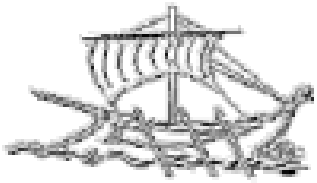


ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ψηφιακή επεξεργασία εικόνας με Visual Studio με την χρήση της βιβλιοθήκης OpenCV

Αδάμ Κωνσταντίνος
Καπετάνιος Αντώνιος

Εισηγητής: Δρ. Ιωάννης Έλληνας, Καθηγητής

ΑΘΗΝΑ
Ιούνιος 2017

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ψηφιακή επεξεργασία εικόνας με Visual Studio με την χρήση της βιβλιοθήκης OpenCV

Αδάμ Κωνσταντίνος Α.Μ. 41796
Καπετάνιος Αντώνιος Α.Μ. 42044

Εισηγητής:

Δρ. Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης:

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι Αδάμ Κωνσταντίνος και Καπετάνιος Αντώνιος,
του Πέτρου και του Παναγιώτη, αντίστοιχα, με αριθμούς μητρώου 41796 και 42044, αντίστοιχα, φοιτητών του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο των συγγραφέων, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι οι συγγραφείς της Π.Ε., οι οποίοι φέρουν και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών των συγγραφέων σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασης της, μετά από αίτηση των ενδιαφερόμενων, τους αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε τις οικογένειές μας για την στήριξη τους σε όλη μας την προσπάθεια, τον επιβλέπον καθηγητή μας, κ. Ιωάννη Έλληνα, για την βοήθεια και την στήριξη του καθ' όλη την πορεία της εκπόνησης της πτυχιακής μας εργασίας.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη αυτόνομων προγραμμάτων για την ψηφιακή επεξεργασία ασπρόμαυρων εικόνων, με την αξιοποίηση των βιβλιοθηκών της OpenCV σε συνδυασμό με το Visual Studio των Windows.

Στο πρώτο κεφάλαιο αναλύεται ο τρόπος εγκατάστασης της OpenCV και του Visual Studio καθώς και η επίλυση τυχών προβλημάτων. Στο δεύτερο κεφάλαιο, παρουσιάζεται ο κώδικας των προγραμμάτων που έχει γραφτεί με την γλώσσα προγραμματισμού C++ και αναλύεται η λειτουργία τους.

ABSTRACT

The present thesis concerns the development of independent programs for digital processing of grayscale images. This can be achieved with the use of the OpenCV libraries combined with Visual Studio for Windows.

The first chapter, concludes the successful installation for OpenCV with Visual Studio and provide solution for any trouble, that might come up. The second chapter, presents the source code of all the programs, which have been compiled with C++.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Προγραμματισμός με C++, OpenCV
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: OpenCV, Visual Studio, C++, Ψηφιακή Επεξεργασία Εικόνας

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

LIB library files

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της πτυχιακής εργασίας και γίνεται μια παρουσίαση στα προβλήματα που παρουσιάστηκαν και επιλύθηκαν επιτυχώς.

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μεμονωμένων προγραμμάτων για την ψηφιακή επεξεργασία εικόνας με την χρήση της βιβλιοθήκης της OpenCV.

1.2 Επιλογή έκδοσης Visual Studio

Η επιτυχής λειτουργία της OpenCV-2.4.9 έγινε με τον συνδυασμό Windows 8.1 Pro με Visual Studio 2013 Professional. Για ποιο καινούργια συστήματα, συστήνουμε Windows 10 Pro με Visual Studio 2017 Enterprise.

1.3 Προβλήματα που εμφανίστηκαν - ασυμβατότητα

Αντιμετωπίσαμε πρόβλημα ασυμβατότητας με την χρήση λειτουργικού συστήματος Windows 10 Pro και Windows 10 Education σε συνδυασμό με Visual Studio 2015 Enterprise και Visual Studio 2013 Professional

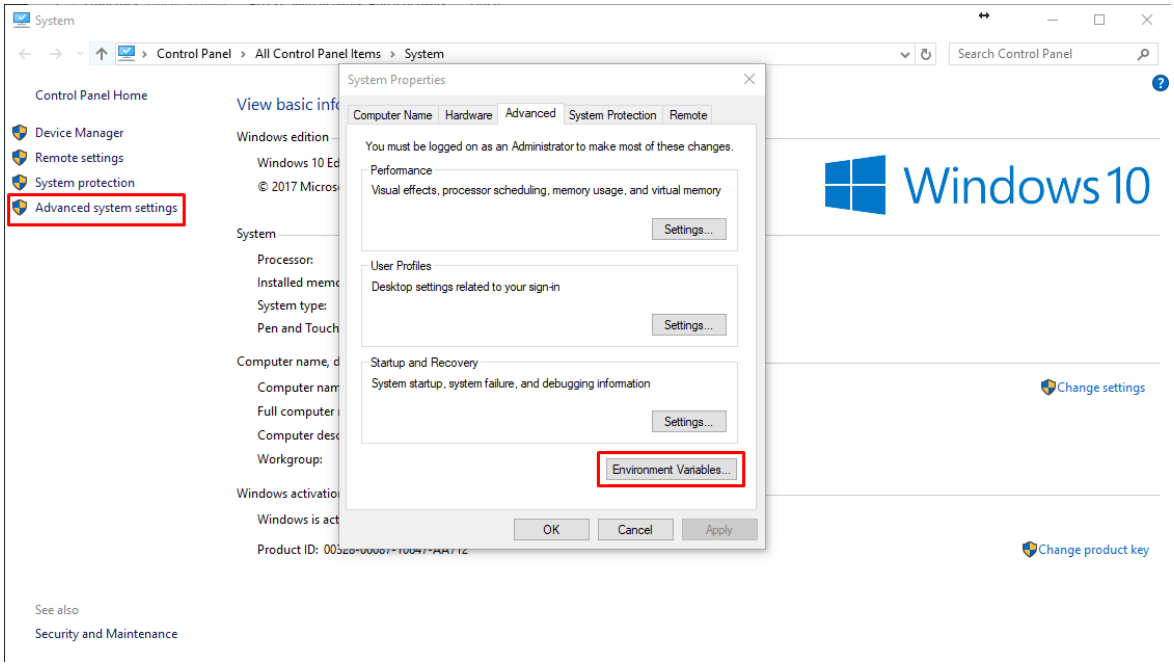
1.4 Η εγκατάσταση της OpenCV – 2.4.9

Έπειτα από την απαραίτητη επανεκκίνηση του υπολογιστή και τις απαραίτητες ενημερώσεις από το Windows Update, ολοκληρώσαμε την εγκατάσταση (αποσυμπίεση) των αρχείων της OpenCV. Επιλέξαμε όλα τα αρχεία και οι συναρτήσεις να βρίσκονται μέσα στον τοπικό δίσκο C. Στη συνέχεια έπρεπε να δηλώσουμε στο λειτουργικό μας σύστημα το path (μονοπάτι) που βρίσκονται οι

συναρτήσεις, προκειμένου να είναι σε θέση να τις αναγνωρίσει και να τις χρησιμοποιήσουμε μέσα από το Visual Studio.

1.5 Δημιουργία μεταβλητών στα windows

Επιλέγουμε Ο Υπολογιστής μου (This PC) και κάνουμε δεξί κλικ -> Ιδιότητες (Properties) -> επιλέγουμε στα αριστερά Ρυθμίσεις για προχωρημένους (Advanced system settings) -> Μεταβλητές συστήματος (Environment Variables)



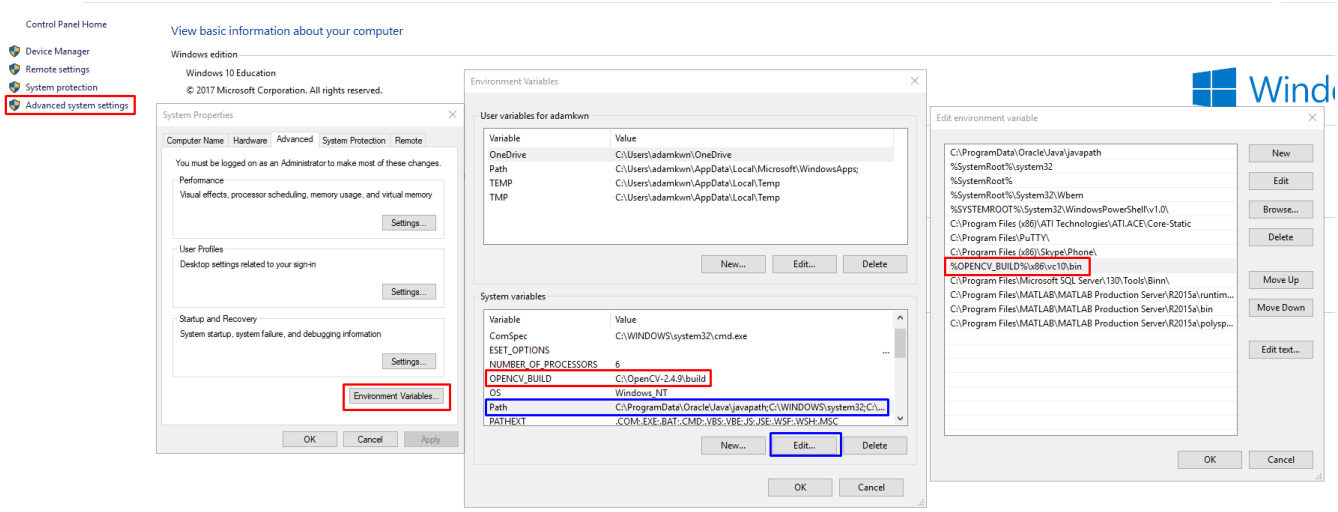
Εικόνα 1.1: Μεταβλητές συστήματος

Επιλέγουμε νέα μεταβλητή συστήματος (προσοχή, όχι μεταβλητές χρήστη) και δίνουμε τις ακόλουθες τιμές:

Όνομα μεταβλητής: OPENCV_BUILD
Τιμή μεταβλητής: C:\OpenCV-2.4.9\build (ή το path στο οποίο έχουμε εξάγει τα αρχεία της OpenCV)

Τέλος, επιλέγουμε την μεταβλητή Path και Edit. Προσθέτουμε στο τέλος των εγγραφών, την ακόλουθη εγγραφή:

%OPENCV_BUILD%\x86\vc10\bin (για 32bit windows) ή
%OPENCV_BUILD%\x64\vc10\bin (για 64bit windows)



Εικόνα 1.2: Εισαγωγή των απαραίτητων μεταβλητών

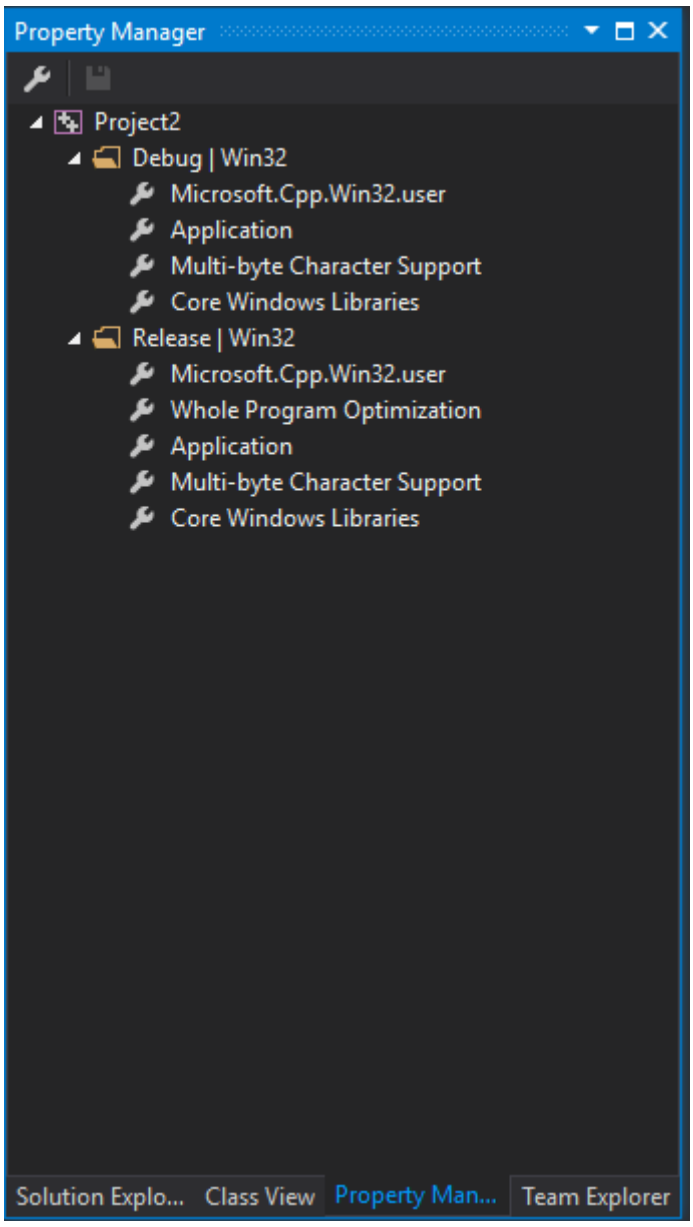
1.6 Ολοκλήρωση εγκατάστασης

Έπειτα από την απαραίτητη επανεκκίνηση προκειμένου τα windows να αναγνωρίσουν τις αλλαγές που πραγματοποιήσαμε και να μπορούμε να χρησιμοποιήσουμε τις συναρτήσεις της OpenCV, ανοίγουμε το Visual Studio ώστε να δημιουργήσουμε τα δύο απαραίτητα αρχεία για την υλοποίηση εφαρμογών με την OpenCV.

1.7 Δημιουργία props αρχείων μέσα από το Visual Studio

Επιλέγουμε το Visual Studio και μόλις ανοίξει, μπορούμε να δημιουργήσουμε το πρώτο πρόγραμμα. Πριν ξεκινήσουμε την συγγραφή του, θα πρέπει να δημιουργήσουμε ένα νέο project. Επιλέγουμε File -> New -> Project. Στα Templates επιλέγουμε Visual C++ και επιλέγουμε Empty Project -> OK.

Στο Solution Explorer, στην αριστερή πλευρά, επιλέγουμε την Τρίτη καρτέλα (Property Manager).



