα του αρχείου PPM.
- **`toFile(java.io.File file)`**: Γράφει την εικόνα σε μορφή PPM μέσα στο αρχείο file. Εάν το αρχείο υπάρχει ήδη, διαγράφει το υφιστάμενο περιεχόμενο.

Η διαδικασία photo stacking

Η διαδικασία του stacking εφαρμόζεται σε εικόνες που η διάρκεια λήψης του διαρκεί αρκετά δευτερόλεπτα. Συνήθως είναι βραδινές λήψεις (τοπία, μνημεία ή εικόνες του βραδινού ουρανού από τηλεσκόπια). Η παρατεταμένη έκθεση μιας εικόνας στο φως εισάγει θόρυβο, ο οποίος "θολώνει" το τελικό αποτέλεσμα. Προκειμένου να ξεπεραστεί το πρόβλημα του εισαγόμενου θορύβου εφαρμόζεται η τεχνική του stacking η οποία συνοψίζεται στο εξής:

Λαμβάνονται αρκετές εικόνες του ιδίου θέματος με χρήση της ίδιας παρατεταμένης διάρκειας έκθεσης. Από τις εικόνες αυτές προκύπτει μία μοναδική εικόνα με βάση τον εξής αλγόριθμο: κάθε μία από τις RGB τιμές οποιουδήποτε pixel της τελικής εικόνας προκύπτει από τη μέση τιμή των RGB τιμών του αντίστοιχου pixel, όλων των εμπλεκόμενων εικόνων.

Επεξήγηση: Υποθέτοντας ότι ο θόρυβος ακολουθεί κανονική κατανομή, εάν πάρουμε ένα αρκετά μεγάλο δείγμα εικόνων ο θόρυβος τείνει να εξαφανιστεί, καθώς η μέση τιμή του είναι 0.

Η κλάση ce325.hw3.PPMImageStacker

Η κλάση έχει τον παρακάτω κατασκευαστή:

- **`public PPMImageStacker(java.io.File dir)`**: Λαμβάνει ως είσοδο ένα αρχείο το οποίο πρέπει να είναι κατάλογος. Εάν το αρχείο δεν υπάρχει εμφανίζει το μήνυμα "**`[ERROR] Directory <dirname> does not exist!`**", ενώ εάν το αρχείο υπάρχει αλλά δεν είναι κατάλογος εμφανίζει το μήνυμα "**`[ERROR] <dirname> is not a directory!`**".

Σημείωση: Είναι **υποχρεωτικό** οι εικόνες που διαβάζει η κλάση **PPMImageStacker** να αποθηκεύονται σε μία δομή τύπου λίστας της οποία η κλάση υλοποιεί το interface **`java.util.List`**. Η κλάση της λίστας μπορεί να είναι έτοιμη (από το πακέτο **java.util**).

και τις εξής μεθόδους:

- **`public void stack()`**: Εφαρμόζει την μέθοδο stacking για τις εικόνες που διάβασε.
- **`public PPMImage getStackedImage()`**: Επιστρέφει την εικόνα που προέκυψε από την διαδικασία του stacking.

Κατεβάστε εδώ τις εικόνες για τη διαδικασία του image stacking

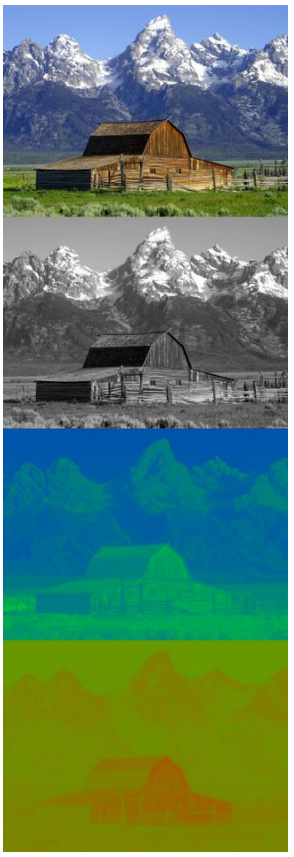
Η εικόνα YUV

Μία εικόνα YUV είναι μία εικόνα της οποίας κάθε pixel περιγράφεται από τις τρεις τιμές ακέραιες τιμές Y, U και V. Η τιμή Y αντιπροσωπεύει την τιμή της φωτεινότητας για το pixel και οι τιμές U, V αντιπροσωπεύουν πληροφορία χρώματος.