Εικόνα 1.3: Property Manager

Επιλέγουμε τον φάκελο Debug|Win32 (εφόσον έχουμε δημιουργήσει τις αντίστοιχες μεταβλητές που περιγράφονται στο κεφάλαιο 1.5 για Windows 32bit ή τον φάκελο Debug|Win64 εφόσον τα Windows είναι 64bit). Κάνουμε δεξί κλικ και στη συνέχεια Add New Project Property Sheet -> Name: OPENCV_DEBUG -> Add. Εφόσον έχει δημιουργηθεί το αρχείο, κάνουμε δεξί κλικ επάνω του και επιλέγουμε Properties.

Properties -> C/C++ -> Additional Include Directories -> Edit -> Τοποθετούμε το: $(OPENCV_BUILD)\include -> OK

Linker -> General -> Additional Library Directories -> Edit -> Τοποθετούμε το: $(OPENCV_BUILD)\x86\vc10\lib -> OK (για 64bit Windows εισάγουμε την τιμή $(OPENCV_BUILD)\x64\vc10\lib).
Τέλος, Linker -> Input -> Additional Dependencies -> Edit και εισάγουμε τα ακόλουθα:

opencv_calib3d249d.lib;
opencv_contrib249d.lib;
opencv_core249d.lib;
opencv_features2d249d.lib;
opencv_flann249d.lib;
opencv_gpu249d.lib;
opencv_highgui249d.lib;
opencv_imgproc249d.lib;
opencv_legacy249d.lib;
opencv_ml249d.lib;
opencv_nonfree249d.lib;
opencv_objdetect249d.lib;
opencv_ocl249d.lib;
opencv_photo249d.lib;
opencv_stitching249d.lib;

opencv_superres249d.lib;
opencv_ts249d.lib;
opencv_video249d.lib;
opencv_videostab249d.lib;

και OK. Με τις ανωτέρω ενέργειες, ολοκληρώσαμε την δημιουργία παραμετροποίηση του αρχείου OPENCV_DEBUG.props.

Τελευταίο βήμα πριν ξεκινήσουμε την ανάπτυξη των προγραμμάτων, δημιουργούμε το αρχείο OPENCV_RELEASE.props ακολουθώντας την παρακάτω διαδικασία:

Επιλέγουμε τον φάκελο Release|Win32 (εφόσον έχουμε Windows 32bit ή τον φάκελο Release|Win64 εφόσον τα Windows είναι 64bit). Κάνουμε δεξί κλικ και στη συνέχεια Add New Project Property Sheet -> Name: OPENCV_RELEASE -> Add. Εφόσον έχει δημιουργηθεί το αρχείο, κάνουμε δεξί κλικ επάνω του και επιλέγουμε Properties.

Properties -> C/C++ -> Additional Include Directories -> Edit -> Τοποθετούμε το: $(OPENCV_BUILD)\include -> OK

Linker -> General -> Additional Library Directories -> Edit -> Τοποθετούμε το: $(OPENCV_BUILD)\x86\vc10\lib -> OK (για 64bit Windows εισάγουμε την τιμή $(OPENCV_BUILD)\x64\vc10\lib).
Τέλος, Linker -> Input -> Additional Dependencies -> Edit και εισάγουμε τα ακόλουθα:

opencv_calib3d249.lib;
opencv_contrib249.lib;
opencv_core249.lib;
opencv_features2d249.lib;
opencv_flann249.lib;
opencv_gpu249.lib;

opencv_highgui249.lib;
opencv_imgproc249.lib;
opencv_legacy249.lib;
opencv_ml249.lib;
opencv_nonfree249.lib;
opencv_objdetect249.lib;
opencv_ocl249.lib;
opencv_photo249.lib;
opencv_stitching249.lib;
opencv_superres249.lib;
opencv_ts249.lib;
opencv_video249.lib;
opencv_videostab249.lib;

Τα δύο αρχεία που δημιουργήθηκαν βρίσκονται στο ακόλουθο path: Visual Studio 2017\Projects\Project2\Project2. Μπορούμε να τα αντιγράψουμε όπου επιθυμούμε προκειμένου να τα χρησιμοποιήσουμε σε όλα τα προγράμματά μας και να μην χρειαστεί να τα δημιουργήσουμε εκ νέου. Εφόσον επιθυμούμε να τα εισάγουμε σε ένα νέο project που θα αξιοποιεί τις βιβλιοθήκες και συναρτήσεις της OpenCV, ακολουθούμε την παραπάνω διαδικασία δημιουργίας, αλλά επιλέγουμε Add Existing Property Sheet αντί για Add New Project Property Sheet και αναζητούμε τα αρχεία που έχουμε δημιουργήσει.

1.8 Τελευταίες παραμετροποιήσεις

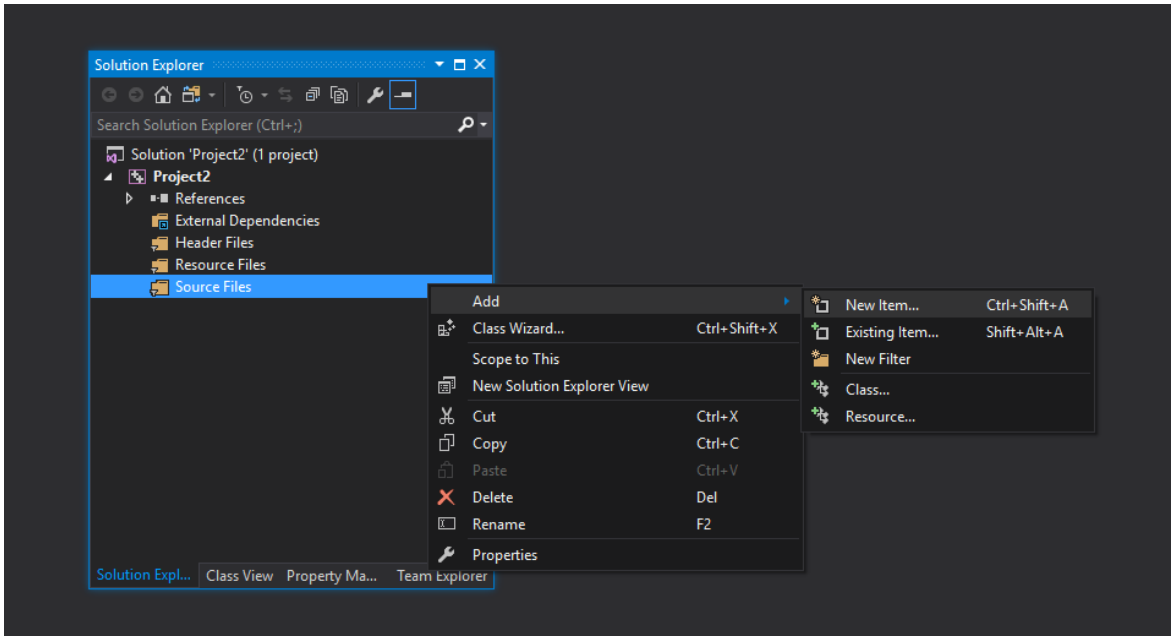
Εφόσον είμαστε σχεδόν έτοιμοι για την ανάπτυξη των προγραμμάτων, θα πρέπει να ολοκληρώσουμε μερικές παραμετροποιήσεις. Προκειμένου να τρέξουν επιτυχώς τα προγράμματά μας, έπρεπε να αντιγράψουμε μερικά lib αρχεία μέσα στον φάκελο System64 των Windows στον υπολογιστή μας. Τα αρχεία, είναι τα ακόλουθα:

msvcr100d.dll
msvcp100d.dll

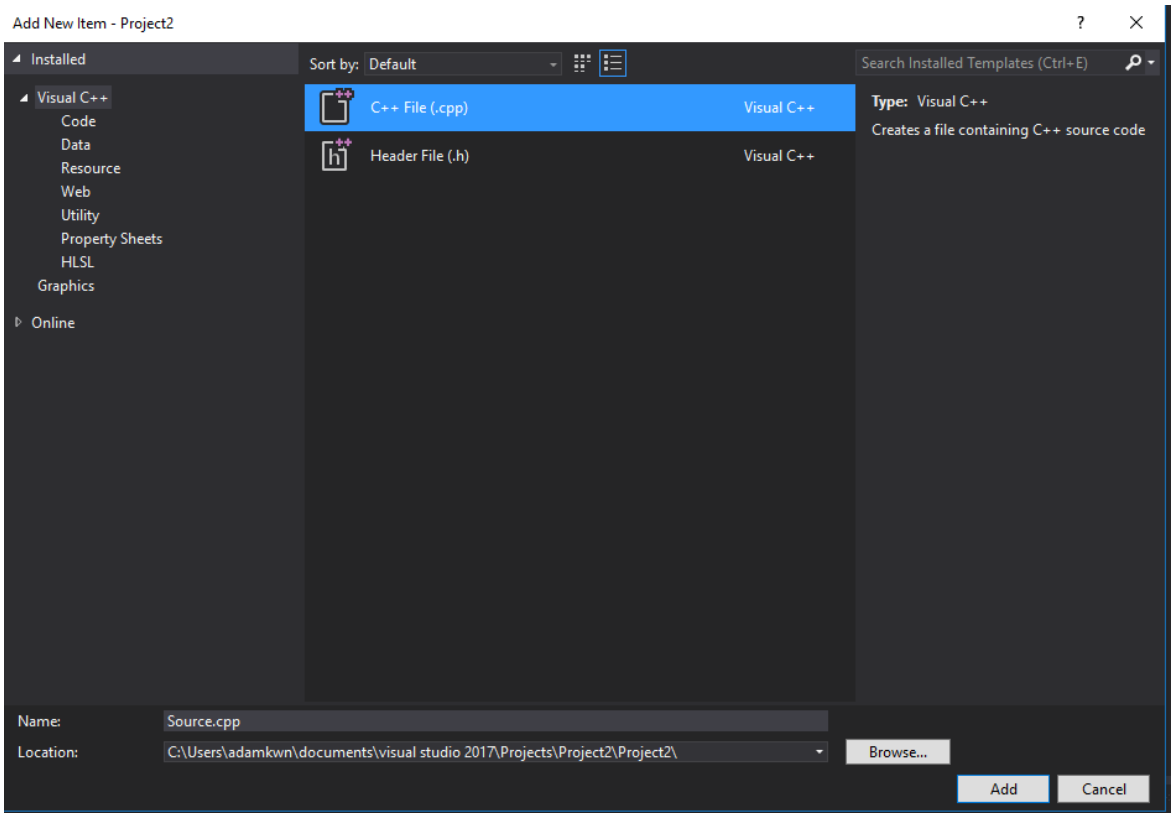
Τα ίδια αρχεία χρειάστηκε να τα αντιγράψουμε και μέσα στον φάκελο του Project που υπάρχουν όλα τα απαραίτητα αρχεία ώστε το Visual Studio να εκτελέσει επιτυχώς προγράμματα που περιλαμβάνουν συναρτήσεις της OpenCV. Επίσης, εφόσον μέσα στον κώδικα μας ορίσουμε το πλήρες path των εικόνων που χρησιμοποιούμε για την επεξεργασία, θα πρέπει να υπάρχουν υποχρεωτικά και μέσα στον φάκελο του project.

Έπειτα από όλα τα ανωτέρω βήματα, επανερχόμαστε στο πρόγραμμα Visual Studio, και δημιουργούμε το αρχείο, στο οποίο θα γράψουμε τον κώδικα. Αυτό το κάνουμε με τα εξής βήματα:

Στον Property Manager, επιλέγουμε την πρώτη καρτέλα (Solution Explorer). Επιλέγουμε τον φάκελο Source Files και κάνουμε δεξί κλικ -> Add -> New Item -> C++ File (.cpp) στο αναδυόμενο παράθυρο και αφού του δώσουμε το όνομα που επιθυμούμε επιλέγουμε OK.



Εικόνα 1.4: Add cpp file



Εικόνα 1.5: Property Manager

Πλέον μπορούμε να ξεκινήσουμε με την συγγραφή των προγραμμάτων μας.

ΠΑΡΑΡΤΗΜΑ Α’

ΚΕΦΑΛΑΙΟ 2

2.1 ΕΙΣΑΓΩΓΗ

Στο παράρτημα αυτό παρατίθεται ο κώδικας ανάπτυξης της κάθε ανεξάρτητης εφαρμογής με μια σύντομη περιγραφή, παρουσίαση του κώδικα και λεπτομερή σχολιασμό του καθώς και τα αποτελέσματα που προκύπτουν κατά την εκτέλεση τους.

2.2 READ AND SHOW AN IMAGE

2.2.1 Λειτουργία προγράμματος

Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει και εμφανίζει μια εικόνα.

2.2.2 Κώδικας προγράμματος

```
1  #include <opencv\cv.h>
2  #include <opencv\highgui.h>
3  #include <iostream>
4  #include <windows.h>
5
6  using namespace cv;
7  using namespace std;
8
9  int main()
10 {
11
12         Mat image = imread("C:\lena.bmp");
13
14         if (image.empty())
15         {
16                 printf("Image is not read! Wrong file or directory!");
17                 Sleep(10000);
18         }
```

```
19        else
20        {
21                imshow("Picture", image);
22        }
23        waitKey(0);
24
25
26        return 0;
27 }
```

2.2.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 4: Εισαγωγή βιβλιοθηκών.

Γραμμές 6 και 7: Χρήση των cv και std, namespaces.

Γραμμή 12: Διάβασμα της εικόνας από τον δίσκο.

Γραμμές 14 έως 22: Έλεγχος για το διάβασμα της εικόνας. Σε περίπτωση που δεν διαβάστηκε ομαλά η εικόνα εμφανίζεται ανάλογο μήνυμα, αλλιώς εμφανίζει την εικόνα (με την χρήση της imshow).

Γραμμή 23: Αναμονή για πληκτρολόγηση οποιουδήποτε πλήκτρου ώστε να τερματιστεί το πρόγραμμα.

2.2.4 Εκτέλεση-Αποτελέσματα προγράμματος

Εικόνα 2.1: Εμφάνιση της εικόνας “lena.bmp”

2.3 IMHIST (ΙΣΤΟΓΡΑΜΜΑ)

2.3.1 Λειτουργία προγράμματος

Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα και στη συνέχεια εμφανίζει το κανονικοποιημένο ιστόγραμμά της. Τέλος, αποθηκεύει το αποτέλεσμα σε καινούρια εικόνα με το όνομα: ‘‘saved_image.png’’.

2.3.2 Κώδικας προγράμματος

```
1  #include "opencv2/objdetect/objdetect.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3  #include "opencv2/imgproc/imgproc.hpp"
4  #include <iostream>
5
6  using namespace std;
7  using namespace cv;
8
9  int main(int, char**)
10 {
11          Mat gray = imread("lena.bmp", 0);
12          namedWindow("Gray", 1);    imshow("Gray", gray);
13
14          // Initialize parameters
15          int histSize = 256;    // bin size
16          float range[] = { 0, 255 };
17          const float *ranges[] = { range };
18
19          // Calculate histogram
20          MatND hist;
21          calcHist(&gray, 1, 0, Mat(), hist, 1, &histSize, ranges, true, false);
22
23          // Plot the histogram
24          int hist_w = 512; int hist_h = 400;
25          int bin_w = cvRound((double)hist_w / histSize);
26
27          Mat histImage(hist_h, hist_w, CV_8UC1, Scalar(0, 0, 0));
28          normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, -1, Mat());
29
30          for (int i = 1; i < histSize; i++)
31          {
32                  line(histImage, Point(bin_w*(i - 1), hist_h - cvRound(hist.at<float>(i - 1))),
33                                  Point(bin_w*(i), hist_h - cvRound(hist.at<float>(i))),
34                                  Scalar(255, 0, 0), 2, 8, 0);
```

```
35          }
36
37          namedWindow("Result", 1);
38          imshow("Result", histImage);
39
40          //convert image to [0-255] in order to save
41          histImage.convertTo(histImage, CV_64F, 255.0 / 1.0);
42          imwrite("saved_image.png", histImage);
43
44          waitKey(0);
45          return 0;
46 }
```

2.3.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 4: Εισαγωγή βιβλιοθηκών

Γραμμές 6 και 7: Χρήση των cv και std namespaces

Γραμμή 11: Διάβασμα της εικόνας lena.bmp

Γραμμή 12: Εμφάνιση εικόνας ώστε να υπολογιστεί το ιστόγραμμα

Γραμμή 15 έως 17: Αρχικοποίηση μεταβλητών

Γραμμή 20: Δημιουργία εικόνας ιστογράμματος

Γραμμή 21: Υπολογισμός ιστογράμματος

Γραμμή 24 και 25: Αρχικοποίηση μεταβλητών για μέγεθος απεικόνισης ιστογράμματος

Γραμμή 28: Κανονικοποίηση ιστογράμματος

Γραμμή 30: Επαναληπτική διαδικασία υπολογισμού κανονικοποιημένου ιστογράμματος

Γραμμές 32 έως 34: Υπολογισμός κανονικοποιημένου ιστογράμματος

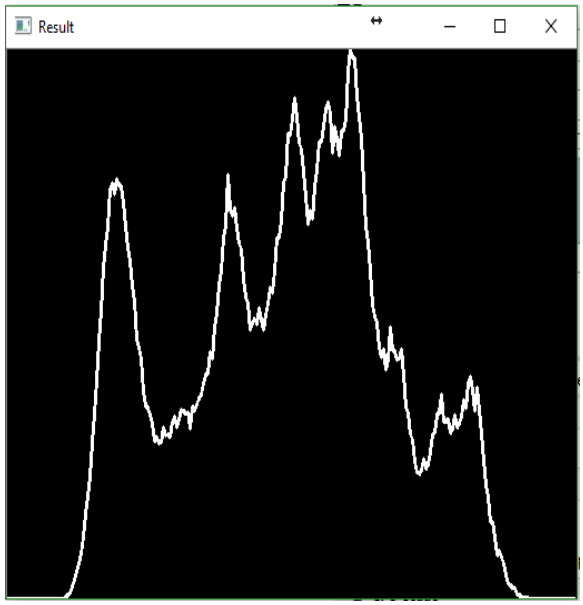
Γραμμή 38: Εμφάνιση ιστογράμματος

Γραμμή 42: Αποθήκευση ιστογράμματος σε μορφή εικόνας

Γραμμή 44: Αναμονή για οποιοδήποτε πλήκτρο ώστε να τερματίσει το πρόγραμμα

2.3.4 Εκτέλεση-Αποτελέσματα προγράμματος

Εικόνα 2.2: Εικόνα εισόδου.



Εικόνα 2.3: Κανονικοποιημένο ιστόγραμμα

2.4 HISTEQUALIZE(ΕΞΙΣΟΡΡΟΠΗΣΗ ΙΣΤΟΓΡΑΜΜΑΤΟΣ)

2.4.1 Λειτουργία προγράμματος

Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα και στη συνέχεια εμφανίζει το κανονικοποιημένο ιστόγραμμά της. Στη συνέχεια, με τη χρήση της συνάρτησης equalizeHist() υπολογίζει και εμφανίζει το εξισορροπημένο ιστόγραμμα καθώς και την εικόνα που προκύπτει από αυτό. Τέλος, αποθηκεύει το κανονικοποιημένο ιστόγραμμα, το εξισορροπημένο ιστόγραμμα και την εικόνα που προκύπτει από αυτό στον δίσκο.

2.4.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include <iostream>
5
6 using namespace std;
7 using namespace cv;
8
9 int main(int, char**)
10{
11      Mat gray = imread("pentagon.png", 0);
12      Mat eqHist;
13
14      equalizeHist(gray, eqHist);
15
16      // Initialize parameters
17      int histSize = 256;    // bin size
18      float range[] = { 0, 255 };
19      const float *ranges[] = { range };
20
21      // Calculate histogram
22      MatND hist, hist2;
23      calcHist(&eqHist, 1, 0, Mat(), hist, 1, &histSize, ranges, true, false);
24      calcHist(&gray, 1, 0, Mat(), hist2, 1, &histSize, ranges, true, false);
25
26      // Plot the histogram
27      int hist_w = 512; int hist_h = 400;
28      int bin_w = cvRound((double)hist_w / histSize);
29
30      Mat histImage(hist_h, hist_w, CV_8UC1, Scalar(0, 0, 0));
31      normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, -1, Mat());
```

```
32
33      Mat histImage2(hist_h, hist_w, CV_8UC1, Scalar(0, 0, 0));
34      normalize(hist2, hist2, 0, histImage2.rows, NORM_MINMAX, -1, Mat());
35
36      for (int i = 1; i < histSize; i++)
37      {
38              line(histImage, Point(bin_w*(i - 1), hist_h - cvRound(hist.at<float>(i - 1))),
39                      Point(bin_w*(i), hist_h - cvRound(hist.at<float>(i))),
40                      Scalar(255, 0, 0), 2, 8, 0);
41
42              line(histImage2, Point(bin_w*(i - 1), hist_h - cvRound(hist2.at<float>(i - 1))),
43                      Point(bin_w*(i), hist_h - cvRound(hist2.at<float>(i))),
44                      Scalar(255, 0, 0), 2, 8, 0);
45      }
46
47
48      namedWindow("Original image", 1);
49      imshow("Original image", gray);
50
51      namedWindow("Equalized Image", 1);
52      imshow("Equalized Image", eqHist);
53
54      namedWindow("Result hist equalize", 1);
55      imshow("Result hist equalize", histImage2);
56
57      namedWindow("Result hist original", 1);
58      imshow("Result hist original", histImage);
59
60      //convert image to [0-255] in order to save
61      eqHist.convertTo(eqHist, CV_64F, 255.0 / 1.0);
62      imwrite("saved_image.png", eqHist);
63
64      histImage.convertTo(histImage, CV_64F, 255.0 / 1.0);
65      imwrite("saved_image2.png", histImage);
66
67      histImage2.convertTo(histImage2, CV_64F, 255.0 / 1.0);
68      imwrite("saved_image3.png", histImage2);
69
70      waitKey(0);
71      return 0;
72}
```

2.4.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 4: Εισαγωγή βιβλιοθηκών

Γραμμές 6 και 7: Χρήση των cv και std namespaces

Γραμμή 11: Διάβασμα της εικόνας (pentagon.png)

Γραμμή 12: Εξισορρόπηση εικόνας με χρήση της συνάρτησης equalizeHist

Γραμμή 17 έως 19: Αρχικοποίηση μεταβλητών

Γραμμή 22: Δήλωση δύο εικόνων, στις οποίες θα αποθηκευτούν τα ιστογράμματα

Γραμμή 23 και 24: Υπολογισμός ιστογραμμάτων αρχικής και βελτιστοποιημένης εικόνας

Γραμμή 27 και 28: Αρχικοποίηση μεταβλητών

Γραμμή 30: Δημιουργίας εικόνας για απεικόνιση ιστογράμματος αρχικής εικόνας

Γραμμή 31: Κανονικοποίηση ιστογράμματος

Γραμμή 33: Δημιουργίας εικόνας για απεικόνιση ιστογράμματος που έχει υποστεί εξισορρόπηση ιστογράμματος

Γραμμή 34: Κανονικοποίηση εξισορροπημένου ιστογράμματος

Γραμμή 36 έως 45: Επαναληπτική διαδικασία για τον υπολογισμό κανονικοποιημένου ιστογράμματος

Γραμμή 49: Εμφάνιση αρχικής εικόνας

Γραμμή 52: Εμφάνιση εικόνας με εξισορρόπηση ιστογράμματος

Γραμμή 55: Εμφάνιση εξισορροπημένου ιστογράμματος

Γραμμή 58: Εμφάνιση αρχικού ιστογράμματος

Γραμμές 61, 64, 67: Μετατροπή εικόνων στο διάστημα [0.255] για αποθήκευση

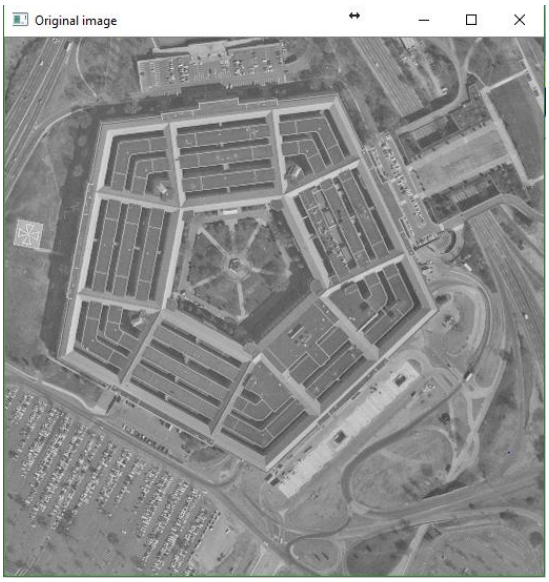
Γραμμή 62: Αποθήκευση εξισορροπημένης εικόνας

Γραμμή 65: Αποθήκευση αρχικού ιστογράμματος

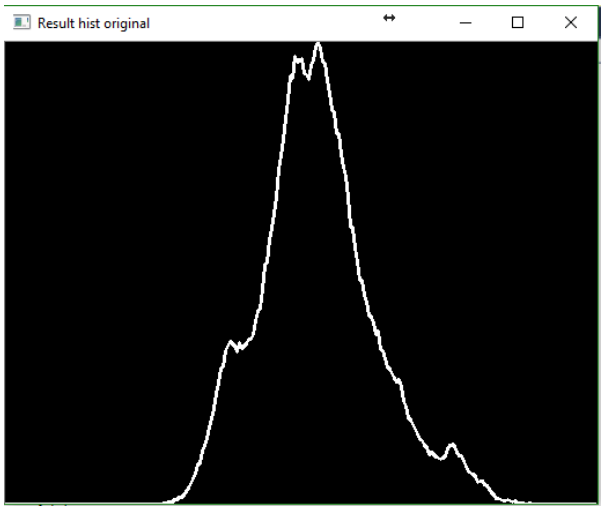
Γραμμή 68: Αποθήκευση εξισορροπημένου ιστογράμματος

Γραμμή 70: Αναμονή για οποιοδήποτε πλήκτρο ώστε να τερματίσει το πρόγραμμα

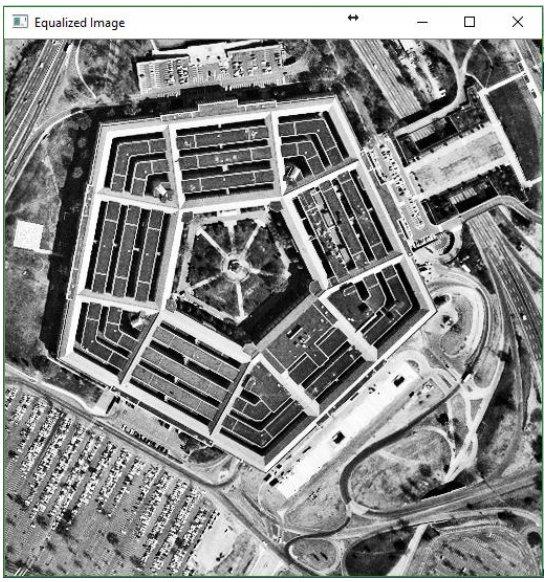
2.4.4 Εκτέλεση-Αποτελέσματα προγράμματος



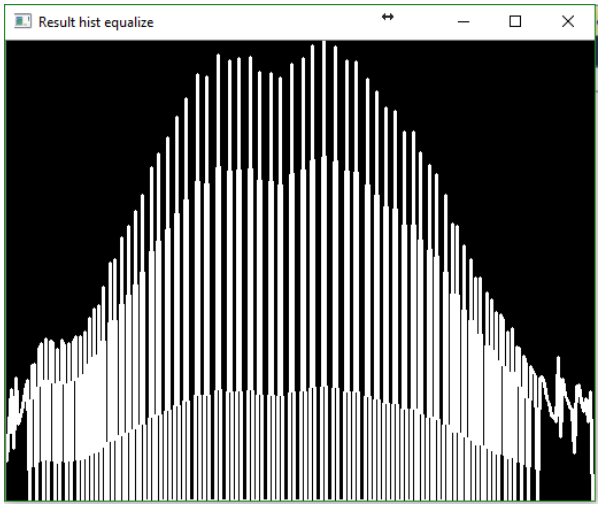
Εικόνα 2.4: Αρχική εικόνα.



Εικόνα 2.5: Κανονικοποιημένο ιστόγραμμα της εικόνας.



Εικόνα 2.6: Εικόνα που προκύπτει από εξισορρόπηση ιστογράμματος.



Εικόνα 2.7: Εξισορροπημένο ιστόγραμμα της εικόνας.

2.5 INVERSE (ΑΝΤΙΣΤΡΟΦΗ ΦΩΤΕΙΝΟΤΗΤΑΣ)

2.5.1 Λειτουργία προγράμματος

Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα. Στη συνέχεια, για να επιτευχθεί η αντιστροφή της φωτεινότητας της εικόνας επεξεργάζεται την τιμή της φωτεινότητας των pixel σύμφωνα με τη σχέση: I΄ = L – I όπου I΄ είναι η νέα τιμή φωτεινότητας των pixel, I είναι η υπάρχουσα τιμή φωτεινότητας των pixel και L=1. Τέλος, αποθηκεύεται το αποτέλεσμα της αντιστροφής φωτεινότητας σε νέα εικόνα στο δίσκο.

2.5.2 Κώδικας προγράμματος