Στην διπλανή εικόνα, δίνεται το παράδειγμα μίας έγχρωμης εικόνας και πως αυτή διακρίνεται στις παραμέτρους Y, U και V, οι οποίες ακολουθούν με τη σειρά αμέσως μετά την κορυφαία εικόνα.

Παρατηρήστε ότι η εικόνα Y (2η στη σειρά εικόνα) είναι ασπρόμαυρη καθώς περιγράφει μόνον τη φωτεινότητα κάθε εικονοστοιχείου.

*πηγή εικόνας Wikipedia



Μετατροπή μεταξύ RGB σε YUV και αντίστροφα

Υπάρχει η δυνατότητα μετατροπής ενός pixel από RGB σε YUV και αντίστροφα. Οι συναρτήσεις μετατροπής ενός RGB pixel σε YUV δίνονται παρακάτω:

```
Y = (  66 * R + 129 * G +  25 * B + 128) >> 8) +  16
U = ( -38 * R -  74 * G + 112 * B + 128) >> 8) + 128
V = ( 112 * R -  94 * G -  18 * B + 128) >> 8) + 128
```

Αντίστοιχα, οι συναρτήσεις μετατροπής ενός YUV pixel σε RGB είναι οι εξής:

```
C = Y - 16
D = U - 128
E = V - 128

R = clip(( 298 * C           + 409 * E + 128) >> 8)
G = clip(( 298 * C - 100 * D - 208 * E + 128) >> 8)
B = clip(( 298 * C + 516 * D           + 128) >> 8)
```

Η συνάρτηση `clip` κάνει το εξής: εάν η τιμή εισόδου της είναι αρνητική την κάνει 0, ενώ εάν είναι μεγαλύτερη από 255 την κάνει 255.

Σημείωση: Κατά την μετατροπή YUV σε RGB μπορείτε να υποθέτετε με ασφάλεια ότι η μέγιστη τιμή φωτεινότητας της RGB εικόνας είναι 255.

Μπορείτε να [κατεβάσετε εδώ ένα λογιστικό φύλλο](#) που μετατρέπει τιμές RGB σε YUV και αντίστροφα.

Το format αρχείου YUV

Τα αρχεία YUV έχουν κατάληξη **`.yuv`**. Η μορφή ενός αρχείου YUV έχει ως εξής:

1. Ξεκινάει με το αλφαριθμητικό **`YUV3`**.
2. Ακολουθεί ένα αλφαριθμητικό που αντιπροσωπεύει ένα ακέραιο που αντιστοιχεί στο πλάτος της εικόνας (σε εικονοστοιχεία).
3. Ακολουθεί ένα αλφαριθμητικό που αντιπροσωπεύει ένα ακέραιο που αντιστοιχεί στο ύψος της εικόνας (σε εικονοστοιχεία).

Τα σημεία 1-4 αποτελούν την κεφαλίδα του αρχείου.

4. Στη συνέχεια για κάθε εικονοστοιχείο εμφανίζονται 3 αλφαριθμητικά (ακέραιοι) για τις τρεις Y, U, V τιμές του εικονοστοιχείου. Το αρχείο ξεκινά με την πληροφορία του επάνω αριστερού εικονοστοιχείου (θεωρούμε ως πρώτη σειρά εικονοστοιχείων, την κορυφαία σειρά), έπεται το αμέσως δεξιότερο εικονοστοιχείο, μέχρι να φτάσουμε στο δεξιότερο στοιχείο της κορυφαίας σειράς. Στη συνέχεια το αρχείο συνεχίζει από τον αριστερότερο στοιχείο της επόμενης σειράς κ.ο.κ.

Το format YUV ορίζεται στο πλαίσιο της εργασίας και δεν μπορείτε να το δείτε με τη βοήθεια κάποιου προγράμματος προβολής εικόνων.

[Παραδείγματα YUV εικόνων μπορείτε να βρείτε εδώ.](#)

Η κλάση ce325.hw3.YUVPixel

Η κλάση YUVPixel αντιπροσωπεύει τις τιμές ενός pixel στο σύστημα YUV. **Η πληροφορία των τριών παραμέτρων YUV για κάθε pixel θα πρέπει να αποθηκεύεται σε ένα private πεδίο τύπου int**. Η κλάση έχει τους εξής κατασκευαστές:

- **`public YUVPixel(short Y, short U, short V)`**: Δημιουργεί ένα αντικείμενο της κλάση YUVPixel με βάση τις τιμές Y, U, V.
- **`public YUVPixel(YUVPixel pixel)`:** Δημιουργεί ένα αντικείμενο που αποτελεί ακριβές αντίγραφο του pixel.
- **`public YUVPixel(RGBPixel pixel)`:** Δημιουργεί ένα αντικείμενο YUVPixel από ένα αντικείμενο RGBPixel.