```
1  #include "opencv2/objdetect/objdetect.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3  #include "opencv2/imgproc/imgproc.hpp"
4  #include "opencv2/core/core.hpp"
5  #include <opencv/cv.h>
6  #include <iostream>
7  #include <Windows.h>
8
9  using namespace std;
10 using namespace cv;
11
12 int main(int, char**)
13{
14          Mat image = imread("spine.png", 0);
15          Mat inverse = image.clone();
16
17          if (image.empty())//check if empty
18          {
19                    printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20                    Sleep(10000); //10 second delay for showing message to user
21          }
22          else
23          {
24                    double new_pixel, pixel;
25
26                    image.convertTo(inverse, CV_64F, 1.0 / 255.0);
27
28
29                    for (int i = 0; i < inverse.rows; i++)
30                              for (int j = 0; j < inverse.cols; j++)
31                              {
```

```
32                              pixel = inverse.at<double>(i, j);//pixels value
33
34                              new_pixel = 1 - pixel;
35
36                              inverse.at<double>(i, j) = new_pixel; //antikatastash pixel
37                              }
38
39                  namedWindow("Original Image", 1);
40                  imshow("Original Image", image);
41
42                  namedWindow("Inverse Image", 1);
43                  imshow("Inverse Image", inverse);
44
45                  //convert image to [0-255] in order to save
46                  inverse.convertTo(inverse, CV_64F, 255.0 / 1.0);
47                  imwrite("saved_image.png", inverse);
48
49                  waitKey();
50
51          }
52          return 0;
53 }
```

2.5.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμές 14 και 15: Διάβασμα εικόνας (pentagon.png) και αντιγραφής του περιεχομένου της σε μια εικόνα επεξεργασία.

Γραμμές 17 έως 21: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 26: Μετατροπή της εικόνας inverse σε εικόνα φωτεινότητας στην περιοχή [0,1] (από [0,255]) για περαιτέρω επεξεργασία.

Γραμμές 29 και 30: Επαναληπτική διαδικασία που θα ελέγξει όλα τα pixel της εικόνας

Γραμμή 32: Διάβασμα της τιμής του εκάστοτε pixel

Γραμμή 34: Αλλαγή τιμής σύμφωνα με την σχέση 1-τιμή_φωτεινότητας (αντιστροφή)

Γραμμή 36: Εισαγωγή τροποποιημένης τιμής στην εικόνα, στην θέση που ήταν.

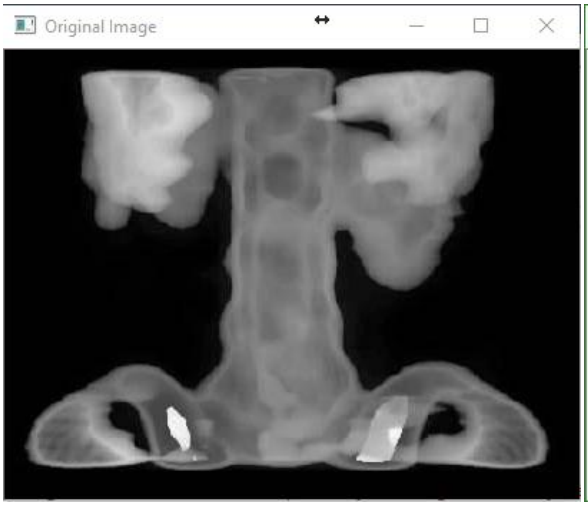
Γραμμή 40: Εμφάνιση αρχικής εικόνας

Γραμμή 42: Εμφάνιση τροποποιημένης εικόνας

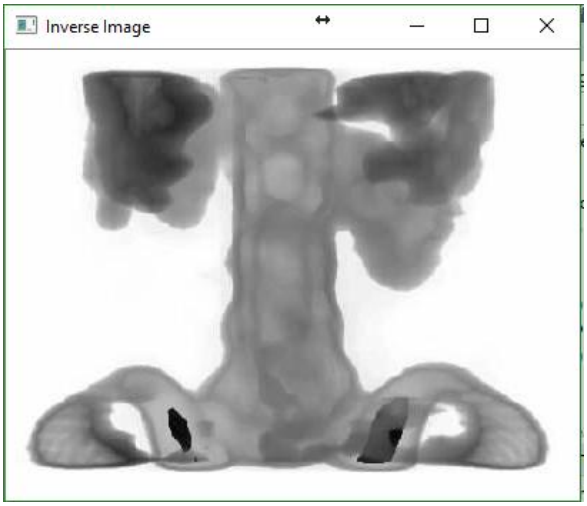
Γραμμή 47: Αποθήκευση τροποποιημένης εικόνας

Γραμμή 49: Αναμονή για οποιοδήποτε πλήκτρο ώστε να τερματιστεί το πρόγραμμα

2.5.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.8: Αρχική εικόνα.



Εικόνα 2.9: Εικόνα με αντιστροφή φωτεινότητας.

2.6 IMADJUST (ΕΠΕΚΤΑΣΗ ΑΝΤΙΘΕΣΗΣ)

2.6.1 Λειτουργία προγράμματος

Επέκταση της αντίθεσης με χρήση της συνάρτησης imadjust. Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα. Στη συνέχεια περιμένει από τον χρήστη έναν αριθμό στο διάστημα [0-1] για το μικρό όριο (low) και έναν για το μεγάλο όριο (high).Η επέκταση της αντίθεσης επιτυγχάνεται με τη μετατροπή των τιμών φωτεινότητας των pixel της εικόνας που είναι μικρότερες από το low σε 0 και μεγαλύτερες ή ίσες με high σε 1.Οι ενδιάμεσες τιμές των pixel διαμορφώνονται από τη σχέση: I΄=(I-low)/(high-low) όπου: I΄ η καινούρια τιμή του pixel και I η υπάρχουσα τιμή. Τέλος, αποθηκεύεται η επεργασμένη εικόνα στο δίσκο.

2.6.2 Κώδικας προγράμματος

```
1  #include "opencv2/objdetect/objdetect.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3  #include "opencv2/imgproc/imgproc.hpp"
4  #include "opencv2/core/core.hpp"
5  #include <opencv/cv.h>
6  #include <iostream>
7  #include <Windows.h>
8
9  using namespace std;
10 using namespace cv;
11
12 int main(int, char**)
13 {
14          Mat image = imread("C:\lena.bmp", 0);
15          Mat slice = image.clone();
16          Mat temp = slice.clone();
17          if (image.empty())//check if empty
18          {
19                    printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20                    waitKey(); //10 second delay for showing message to user
21          }
22          else
23          {
24                    double new_pixel, pixel, a, b;
25
26                    image.convertTo(slice, CV_64F, 1.0 / 255.0);
27                    image.convertTo(temp, CV_64F, 1.0 / 255.0);
28
29                    printf("Give low in double [0-1]:\n");
```

```
30		cin >> a;
31
32
33		if (a >= 0 && a <= 1)
34		{
35			printf("Give high in double [0-1]:\n");
36			cin >> b;
37			if (b >= 0 && b <= 1)
38			{
39				//peirazei mono apo low kai katw kai apo high kai panw
40				for (int i = 0; i < slice.rows; i++)
41					for (int j = 0; j < slice.cols; j++)
42					{
43
44					pixel = slice.at<double>(i, j);//pixels value
45
46					if (pixel <= a)
47					{
48						new_pixel = 0;
49					}
50					else if (pixel >= b)
51					{
52						new_pixel = 1;
53					}
54					else
55					{
56						new_pixel = (pixel - a) / (b - a);
57					}
58
59					slice.at<double>(i, j) = new_pixel;
60					}
61				namedWindow("Original", 1);
62				imshow("Original", image);
63
64				namedWindow("adjusted Image", 1);
65				imshow("adjusted Image", slice);
66
67				//convert image to [0-255] in order to save
68				slice.convertTo(slice, CV_64F, 255.0 / 1.0);
69				imwrite("saved_image.png", slice);
70
71				waitKey();
72			}
73			else
74			{
```

```
75                                  printf("b was off limits! Try again.");
76                          }
77                  }
78                  else
79                  {
80                          printf("a was off limits! Try again.");
81                  }
82          }
83          return 0;
84 }
```

2.6.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμές 14 έως 16: Διάβασμα εικόνας (lena.bmp) και αντιγραφής του περιεχομένου της σε δύο εικόνες για επεξεργασία.

Γραμμές 17 έως 21: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 26 και 27: Μετατροπή των εικόνων slice και temp σε εικόνες φωτεινότητας στην περιοχή [0,1] (από [0,255]) για περαιτέρω επεξεργασία.

Γραμμές 29 και 30: Διάβασμα από τον χρήστη (πληκτρολόγιο) το επιθυμητό low

Γραμμή 33: Έλεγχο για σωστή είσοδο από χρήστη

Γραμμές 35 και 36: Διάβασμα από τον χρήστη, το επιθυμητό high

Γραμμή 37: Έλεγχο για σωστή είσοδο του χρήστη

Γραμμές 40 και 41: Επαναληπτική διαδικασία που θα ελέγξει όλα τα pixel της εικόνας

Γραμμή 44: Διάβασμα της τιμής του εκάστοτε pixel

Γραμμή 46: Έλεγχος εάν η τιμή του pixel είναι μικρότερη από το low που έχει δώσει ο χρήστης.

Γραμμή 50: Έλεγχος εάν η τιμή του pixel είναι μεγαλύτερη από το high που έχει δώσει ο χρήστης.

Γραμμή 54: Εφόσον η τιμή του pixel είναι εντός των ορίων του χρήστη, θα τροποποιηθεί σύμφωνα με την σχέση νέα τιμή = (παλιά τιμή – low) / (high – low)

Γραμμή 59: Εισαγωγή τροποποιημένης τιμής (σύμφωνα με τις ανωτέρω Γραμμές) στην εικόνα, στην θέση που ήταν.

Γραμμή 62: Εμφάνιση αρχικής εικόνας

Γραμμή 65: Εμφάνιση τροποποιημένης εικόνας

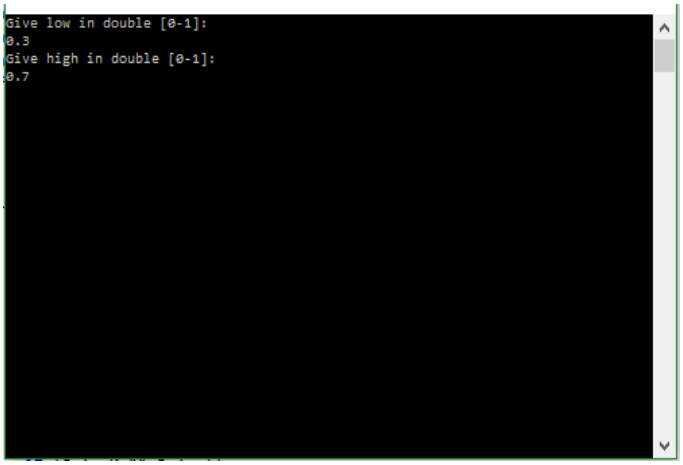
Γραμμή 69: Αποθήκευση τροποποιημένης εικόνας

Γραμμή 71: Αναμονή για οποιοδήποτε πλήκτρο ώστε να τερματιστεί το πρόγραμμα

Γραμμή 75: Εμφάνιση κατάλληλου μηνύματος στον χρήστη, εάν το high που έχει εισάγει είναι εκτός των ορίων, δηλαδή εκτός του [0,1].

Γραμμή 80: Εμφάνιση κατάλληλου μηνύματος στον χρήστη, εάν το low που έχει εισάγει είναι εκτός των ορίων, δηλαδή εκτός του [0,1].

2.6.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.10: Κατά την εκτέλεση δίνουμε τιμές 0.3 για low και 0.7 για high.

Εικόνα 2.11: Αρχική εικόνα.

Εικόνα 2.12: Εικόνα με επέκταση αντίθεσης.

2.7 RESCALE (ΚΛΙΜΑΚΩΣΗ ΦΩΤΕΙΝΟΤΗΤΑΣ)

2.7.1 Λειτουργία προγράμματος

Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα. Στη συνέχεια, βρίσκει την ελάχιστη (min) και μέγιστη (max) τιμή φωτεινότητας της αρχικής εικόνας . Η κλιμάκωση της φωτεινότητας επιτυγχάνεται θέτοντας ως κατώτατο όριο των τιμών φωτεινότητας της νέας εικόνας το ελάχιστο (min) της αρχικής και ως ανώτατο όριο το μέγιστο (max).Οι ενδιάμεσες τιμές φωτεινότητας των pixel διαμορφώνονται από τη σχέση: I΄=(I-min)/(max-min) όπου: I΄ η καινούρια τιμή του pixel και I η υπάρχουσα τιμή. Τέλος, αποθηκεύεται η επεργασμένη εικόνα στο δίσκο.

2.7.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <iostream>
6 #include <Windows.h>
7
8 using namespace std;
9 using namespace cv;
10
11int main(int, char**)
12{
13          Mat image = imread("C:\lena.bmp", 0);
14          Mat resc = image.clone();
15
16          if (image.empty())//check if empty
17          {
18                    printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
19                    waitKey(); //10 second delay for showing message to user
20          }
21          else
22          {
23                    double xmin, xmax;
24
25                    image.convertTo(resc, CV_64F, 1.0 / 255.0);
26
27                    minMaxLoc(resc, &xmin, &xmax);
28
29                    namedWindow("Image", 1);
```

```
30                  imshow("Image", image);
31
32
33
34
35                  double pixel, new_pixel;
36                  Mat rescale = Mat::ones(resc.size(), resc.type());
37
38                  for (int i = 0; i < resc.rows; i++)
39                              for (int j = 0; j < resc.cols; j++)
40                              {
41
42                              pixel = resc.at<double>(i, j);//pixels value
43
44                              new_pixel = (pixel - xmin) / (xmax - xmin);
45
46                              rescale.at<double>(i, j) = new_pixel;
47                              }
48
49
50                  namedWindow("Rescale", 1);
51                  imshow("Rescale", rescale);
52
53                  //convert image to [0-255] in order to save
54                  rescale.convertTo(rescale, CV_64F, 255.0 / 1.0);
55                  imwrite("saved_image.png", rescale);
56
57                  waitKey();
58          }
59          return 0;
60 }
```

2.7.3 Ανάλυση κώδικα προγράμματος

Γραμμή 1 έως 6: Εισαγωγή βιβλιοθηκών

Γραμμή 8 και 9: Χρήση cv και std namespaces

Γραμμή 13: Διάβασμα εικόνας lena.bmp από τον δίσκο

Γραμμή 14: Κλωνοποίηση εικόνας στην resc (ίδιο μέγεθος, ίδιες ιδιότητες)

Γραμμή 16: Έλεγχος ορθού διαβάσματος εικόνας

Γραμμή 18: Εμφάνιση μηνύματος στον χρήστη για λανθασμένη εικόνα

Γραμμή 19: Αναμονή πατήματος πλήκτρου για τερματισμό προγράμματος

Γραμμή 23: Δήλωση μεταβλητών μέγιστης και ελάχιστης φωτεινότητας

Γραμμή 25: Μετατροπή εικόνας στο διάστημα [0,1]

Γραμμή 27: Εύρεση μέγιστης και ελάχιστης φωτεινότητας της εικόνας και αποθήκευση τιμών στα xmin και xmax

Γραμμή 30: Εμφάνιση αρχικής εικόνας

Γραμμή 35: Δήλωση δύο μεταβλητών για διάβασμα του εκάστοτε pixel και επεξεργασία φωτεινότητας

Γραμμή 36: Δήλωση και δημιουργία εικόνας που όλες οι φωτεινότητες έχουν την τιμή 1

Γραμμή 38 και 39: Επαναληπτικές διαδικασίες για διάβασμα και παραμετροποίηση pixel εικόνας

Γραμμή 42: Διάβασμα τιμής φωτεινότητας εκάστοτε pixel της εικόνας

Γραμμή 44: Τροποποίηση φωτεινότητας pixel σύμφωνα με την σχέση, νέα τιμή = (παλιά τιμή – min) / (max – min)

Γραμμή 46: Εισαγωγή νέας τιμής φωτεινότητας στην ίδια θέση της εικόνας

Γραμμή 51: Εμφάνιση τελικής εικόνας με αντεστραμμένη φωτεινότητα

Γραμμή 54: Μετατροπή εικόνας στο διάστημα [0,255]

Γραμμή 55: Αποθήκευση αντεστραμμένης εικόνας

Γραμμή 57: Αναμονή από τον χρήστη για πάτημα οποιουδήποτε πλήκτρου ώστε να τερματιστεί το πρόγραμμα

2.7.4 Εκτέλεση-Αποτελέσματα προγράμματος

Εικόνα 2.13: Αρχική εικόνα.

Εικόνα 2.14: Εικόνα με κλιμάκωση φωτεινότητας.

2.8 SLICEIM (ΤΕΜΑΧΙΣΜΟΣ ΦΩΤΕΙΝΟΤΗΤΑΣ)

2.8.1 Λειτουργία προγράμματος

Ο τεμαχισμός φωτεινότητας επιτυγχάνεται με δύο τεχνικές. Η μία περίπτωση οδηγεί μόνο μια περιοχή [A,B] στη μέγιστη τιμή φωτεινότητας χωρίς να επηρεάσει την υπόλοιπη περιοχή ενώ η δεύτερη περίπτωση οδηγεί την περιοχή [A,B] στη μέγιστη τιμή φωτεινότητας υποβαθμίζοντας την υπόλοιπη περιοχή στο μηδέν. Το πρόγραμμα αυτό κατά την εκτέλεσή του διαβάζει μια εικόνα. Στη συνέχεια, ζητάει από τον χρήστη να πληκτρολογήσει έναν ακέραιο αριθμό στο διάστημα [0-255] για Α και έναν ακέραιο στο ίδιο διάστημα για Β. Επιπλέον, ζητάει από τον χρήστη να πληκτρολογήσει τον αριθμό ‘1’ που αντιστοιχεί στην πρώτη περίπτωση και ‘2’ που αντιστοιχεί στην δεύτερη. Τέλος, εμφανίζονται τα αποτελέσματα του τεμαχισμού της εικόνας ανάλογα με την επιλογή του χρήστη και αποθηκεύεται η επεργασμένη εικόνα στο δίσκο.

2.8.2 Κώδικας προγράμματος

```
1  #include "opencv2/objdetect/objdetect.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3  #include "opencv2/imgproc/imgproc.hpp"
4  #include "opencv2/core/core.hpp"
5  #include <opencv/cv.h>
6  #include <iostream>
7  #include <Windows.h>
8
9  using namespace std;
10 using namespace cv;
11
12 int main(int, char**)
13 {
14          Mat image = imread("C:\lena.bmp", 0);
15          Mat slice = image.clone();
16          Mat temp = slice.clone();
17          if (image.empty())//check if empty
18          {
19                    printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20                    waitKey(); //10 second delay for showing message to user
21          }
22          else
23          {
24                    double xmin, xmax, new_pixel, pixel;
25
26                    image.convertTo(slice, CV_64F, 1.0 / 255.0);
```

```
27          image.convertTo(temp, CV_64F, 1.0 / 255.0);
28
29          minMaxLoc(slice, &xmin, &xmax);
30
31          double a, b;
32          int choice, A, B;
33
34          printf("Give integer [0-255] for A:\n");
35          cin >> A;
36
37
38          if (A >= 0 && A <= 255)
39          {
40                  printf("Give integer [0-255] for B:\n");
41                  cin >> B;
42                  if (B >= 0 && B <= 255)
43                  {
44                          a = (double)A / 255; // convert integers to double
45                          b = (double)B / 255;
46
47                          printf("Select case: 1 or 2\n");
48                          cin >> choice;
49                  }
50                  else
51                  {
52                          exit;
53                  }
54          }
55          else
56          {
57                  exit;
58          }
59
60          switch (choice)
61          {
62          case 1: //peirazei mono metaxu A kai B. Ola ta alla paramenoyn opws einai
63                  for (int i = 0; i < slice.rows; i++)
64                          for (int j = 0; j < slice.cols; j++)
65                          {
66
67                          pixel = slice.at<double>(i, j);//pixels value
68
69                          if (pixel >= a && pixel <= b)
70                          {
71                                  new_pixel = 1;
```

```
72                                      }
73                                      else
74                                      {
75                                              new_pixel = pixel;
76                                      }
77
78
79                                      slice.at<double>(i, j) = new_pixel;
80                                      }
81                              namedWindow("Original", 1);
82                              imshow("Original", image);
83
84                              namedWindow("Slice Image", 1);
85                              imshow("Slice Image", slice);
86
87                              waitKey();
88
89                      case 2: //kanei sto 1 metaxu A kai B. Ta upoloipa ta kanei 0.
90
91                              for (int i = 0; i < slice.rows; i++)
92                                      for (int j = 0; j < slice.cols; j++)
93                                      {
94
95                                      pixel = slice.at<double>(i, j);//pixels value
96
97                                      if (pixel >= a && pixel <= b)
98                                      {
99                                              new_pixel = 1;
100                                     }
101                                     else
102                                     {
103                                             new_pixel = 0;
104                                     }
105
106
107                                     slice.at<double>(i, j) = new_pixel;
108                                     }
109                             namedWindow("Original", 1);
110                             imshow("Original", image);
111
112                             namedWindow("Slice Image", 1);
113                             imshow("Slice Image", slice);
114
115                             //convert image to [0-255] in order to save
116                             slice.convertTo(slice, CV_64F, 255.0 / 1.0);
```

```
117                             imwrite("saved_image.png", slice);
118
119                             waitKey();
120
121
122
123                     default:
124                             printf("Wrong case given! Try again!");
125                     }
126
127                     Sleep(5000);
128             }
129             return 0;
130 }
```

2.8.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμές 14 έως 16: Διάβασμα εικόνας (lena.bmp) και αντιγραφής του περιεχομένου της σε δύο εικόνες για επεξεργασία.

Γραμμές 17 έως 21: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 24: Δήλωση των μεταβλητών μέγιστης και ελάχιστης φωτεινότητας της εικόνας, της τρέχουσας και νέας φωτεινότητας

Γραμμή 26 και 27: Μετατροπή των εικόνων στο διάστημα [0,1]

Γραμμή 29: Εύρεση ελάχιστης και μέγιστης τιμής φωτεινότητας της εικόνας

Γραμμή 31: Δήλωση των μεταβλητών a και b

Γραμμή 32: Δήλωση των μεταβλητών A και B (για εισαγωγή φωτεινότητας)

Γραμμή 34: Εμφάνιση μηνύματος στην κονσόλα

Γραμμή 35: Αναμονή για διάβασμα από το πληκτρολόγιο

Γραμμή 38: Έλεγχος εάν ότι έδωσε ο χρήστης είναι μεταβλητή στο διάστημα [0,255]

Γραμμή 40: Εμφάνιση μηνύματος στην κονσόλα

Γραμμή 41: Αναμονή για διάβασμα από το πληκτρολόγιο

Γραμμή 42: Έλεγχος εάν ότι έδωσε ο χρήστης είναι μεταβλητή στο διάστημα [0,255]

Γραμμή 44 και 45: Μετατροπή αριθμών εισόδου (που έδωσε ο χρήστης) με το 255, προκειμένου να βρίσκεται στο διάστημα [0,1]

Γραμμή 47: Εμφάνιση μηνύματος στην κονσόλα

Γραμμή 48: Διάβασμα από το πληκτρολόγιο

Γραμμή 50 και 52: Έξοδος προγράμματος εάν η επιλογή Β είναι εκτός ορίων

Γραμμή 55 και 57: Έξοδος προγράμματος εάν η επιλογή Α είναι εκτός ορίων

Γραμμή 60: Εκκίνηση switch επιλογής

Γραμμή 62: Στην περίπτωση που ο χρήστης έδωσε 1 στην τελευταία επιλογή

Γραμμή 63 και 64: Επαναληπτικές διαδικασίες προκειμένου να επεξεργαστούμε όλα τα pixel της εικόνας

Γραμμή 67: Διάβασμα του εκάστοτε pixel

Γραμμή 69: Έλεγχος εάν το εκάστοτε pixel βρίσκεται μέσα στα όρια του χρήστη (οι δύο πρώτες επιλογές του)

Γραμμή 70 και 71: Εφόσον είναι μέσα στα όρια, τότε το νέο pixel θα γίνει 1

Γραμμή 73: Εφόσον είναι εκτός των ορίων του χρήστη, το νέο pixel θα παραμείνει στην ίδια φωτεινότητα

Γραμμή 75: Αποθήκευση ίδιας φωτεινότητας στην new_pixel ώστε να εισαχθεί στην εικόνα

Γραμμή 79: Εισαγωγή νέας τιμής φωτεινότητας στην εικόνα

Γραμμή 82: Εμφάνιση αρχικής εικόνας

Γραμμή 85: Εμφάνιση επεξεργασμένης εικόνας

Γραμμή 87: Αναμονή για πληκτρολόγηση οποιουδήποτε πλήκτρου για έξοδο του προγράμματος

Γραμμή 89: Δεύτερη επιλογή του χρήστη (δηλαδή δεύτερη λειτουργεία του προγράμματος)

Γραμμή 91 και 92: Επαναληπτικές διαδικασίες προκειμένου να επεξεργαστούμε όλα τα pixel της εικόνας

Γραμμή 95: Διάβασμα του εκάστοτε pixel

Γραμμή 97: Έλεγχος εάν το εκάστοτε pixel βρίσκεται μέσα στα όρια του χρήστη (οι δύο πρώτες επιλογές του)

Γραμμή 99: Εφόσον είναι μέσα στα όρια, τότε το νέο pixel θα γίνει 1

Γραμμή 101 και 103: Εφόσον είναι εκτός των ορίων του χρήστη, το νέο pixel θα πάρει την τιμή 0

Γραμμή 107: Εισαγωγή νέας τιμής φωτεινότητας στην εικόνα

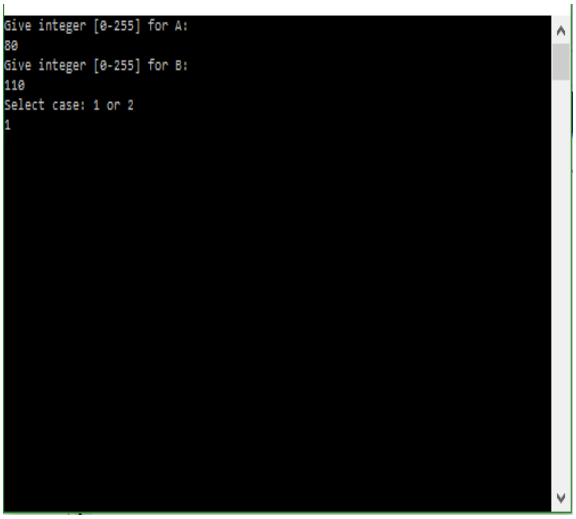
Γραμμή 110: Εμφάνιση αρχικής εικόνας

Γραμμή 113: Εμφάνιση επεξεργασμένης εικόνας

Γραμμή 116: Μετατροπή επεξεργασμένης εικόνας στο διάστημα [0,255] για αποθήκευση
Γραμμή 117: Αποθήκευση επεξεργασμένης εικόνας
Γραμμή 119: Αναμονή για οποιαδήποτε πλήκτρο από τον χρήστη για τερματισμό του προγράμματος
Γραμμή 123: Εφόσον η επιλογή του χρήστη δεν ήταν το 1 ή το 2, εμφανίζεται μήνυμα στην κονσόλα
Γραμμή 127: Αναμονή του προγράμματος για 5 δευτερόλεπτα και μετά τερματισμός

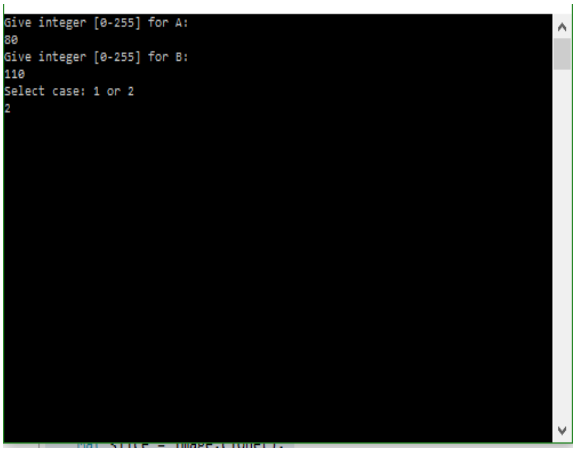
2.8.4 Εκτέλεση-Αποτελέσματα προγράμματος

Εικόνα 2.15: Αρχική εικόνα.