και τις **public** μεθόδους:

- **`short getY()`**: επιστρέφει την τιμή της παραμέτρου Y.
- **`short getU()`**: επιστρέφει την τιμή της παραμέτρου U.
- **`short getV()`**: επιστρέφει την τιμή της παραμέτρου V.
- **`void setY(short Y)`**: θέτει την τιμή της παραμέτρου Y.
- **`void setU(short U)`**: θέτει την τιμή της παραμέτρου U.
- **`void setV(short V)`**: θέτει την τιμή της παραμέτρου V.

Η κλάση ce325.hw3.YUVImage

Η κλάση YUVImage αντιπροσωπεύει μία εικόνα στο σύστημα YUV. Τα εικονοστοιχεία της εικόνας αποτελούνται από αντικείμενα της κλάσης YUVPixel. Η κλάση θα πρέπει να διαθέτει τους εξής κατασκευαστές:

1. **`public YUVImage(int width, int height)`**: Δημιουργεί ένα αντικείμενο τύπου YUVImage με διαστάσεις **`width x height`**. Οι τιμές για τις παραμέτρους Y,U,V ορίζονται ως εξής: **Y=16, U=128, V=128**.
2. **`public YUVImage(YUVImage copyImg)`**: Δημιουργεί ένα αντικείμενο τύπου YUVImage από ένα άλλο αντικείμενο τύπου YUVImage. Η νέα εικόνα είναι αντίγραφο της αρχικής (copy constructor).
3. **`public YUVImage(RGBImage RGBImg)`**: Δημιουργεί ένα αντικείμενο τύπου YUVImage από ένα αντικείμενο τύπου RGBImage.
4. **`public YUVImage(java.io.File file)`**: Δημιουργεί ένα αντικείμενο τύπου YUVImage την πληροφορία του οποίου διαβάζει από το αρχείο **`file`**. Η κωδικοποίηση του αρχείου είναι YUV.
5. Στην περίπτωση που
    a. το αρχείο file δεν υπάρχει παράγει ένα **java.io.FileNotFoundException**.
    b. το αρχείο file δεν είναι τύπου **YUV** τότε παράγει ένα **ce325.hw3.UnsupportedFileFormatException**.

Επιπλέον, έχει τις παρακάτω **public** μεθόδους:

- **`toString()`**: επιστρέφει ένα java.lang.String που περιέχει τα περιεχόμενα του αρχείου σε format YUV.
- **`toFile(java.io.File file)`**:Γράφει την εικόνα σε μορφή YUV μέσα στο αρχείο **`file`**. Εάν το αρχείο υπάρχει ήδη, διαγράφει το υφιστάμενο περιεχόμενο.
- **equalize():** Εξισορροπεί την εικόνα χρησιμοποιώντας τον αλγόριθμο εξισορρόπησης ιστογράμματος που αναφέρεται παρακάτω.

Εξισορρόπηση ιστογράμματος

Τι είναι το ιστόγραμμα μιας εικόνας

Για να κατανοήσετε την έννοια του ιστογράμματος υποθέστε ότι έχετε μία ασπρόμαυρη εικόνα, δηλαδή μία εικόνα που οι τιμές RGB για κάθε pixel είναι ίδιες. Το ιστόγραμμα δημιουργείται εάν σε ένα πίνακα εύρους 256 (από 0 έως MAX_COLOR → 255), τοποθετήσουμε στη θέση **f** τον αριθμό των εικονοστοιχείων της εικόνας με τιμή φωτεινότητας **f**. Το ιστόγραμμα αποτελεί την κατανομή της φωτεινότητας των εικονοστοιχείων της εικόνας.

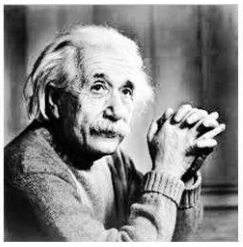
Ιστόγραμμα έγχρωμων εικόνων

Σε εικόνες τύπου RGB το ιστόγραμμα είναι δύσκολο να οριστεί, καθώς και οι τρεις τιμές RGB συμβάλλουν στην φωτεινότητα. Για τον λόγο αυτό προτιμάμε την μετατροπή της εικόνας σε YUV και τον υπολογισμό του ιστογράμματος πάνω στην παράμετρο Y που αποτελεί τον μέτρο της φωτεινότητας της εικόνας.

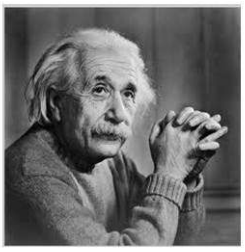
Εξισορρόπησης ιστογράμματος

Συχνά μία εικόνα εμφανίζεται σκοτεινή και συγκεκριμένα τμήματα της είναι δυσδιάκριτα. Η ευκρίνεια της εικόνας μπορεί να βελτιωθεί σημαντικά εάν το ιστόγραμμα διευρυνθεί ώστε να “απλώσει”. Ενδεικτικό είναι το παρακάτω σχήμα.

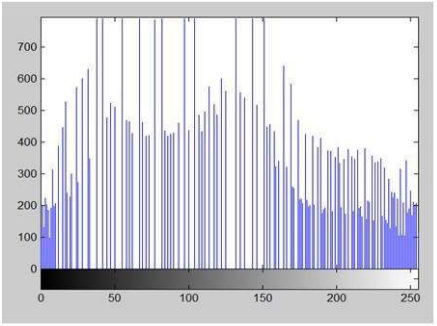
New Image



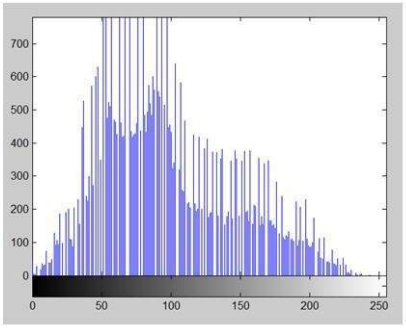
Old image



New Histogram



Old Histogram



Η διαδικασία της εξισορρόπησης του ιστογράμματος έχει ως εξής:

1. Με βάση το ιστόγραμμα, υπολογίζουμε την κατανομή πιθανότητας της φωτεινότητας των εικονοστοιχείων της εικόνας.
2. Με βάση το βήμα 1 υπολογίζουμε την αθροιστική κατανομή πιθανότητας, δηλαδή την πιθανότητα ένα pixel να έχει τιμή φωτεινότητας μικρότερη ή ίση της τιμής X. Η τιμή αποθηκεύεται στη θέση X του πίνακα αθροιστικής πιθανότητας.
3. Επιλέγουμε τη μέγιστη τιμή φωτεινότητας την οποία θέλουμε να έχει η νέα εικόνα. Στο σύστημα YUV, η μέγιστη τιμή φωτεινότητας είναι 235 (δοκιμάστε στο λογιστικό φύλλο να βάλετε τιμές RGB 255, 255, 255 και δείτε ποια είναι η τιμή της παραμέτρου Y). Επιλέξτε ως μέγιστη τιμή φωτεινότητας την τιμή 235.
4. Πολλαπλασιάζουμε τον πίνακα που προκύπτει από το βήμα 2 (αθροιστική κατανομή πιθανότητας) με τη μέγιστη τιμή φωτεινότητας του βήματος 3 και αποθηκεύουμε την τιμή σε έναν πίνακα ακεραίων αποκόπτοντας έτσι το δεκαδικό μέρος του αριθμού που προκύπτει από το γινόμενο.
5. Από τον βήμα 4 προκύπτει η προτεινόμενη μεταβολής της φωτεινότητας. Ας υποθέσουμε ότι στη θέση X του πίνακα ακεραίων που προκύπτει στο βήμα 4 είναι αποθηκευμένη η τιμή Y. Με βάση το παραπάνω, όλα τα εικονοστοιχεία με φωτεινότητα X θα πρέπει να μετατραπούν σε εικονοστοιχεία με φωτεινότητα Y. Μετασχηματίζοντας τη φωτεινότητα για όλα τα εικονοστοιχεία της εικόνας λαμβάνουμε την τελική YUV εικόνα η οποία είναι εξισορροπημένη ως προς το ιστόγραμμα της.

Παρατήρηση: Η εξισορρόπηση ιστογράμματος δεν μεταβάλλει τη χρώματα της εικόνας, παρά μόνο τη φωτεινότητα τους.