Εικόνα 2.16: Κατά την εκτέλεση δίνουμε τις τιμές 80 και 110 για A και B και ‘1’ για την πρώτη περίπτωση.

Εικόνα 2.17: Τεμαχισμός εικόνας στο διάστημα [80,110] με την πρώτη τεχνική.



Εικόνα 2.18: Κατά την εκτέλεση δίνουμε τις τιμές 80 και 110 για A και B και ‘2’ για την δεύτερη περίπτωση.

Εικόνα 2.19: Τεμαχισμός εικόνας στο διάστημα [80,110] με την δεύτερη τεχνική.

2.9 BITPLANE (ΑΝΑΛΥΣΗ ΣΕ ΔΥΑΔΙΚΕΣ ΕΙΚΟΝΕΣ)

2.9.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα και αναλύεται σε δυαδικές. Η ανάλυση της εικόνας σε δυαδικές επιτυγχάνεται σύμφωνα με τη σχέση: $I(x,y) = a_7 2^7 + a_6 2^6 + a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_0$, όπου a_0, a_1,…, a_7 είναι τα δυαδικά βάρη, τα οποία παίρνουν τιμές 0 ή 1.Συνεπώς, η εικόνα αναλύεται σε οκτώ δυαδικές εικόνες, οι οποίες καλούνται bit planes και περιέχουν τα αντίστοιχα βάρη της προηγούμενης σχέσης. Τέλος, αποθηκεύονται οι δυαδικές εικόνες που προκύπτουν από την αρχική στο δίσκο.

2.9.2 Κώδικας προγράμματος

```
1  #include "opencv2/objdetect/objdetect.hpp"
2  #include "opencv2/highgui/highgui.hpp"
3  #include "opencv2/imgproc/imgproc.hpp"
4  #include "opencv2/core/core.hpp"
5  #include <opencv/cv.h>
6  #include <iostream>
7  #include <Windows.h>
8
9  using namespace std;
10 using namespace cv;
11
12 int main(int, char**)
13 {
14          Mat image = imread("C:\lena.bmp", 0);
15
16          if (image.empty())//check if empty
17          {
18                    printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
19                    waitKey(); //10 second delay for showing message to user
20          }
21          else
22          {
23
24                    Mat bin1(image.rows, image.cols, CV_8UC1, Scalar(0)); //MSB
25                    bin1 = (image / 128);
26                    for (int i = 0; i < image.rows; i++)
27                              for (int j = 0; j < image.cols; j++)
28                              {
29                              bin1.at<uchar>(i, j) = (bin1.at<uchar>(i, j) % 2);
30                              }
```

```
31		bin1 = bin1 * 255;
32
33		Mat bin2(image.rows, image.cols, CV_8UC1, Scalar(0));
34		bin2 = (image / 64);
35		for (int i = 0; i < image.rows; i++)
36			for (int j = 0; j < image.cols; j++)
37			{
38			bin2.at<uchar>(i, j) = (bin2.at<uchar>(i, j) % 2);
39
40			}
41		bin2 = bin2 * 255;
42
43		Mat bin3(image.rows, image.cols, CV_8UC1, Scalar(0));
44		bin3 = (image / 32);
45		for (int i = 0; i < image.rows; i++)
46			for (int j = 0; j < image.cols; j++)
47			{
48			bin3.at<uchar>(i, j) = (bin3.at<uchar>(i, j) % 2);
49
50			}
51		bin3 = bin3 * 255;
52
53		Mat bin4(image.rows, image.cols, CV_8UC1, Scalar(0));
54		bin4 = (image / 16);
55		for (int i = 0; i < image.rows; i++)
56			for (int j = 0; j < image.cols; j++)
57			{
58			bin4.at<uchar>(i, j) = (bin4.at<uchar>(i, j) % 2);
59
60			}
61		bin4 = bin4 * 255;
62
63		Mat bin5(image.rows, image.cols, CV_8UC1, Scalar(0));
64		bin5 = (image / 8);
65		for (int i = 0; i < image.rows; i++)
66			for (int j = 0; j < image.cols; j++)
67			{
68			bin5.at<uchar>(i, j) = (bin3.at<uchar>(i, j) % 2);
69
70			}
71		bin5 = bin5 * 255;
72
73		Mat bin6(image.rows, image.cols, CV_8UC1, Scalar(0));
74		bin6 = (image / 4);
75		for (int i = 0; i < image.rows; i++)
```

```
76                      for (int j = 0; j < image.cols; j++)
77                      {
78                      bin6.at<uchar>(i, j) = (bin6.at<uchar>(i, j) % 2);
79
80                      }
81              bin6 = bin6 * 255;
82
83              Mat bin7(image.rows, image.cols, CV_8UC1, Scalar(0));
84              bin7 = (image / 2);
85              for (int i = 0; i < image.rows; i++)
86                      for (int j = 0; j < image.cols; j++)
87                      {
88                      bin7.at<uchar>(i, j) = (bin7.at<uchar>(i, j) % 2);
89
90                      }
91              bin7 = bin7 * 255;
92
93              Mat bin8(image.rows, image.cols, CV_8UC1, Scalar(0));
94              bin8 = (image / 1);
95              for (int i = 0; i < image.rows; i++)
96                      for (int j = 0; j < image.cols; j++)
97                      {
98                      bin8.at<uchar>(i, j) = (bin8.at<uchar>(i, j) % 2);
99
100                     }
101             bin8 = bin8 * 255;
102
103             namedWindow("Original", 1);
104             imshow("Original", image);
105
106             namedWindow("Binary 1", 1);
107             imshow("Binary 1", bin1);
108
109             namedWindow("Binary 2", 1);
110             imshow("Binary 2", bin2);
111
112             namedWindow("Binary 3", 1);
113             imshow("Binary 3", bin3);
114
115             namedWindow("Binary 4", 1);
116             imshow("Binary 4", bin4);
117
118             namedWindow("Binary 5", 1);
119             imshow("Binary 5", bin5);
120
```

```
121                 namedWindow("Binary 6", 1);
122                 imshow("Binary 6", bin6);
123
124                 namedWindow("Binary 7", 1);
125                 imshow("Binary 7", bin7);
126
127                 namedWindow("Binary 8", 1);
128                 imshow("Binary 8", bin8);
129
130                 //convert image to [0-255] in order to save
131                 bin1.convertTo(bin1, CV_64F, 255.0 / 1.0);
132                 imwrite("saved_image1.png", bin1);
133
134                 bin2.convertTo(bin2, CV_64F, 255.0 / 1.0);
135                 imwrite("saved_image2.png", bin2);
136
137                 bin3.convertTo(bin3, CV_64F, 255.0 / 1.0);
138                 imwrite("saved_image3.png", bin3);
139
140                 bin4.convertTo(bin4, CV_64F, 255.0 / 1.0);
141                 imwrite("saved_image4.png", bin4);
142
143                 bin5.convertTo(bin5, CV_64F, 255.0 / 1.0);
144                 imwrite("saved_image5.png", bin5);
145
146                 bin6.convertTo(bin6, CV_64F, 255.0 / 1.0);
147                 imwrite("saved_image6.png", bin6);
148
149                 bin7.convertTo(bin7, CV_64F, 255.0 / 1.0);
150                 imwrite("saved_image7.png", bin7);
151
152                 bin8.convertTo(bin8, CV_64F, 255.0 / 1.0);
153                 imwrite("saved_image8.png", bin8);
154
155
156                 waitKey();
157         }
158         return 0;
159 }
```

2.9.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμές 14: Διάβασμα εικόνας (lena.bmp).

Γραμμές 16 έως 20: Έλεγχος για ορθό διάβασμα της εικόνας και εμφάνιση μηνύματος εάν δεν διαβαστεί

Γραμμή 21: Εφόσον έχει διαβαστεί επιτυχώς, ξεκινάει η επεξεργασία

Γραμμή 24: Δημιουργία της εικόνας με τα πιο σημαντικά bit της αρχικής

Γραμμή 25: Διαιρώντας με το 128 παίρνουμε το πιο σημαντικό bit από την αρχική

Γραμμή 26 και 27: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 29: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 31: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 33: Δημιουργία της εικόνας με το δεύτερο πιο σημαντικό bit της αρχικής

Γραμμή 34: Διαιρώντας με το 64 παίρνουμε το δεύτερο πιο σημαντικό bit από την αρχική

Γραμμή 35 και 36: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 38: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 41: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 43: Δημιουργία της εικόνας με το τρίτο πιο σημαντικό bit της αρχικής

Γραμμή 44: Διαιρώντας με το 32 παίρνουμε το τρίτο πιο σημαντικό bit από την αρχική

Γραμμή 45 και 46: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 48: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 51: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 53: Δημιουργία της εικόνας με το τέταρτο πιο σημαντικό bit της αρχικής

Γραμμή 54: Διαιρώντας με το 16 παίρνουμε το τέταρτο πιο σημαντικό bit από την αρχική

Γραμμή 55 και 56: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 58: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 61: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 63: Δημιουργία της εικόνας με το πέμπτο πιο σημαντικό bit της αρχικής

Γραμμή 64: Διαιρώντας με το 8 παίρνουμε το πέμπτο πιο σημαντικό bit από την αρχική

Γραμμή 65 και 66: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 68: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 71: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 73: Δημιουργία της εικόνας με το έκτο πιο σημαντικό bit της αρχικής

Γραμμή 74: Διαιρώντας με το 4 παίρνουμε το έκτο πιο σημαντικό bit από την αρχική

Γραμμή 75 και 76: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 78: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 81: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 83: Δημιουργία της εικόνας με το έβδομο πιο σημαντικό bit της αρχικής

Γραμμή 84: Διαιρώντας με το 2 παίρνουμε το έβδομο πιο σημαντικό bit από την αρχική

Γραμμή 85 και 86: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 88: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2

Γραμμή 91: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα

Γραμμή 93: Δημιουργία της εικόνας με το λιγότερο σημαντικό bit της αρχικής

Γραμμή 94: Διαιρώντας με το 1 παίρνουμε το λιγότερο σημαντικό bit από την αρχική

Γραμμή 95 και 96: Επαναληπτική διαδικασία για τροποποίηση της τιμής των pixel της εικόνας

Γραμμή 98: Εισάγουμε στο εκάστοτε pixel την τιμή από το υπόλοιπο της διαίρεσης με το 2
Γραμμή 101: Πολλαπλασιάζουμε με το 255 όλα τα pixel του πίνακα
Γραμμή 104: Εμφάνιση αρχικής εικόνας
Γραμμή 107: Εμφάνιση εικόνας με το πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 110: Εμφάνιση εικόνας με το δεύτερο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 113: Εμφάνιση εικόνας με το τρίτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 116: Εμφάνιση εικόνας με το τέταρτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 119: Εμφάνιση εικόνας με το πέμπτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 122: Εμφάνιση εικόνας με το έκτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 125: Εμφάνιση εικόνας με το έβδομο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 128: Εμφάνιση εικόνας με το λιγότερο σημαντικό bit της αρχικής εικόνας
Γραμμή 131: Μετατροπή εικόνας με το πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 132: Αποθήκευση εικόνας με το πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 134: Μετατροπή εικόνας με το δεύτερο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 135: Αποθήκευση εικόνας με το δεύτερο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 137: Μετατροπή εικόνας με το τρίτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 138: Αποθήκευση εικόνας με το τρίτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 140: Μετατροπή εικόνας με το τέταρτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 141: Αποθήκευση εικόνας με το τέταρτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 143: Μετατροπή εικόνας με το πέμπτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 144: Αποθήκευση εικόνας με το πέμπτο πιο σημαντικό bit της αρχικής εικόνας
Γραμμή 146: Μετατροπή εικόνας με το έκτο πιο σημαντικό bit της αρχικής εικόνας

Γραμμή 147: Αποθήκευση εικόνας με το έκτο πιο σημαντικό bit της αρχικής εικόνας

Γραμμή 149: Μετατροπή εικόνας με το έβδομο πιο σημαντικό bit της αρχικής εικόνας

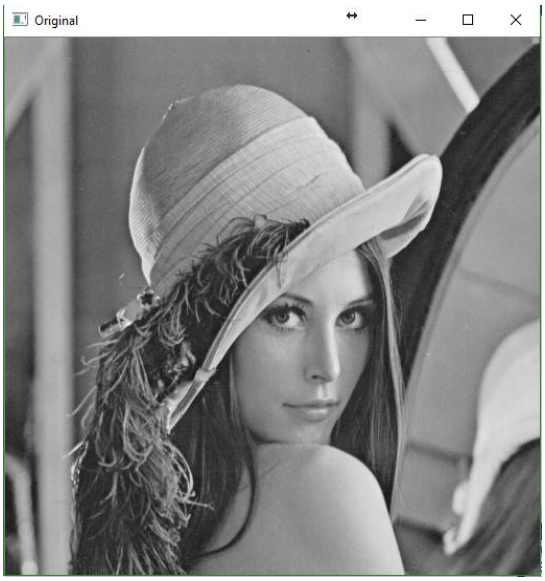
Γραμμή 150: Αποθήκευση εικόνας με το έβδομο πιο σημαντικό bit της αρχικής εικόνας

Γραμμή 152: Μετατροπή εικόνας με το λιγότερο σημαντικό bit της αρχικής εικόνας

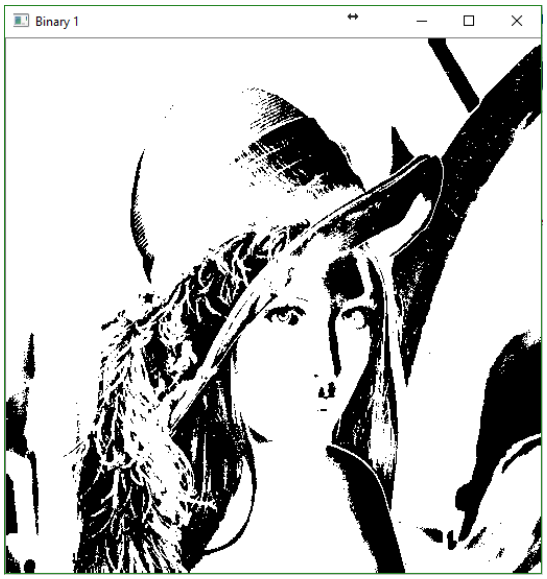
Γραμμή 153: Αποθήκευση εικόνας με το λιγότερο σημαντικό bit της αρχικής εικόνας

Γραμμή 156: Αναμονή για οποιαδήποτε πλήκτρο από τον χρήστη για τερματισμό του προγράμματος

2.9.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.20: Αρχική εικόνα.