Αναλυτικά παραδείγματα για την εξισορρόπηση ιστογράμματος μπορείτε να δείτε εδώ και εδώ.

Η κλάση ce325.hw3.Histogram

Η κλάση ce325.hw3.Histogram δημιουργεί το ιστόγραμμα μιας εικόνας YUV και διαθέτει τις κατάλληλες μεθόδους για την εξισορρόπηση του ιστογράμματος αυτού. Η κλάση έχει τον εξής ένα κατασκευαστή

`public Histogram(YUVImage img)`: Δημιουργεί το ιστόγραμμα μιας εικόνας YUV.

και τις **public** μεθόδους:

- **`public String toString()`**: Εκτυπώνει το ιστόγραμμα σε ένα String ως εξής. Κάθε τιμή φωτεινότητας καταλαμβάνει μία γραμμή. Η γραμμή περιέχει τα εξής:
    a. την αριθμητική τιμή της φωτεινότητας και
    b. τόσους χαρακτήρες '*' όσους αντιστοιχούν (αναλογικά) στα pixels που έχουν τη συγκεκριμένη τιμή φωτεινότητας. Επειδή κάποιες τιμές φωτεινότητας είναι πολύ υψηλές και ξεφεύγουν το μέγιστο εύρος γραμμής, οι τιμές φωτεινότητας κανονικοποιούνται ώστε η μέγιστη τιμή της φωτεινότητας για την εικόνα να περιγράφεται από max 80 χαρακτήρες '*'.
- **`public void toFile(File file)`**: εκτυπώνει το String της μεθόδου toString() σε ένα αρχείο.
- **`public void equalize()`**: Η μέθοδος εξισορροπεί το ιστόγραμμα.
- **`public short getEqualizedLuminocity(int luminocity)`**: Επιστρέφει την εξισορροπημένη τιμή φωτεινότητας που αντιστοιχεί στην δοθείσα τιμή φωτεινότητας **`luminocity`**.

Η μέθοδος equalize της κλάσης YUVImage

Η μέθοδος **equalize** της κλάσης YUVImage δημιουργεί μία νέα εξισορροποιημένη εικόνα σε σχέση με την αρχική εικόνα που περιείχε το αντικείμενο της κλάσης. Η αρχική εικόνα του αντικειμένου αντικαθίσταται από την νέα.

Παραδείγματα εικόνων που έχουν γίνει equalize

Μπορείτε να βρείτε παραδείγματα εικόνων εδώ.

Το κυρίως πρόγραμμα

Η κλάση ce325.hw3.ImageProcessing

Σας παρέχεται έτοιμη η κλάση ce325.hw3.ImageProcessing η οποία υλοποιεί κατάλληλο γραφικό περιβάλλον μέσω του οποίου ο χρήστης έχει τον έλεγχο όλων των λειτουργιών που περιγράφονται παραπάνω. Μετά από κάθε επιλογή του, ο χρήστης βλέπει τη νέα εικόνα που προκύπτει μετά την εφαρμογή της αλλαγής στην αρχική εικόνα.

Το μενού του γραφικού περιβάλλοντος της εφαρμογής έχει ως εξής:

1. File	1.1 Open	1.1.1. PPM File
		1.1.2. YUV File
		1.1.3. Other Format (διαβάζει PNG, JPG κλπ)
	1.2 Save	1.2.1. PPM File

		1.2.2. YUV File
2. Actions	2.1 Grayscale	
	2.2 Increase Size	
	2.3 Decrease Size	
	2.4 Rotate Clockwise	
	2.5 Equalize Histogram	
	2.6 Stacking Algorithm	2.6.2 Select directory

Οι κλάσεις ce325.hw3.PPMFileFilter και ce325.hw3.YUVFileFilter

Σας δίνονται έτοιμες οι παραπάνω κλάσεις.

Οδηγίες Αποστολής

Πακετάρετε μόνο τον κατάλογο **src** στον οποίο έχετε αποθηκεύσει τα αρχεία **.java** μαζί με τους καταλόγους στους οποίους αυτά ανήκουν σε ένα αρχείο με κατάληξη **.zip** ή **.tar.gz** και όνομα τα ΑΕΜ των μελών της ομάδας (π.χ. 1923_1731.zip ή 1625.tar.gz - για ομάδα ενός ατόμου).

Υποβάλετε την εργασία σας στη 3η εργασία του μαθήματος.