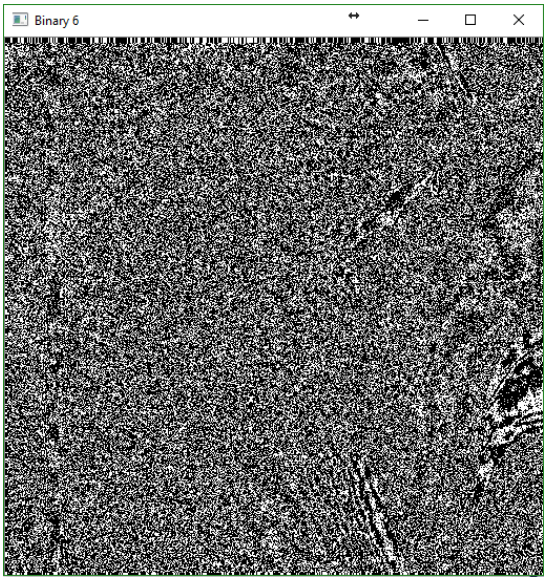
Εικόνα 2.21: Δυαδική εικόνα από το συντελεστή a_7.

Εικόνα 2.22: Δυαδική εικόνα από το συντελεστή a_6.

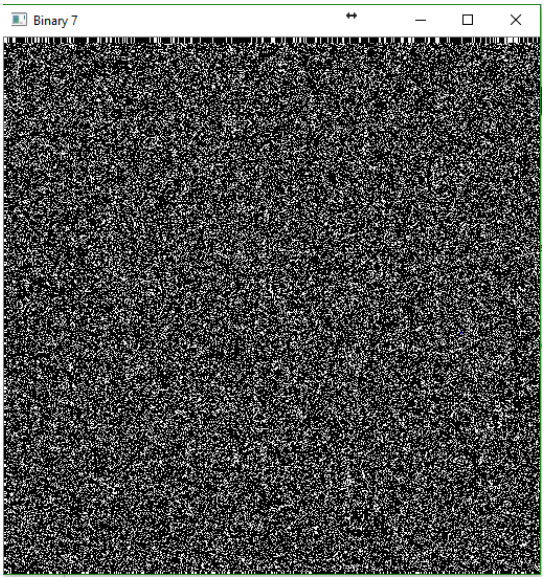
Εικόνα 2.23: Δυαδική εικόνα από το συντελεστή a_5.

Εικόνα 2.24: Δυαδική εικόνα από το συντελεστή a_4.

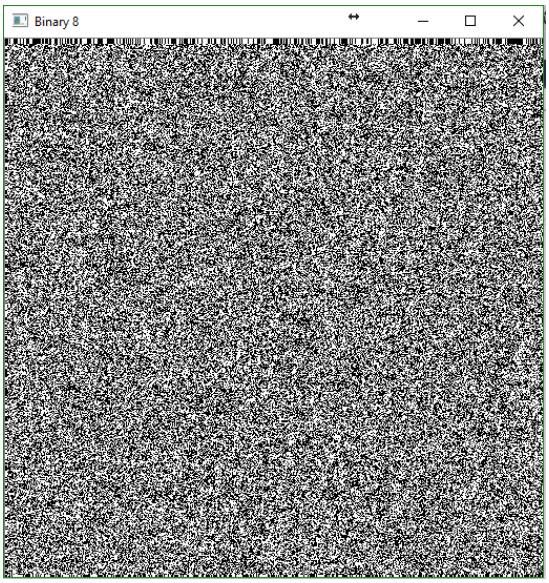
Εικόνα 2.25: Δυαδική εικόνα από το συντελεστή a_3.



Εικόνα 2.26: Δυαδική εικόνα από το συντελεστή a_2.



Εικόνα 2.27: Δυαδική εικόνα από το συντελεστή a_1.



Εικόνα 2.28: Δυαδική εικόνα από το συντελεστή a_0.

2.10 ΦΙΛΤΡΟ LAPLACIAN

2.10.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια, εφαρμόζεται σε αυτήν το φίλτρο Laplacian, με τη χρήση της συνάρτησης Laplacian(), και εμφανίζονται ως αποτέλεσμα δύο εικόνες. Η μία είναι η εικόνα που προέκυψε από το φιλτράρισμα η οποία αναδεικνύει τις ακμές της εικόνας εισόδου και η άλλη είναι η τελική εικόνα, η οποία είναι το αποτέλεσμα της αφαίρεσης της εικόνας που προέκυψε από το φιλτράρισμα από την αρχική εικόνα. Τέλος, αποθηκεύονται στο δίσκο οι δύο αυτές εικόνες.

2.10.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14        Mat image = imread("C:\lena.bmp", 0);
15        Mat laplace_im = image.clone();
16
17        if (image.empty())//check if empty
18        {
19                printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20                Sleep(10000); //10 second delay for showing message to user
21        }
22        else
23        {
24                Laplacian(image, laplace_im, image.depth(), 3);
25
26                Mat laplace = image - laplace_im;
27
28                namedWindow("Original Image", 1);
29                imshow("Original Image", image);
30
```

```
31                  namedWindow("Image with Laplacian", 1);
32                  imshow("Image with Laplacian", laplace_im);
33
34                  namedWindow("Image after Laplacian", 1);
35                  imshow("Image after Laplacian", laplace);
36
37                  //convert image to [0-255] in order to save
38                  laplace.convertTo(laplace_im, CV_64F, 255.0 / 1.0);
39                  imwrite("saved_image.png", laplace_im);
40
41                  //convert image to [0-255] in order to save
42                  laplace.convertTo(laplace, CV_64F, 255.0 / 1.0);
43                  imwrite("saved_image.png", laplace);
44
45                  waitKey();
46          }
47          return 0;
48 }
```

2.10.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15: Κλωνοποίηση εικόνας

Γραμμή 17: Έλεγχος για ορθό διάβασμα εικόνας

Γραμμή 19: Εάν η εικόνα δεν διαβάστηκε σωστά, εμφάνιση αντίστοιχου μηνύματος

Γραμμή 20: Αναμονή 10 δευτερόλεπτα για εμφάνιση προηγούμενου μηνύματος

Γραμμή 24:Εφαρμογή φίλτρου Laplacian στην εικόνα image

Γραμμή 26: Αποθήκευση στην εικόνα laplace, την τιμή της αφαίρεσης της τιμής φωτεινότητας από την αρχική εικόνα μείον την εικόνα με φίλτρο laplace

Γραμμή 29: Εμφάνιση αρχικής εικόνας

Γραμμή 32: Εμφάνιση εικόνας με φίλτρο laplace (ακμές)

Γραμμή 35: Εμφάνιση τελικής εικόνας έπειτα από την εφαρμογή του φίλτρου

Γραμμή 38: Μετατροπή εικόνας με φίλτρο laplace στο διάστημα [0,255]

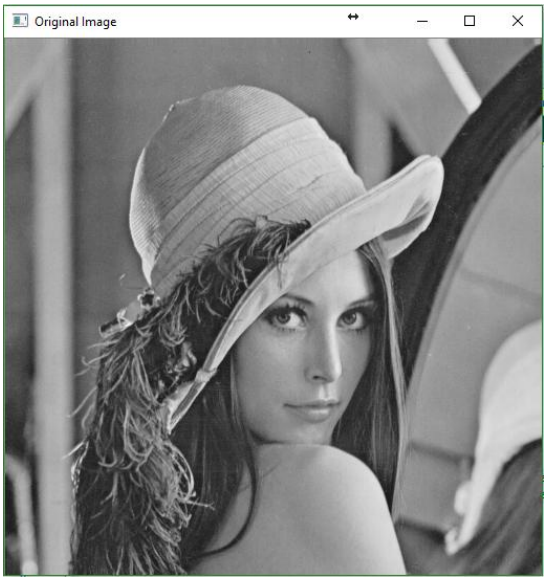
Γραμμή 39: Αποθήκευση εικόνας

Γραμμή 42: Μετατροπή τελικής εικόνας στο διάστημα [0,255]

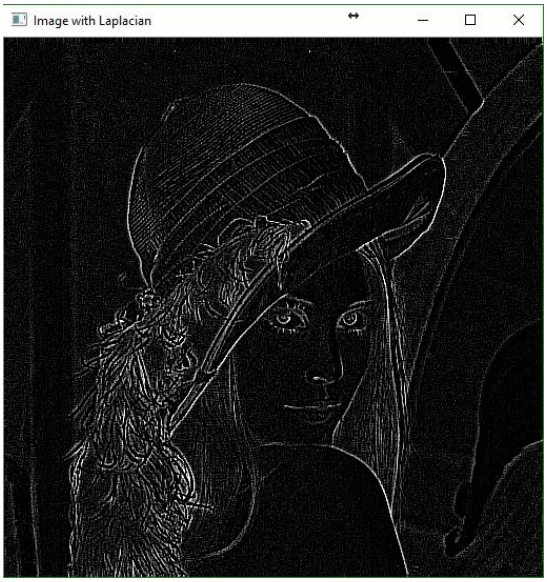
Γραμμή 43: Αποθήκευση τελικής εικόνας

Γραμμή 45: Αναμονή για πίεση οποιουδήποτε πλήκτρου για τερματισμό του προγράμματος

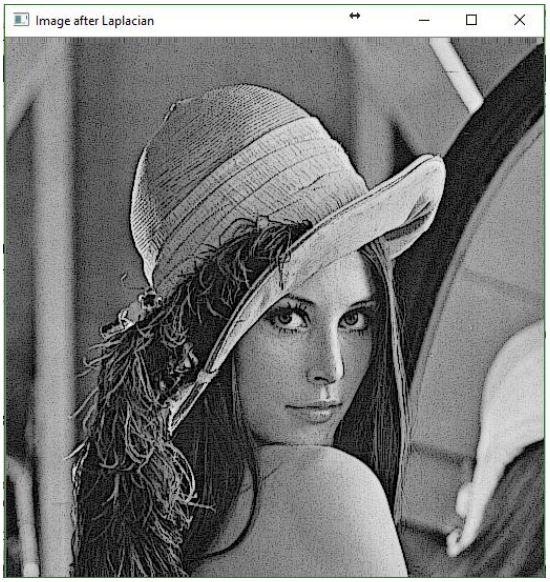
2.10.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.29: Αρχική εικόνα.



Εικόνα 2.30: Ανάδειξη των ακμών.



Εικόνα 2.31: Τελική εικόνα με ευκρινείς ακμές.

2.11 ΦΙΛΤΡΟ ΠΡΩΤΗΣ ΠΑΡΑΓΩΓΟΥ(SOBEL)

2.11.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια, εφαρμόζεται σε αυτήν το φίλτρο πρώτης παραγώγου, με τη χρήση της συνάρτησης Sobel(), και εμφανίζονται ως αποτέλεσμα δύο εικόνες. Η μία είναι η εικόνα που προέκυψε από την εφαρμογή του φίλτρου κάθετης πρώτης παραγώγου, η οποία προσδιορίζει τις οριζόντιες ακμές της εικόνας και η άλλη είναι η τελική εικόνα που προέκυψε από την εφαρμογή του φίλτρου οριζόντιας πρώτης παραγώγου, η οποία προσδιορίζει τις κάθετες ακμές της εικόνας. Τέλος, αποθηκεύονται στο δίσκο οι δύο αυτές εικόνες.

2.11.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14      Mat image = imread("C:\cameraman.tif", 0);
15      Mat sobel_im = image.clone();
16      Mat sobel_im_2 = image.clone();
17
18      if (image.empty())//check if empty
19      {
20            printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
21            Sleep(10000); //10 second delay for showing message to user
22      }
23      else
24      {
25            Sobel(image, sobel_im, image.depth(), 0, 1, 3);
26            Sobel(image, sobel_im_2, image.depth(), 1, 0, 3);
27
28            namedWindow("Original Image", 1);
29            imshow("Original Image", image);
```

```
30
31                  namedWindow("Image with Sobel katheto", 1);
32                  imshow("Image with Sobel katheto", sobel_im);
33
34                  namedWindow("Image with Sobel orizontio", 1);
35                  imshow("Image with Sobel orizontio", sobel_im_2);
36
37                  //convert image to [0-255] in order to save
38                  sobel_im.convertTo(sobel_im, CV_64F, 255.0 / 1.0);
39                  imwrite("saved_image.png", sobel_im);
40
41                  sobel_im_2.convertTo(sobel_im_2, CV_64F, 255.0 / 1.0);
42                  imwrite("saved_image2.png", sobel_im_2);
43
44                  waitKey();
45          }
46          return 0;
47 }
```

2.11.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15 και 16: Κλωνοποίηση αρχικής εικόνας

Γραμμή 18: Έλεγχος για ορθό διάβασμα εικόνας

Γραμμή 20: Εάν η εικόνα δεν διαβάστηκε σωστά, εμφάνιση αντίστοιχου μηνύματος

Γραμμή 21: Αναμονή 10 δευτερόλεπτα για εμφάνιση προηγούμενου μηνύματος

Γραμμή 25: Εφαρμογή φίλτρου Sobel για έμφαση στις κάθετες γραμμές

Γραμμή 26: Εφαρμογή φίλτρου Sobel για έμφαση στις οριζόντιες γραμμές

Γραμμή 29: Εμφάνιση αρχικής εικόνας

Γραμμή 32: Εμφάνιση εικόνας με έμφαση στις κάθετες γραμμές

Γραμμή 35: Εμφάνιση εικόνας με έμφαση στις οριζόντιες γραμμές

Γραμμή 38: Μετατροπή εικόνας με έμφαση στις κάθετες γραμμές, στο διάστημα [0,255]

Γραμμή 39: Αποθήκευση εικόνας

Γραμμή 41: Μετατροπή εικόνας με έμφαση στις οριζόντιες γραμμές, στο διάστημα [0,255]

Γραμμή 42: Αποθήκευση εικόνας

Γραμμή 44: Αναμονή για πίεση οποιουδήποτε πλήκτρου για τερματισμό του προγράμματος

2.11.4 Εκτέλεση-Αποτελέσματα προγράμματος

Εικόνα 2.32: Αρχική εικόνα.

Εικόνα 2.33: Οριζόντιες ακμές με το φίλτρο κάθετης πρώτης παραγώγου.

Εικόνα 2.34: Κάθετες ακμές με το φίλτρο οριζόντιας πρώτης παραγώγου.

2.12 IMNOISE (ΠΡΟΣΘΗΚΗ ΘΟΡΥΒΟΥ)

2.12.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια ζητείται από τον χρήστη να πληκτρολογήσει έναν αριθμό στο διάστημα [0-1], για την προσθήκη θορύβου. Έπειτα, εμφανίζεται η εικόνα με τον θόρυβο και αποθηκεύεται το αποτέλεσμα σε νέα εικόνα στο δίσκο.

2.12.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14	Mat image = imread("C:\lena.bmp", 0);
15	Mat result = image.clone();
16
17	if (image.empty())//check if empty
18	{
19		printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20		Sleep(10000); //10 second delay for showing message to user
21	}
22	else
23	{
24		double value;
25
26		printf("Give double for noise [0.00-1.00] :\n");
27		cin >> value;
28
29		if (value >= 0.00 && value <= 1.00)
30		{
31			// imGray is the grayscale of the input image
32			Mat noise = Mat(image.size(), CV_64F);
33			normalize(image, result, 0.0, 1.0, CV_MINMAX, CV_64F);
```

```
34                              randn(noise, 0, value);
35
36                              result = result + noise;
37                              normalize(result, result, 0.0, 1.0, CV_MINMAX, CV_64F);
38                              imshow("Output image with noise", result);
39
40                              namedWindow("Original Image", 1);
41                              imshow("Original Image", image);
42
43                              //convert image to [0-255] in order to save
44                              result.convertTo(result, CV_64F, 255.0 / 1.0);
45                              imwrite("saved_image.png", result);
46
47                              waitKey();
48                      }
49                      else
50                      {
51                              printf("Wrong input! Try again!\n");
52                      }
53              }
54              return 0;
55 }
```

2.12.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15: Κλωνοποίηση εικόνας

Γραμμές 17 έως 21: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 24: Ορισμός μεταβλητής

Γραμμή 26: Εμφάνιση μηνύματος στον χρήστη

Γραμμή 27: Διάβασμα από το πληκτρολόγιο

Γραμμή 29: Έλεγχος εάν η τιμή από το πληκτρολόγιο είναι μεταξύ 0 και 1

Γραμμή 32: Δημιουργία εικόνας με το ίδιο μέγεθος με την αρχική

Γραμμή 33: Εξισορρόπηση εικόνας

Γραμμή 34: Δημιουργία ψευδοτυχαίου θορύβου

Γραμμή 36: Προσθήκη θορύβου στην εικόνα

Γραμμή 37: Εξισορρόπηση τελικής εικόνας με θόρυβο

Γραμμή 38: Εμφάνιση εικόνας με θόρυβο

Γραμμή 41: Εμφάνιση αρχικής εικόνας

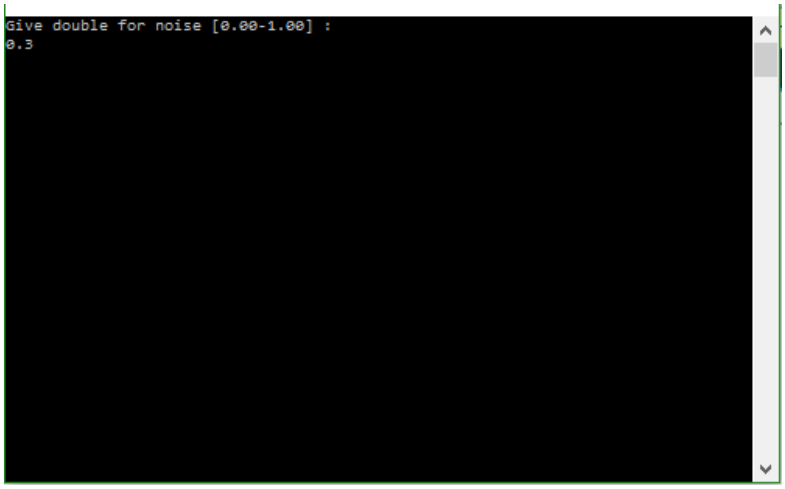
Γραμμή 44: Μετατροπή εικόνας στο διάστημα [0,255] για αποθήκευση

Γραμμή 45: Αποθήκευση εικόνας

Γραμμή 47: Αναμονή για πάτημα οποιουδήποτε πλήκτρου για τερματισμό του προγράμματος

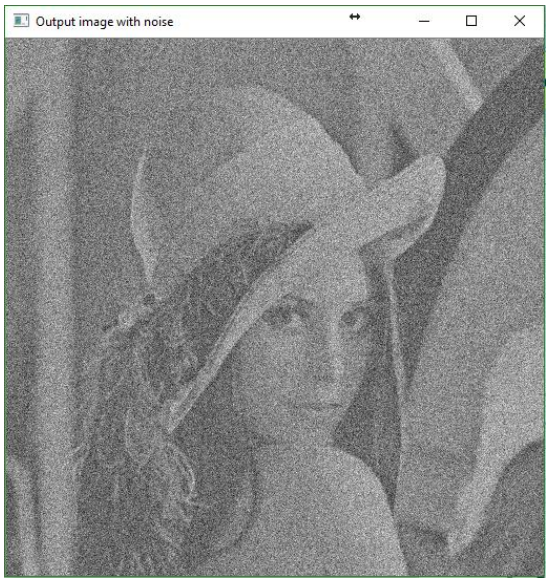
Γραμμή 51: Εφόσον η τιμή που έδωσε ο χρήστης δεν είναι στο [0,1], εμφάνιση μηνύματος λάθους

2.12.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.35: Κατά την εκτέλεση δίνουμε την τιμή 0.3 για την προσθήκη θορύβου στην εικόνα.

Εικόνα 2.36: Αρχική εικόνα.



Εικόνα 2.37: Εικόνα μετά από την προσθήκη θορύβου.

2.13 ΦΙΛΤΡΟ ΑΡΙΘΜΗΤΙΚΗΣ ΜΕΣΗΣ ΤΙΜΗΣ

2.13.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια ζητείται από τον χρήστη να πληκτρολογήσει έναν αριθμό στο διάστημα [0-1], για την προσθήκη θορύβου στην αρχική εικόνα. Έπειτα, για να επιτευχθεί η απαλοιφή του θορύβου, εφαρμόζεται στην εικόνα με τον θόρυβο, με την χρήση της συνάρτησης blur(), το φίλτρο αριθμητικής μέσης τιμής. Τέλος, εμφανίζεται η εικόνα με τον θόρυβο καθώς και η εικόνα μετά την χρήση του φίλτρου η οποία αποθηκεύεται και στο δίσκο.

2.13.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14        Mat image = imread("C:\cameraman.tif", 0);
15        Mat result = image.clone();
16
17        if (image.empty())//check if empty
18        {
19                printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
20                Sleep(10000); //10 second delay for showing message to user
21        }
22        else
23        {
24                double value;
25
26                printf("Give double for noise [0.00-1.00] :\n");
27                cin >> value;
28
29                if (value >= 0.00 && value <= 1.00)
30                {
```

```
31                          // imGray is the grayscale of the input image
32                          Mat noise = Mat(image.size(), CV_64F);
33                          normalize(image, result, 0.0, 1.0, CV_MINMAX, CV_64F);
34                          randn(noise, 0, value);
35
36                          result = result + noise * 1;
37                          normalize(result, result, 0.0, 1.0, CV_MINMAX, CV_64F);
38                          imshow("Output image with noise", result);
39
40                          blur(result, result, cv::Size(3, 3));
41
42                          namedWindow("Image blur", 1);
43                          imshow("Image blur", result);
44
45                          namedWindow("Original Image", 1);
46                          imshow("Original Image", image);
47
48                          //convert image to [0-255] in order to save
49                          result.convertTo(result, CV_64F, 255.0 / 1.0);
50                          imwrite("saved_image.png", result);
51
52                          waitKey();
53                  }
54                  else
55                  {
56                          printf("Wrong input! Try again!\n");
57                  }
58          }
59          return 0;
60 }
```

2.13.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15: Κλωνοποίηση εικόνας

Γραμμές 17 έως 21: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 24: Ορισμός μεταβλητής

Γραμμή 26: Εμφάνιση μηνύματος στον χρήστη

Γραμμή 27: Διάβασμα από το πληκτρολόγιο

Γραμμή 29: Έλεγχος εάν η τιμή από το πληκτρολόγιο είναι μεταξύ 0 και 1

Γραμμή 32: Δημιουργία εικόνας με το ίδιο μέγεθος με την αρχική

Γραμμή 33: Εξισορρόπηση εικόνας

Γραμμή 34: Δημιουργία ψευδοτυχαίου θορύβου

Γραμμή 36: Προσθήκη θορύβου στην εικόνα

Γραμμή 37: Εξισορρόπηση τελικής εικόνας με θόρυβο

Γραμμή 38: Εμφάνιση εικόνας με θόρυβο

Γραμμή 40: Εφαρμογή φίλτρου με χρήση της συνάρτησης blur με μάσκα 3x3

Γραμμή 43: Εμφάνιση εικόνας μετά τη χρήση του φίλτρου

Γραμμή 46: Εμφάνιση αρχικής εικόνας

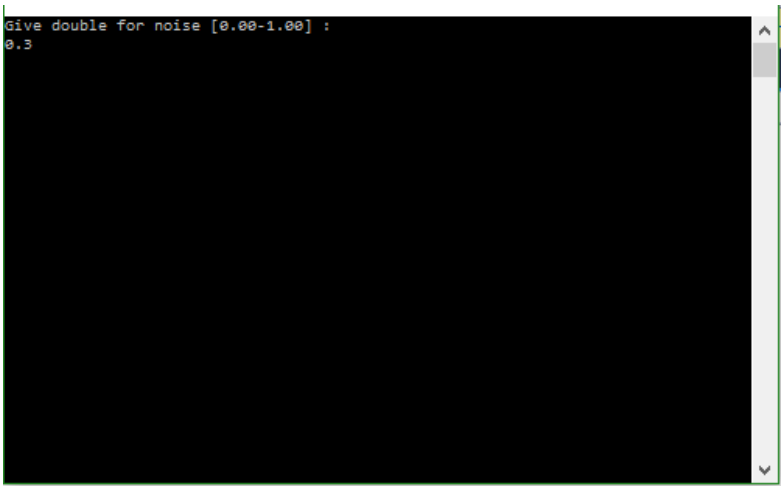
Γραμμή 49: Μετατροπή εικόνας στο διάστημα [0,255] για αποθήκευση

Γραμμή 50: Αποθήκευση εικόνας

Γραμμή 52: Αναμονή για πάτημα οποιουδήποτε πλήκτρου για τερματισμό του προγράμματος

Γραμμή 56: Εφόσον η τιμή που έδωσε ο χρήστης δεν είναι στο [0,1], εμφάνιση μηνύματος λάθους

2.13.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.38: Κατά την εκτέλεση δίνουμε την τιμή 0.3 για την προσθήκη θορύβου στην εικόνα.

Εικόνα 2.39: Αρχική εικόνα.

Εικόνα 2.40: Εικόνα μετά την προσθήκη θορύβου.

Εικόνα 2.41: Εικόνα μετά την χρήση του φίλτρου αριθμητικής μέσης τιμής.

2.14 ΦΙΛΤΡΟ ΜΕΣΑΙΑΣ ΤΙΜΗΣ (MEDIAN)

2.14.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια ζητείται από τον χρήστη να πληκτρολογήσει έναν αριθμό στο διάστημα [0-1], για την προσθήκη θορύβου στην αρχική εικόνα. Έπειτα, για να επιτευχθεί η απαλοιφή του θορύβου, εφαρμόζεται στην εικόνα με τον θόρυβο, με την χρήση της συνάρτησης medianblur(), το φίλτρο μεσαίας τιμής. Τέλος, εμφανίζεται η εικόνα με τον θόρυβο καθώς και η εικόνα μετά την χρήση του φίλτρου η οποία αποθηκεύεται και στο δίσκο.

2.14.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14	Mat image = imread("cameraman.tif", 0);
15	Mat result = image.clone();
16
17
18	if (image.empty())//check if empty
19	{
20		printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
21		Sleep(10000); //10 second delay for showing message to user
22	}
23	else
24	{
25
26		double value;
27
28		printf("Give double for noise [0.00-1.00] :\n");
29		cin >> value;
30
```

```
31                  if (value >= 0.00 && value <= 1.00)
32                  {
33                              // imGray is the grayscale of the input image
34                              Mat noise = Mat(image.size(), CV_64F);
35                              normalize(image, result, 0.0, 1.0, CV_MINMAX, CV_64F);
36                              randn(noise, 0, value);
37
38                              result = result + noise;
39                              normalize(result, result, 0.0, 1.0, CV_MINMAX, CV_64F);
40                              imshow("Output image with noise", result);
41
42                              Mat med = result.clone();
43
44                              med.convertTo(med, CV_8U, 255.0 / 1.0);
45
46                              medianBlur(med, med, 5);
47
48                              namedWindow("Image with median", 1);
49                              imshow("Image with median", med);
50
51                              namedWindow("Original Image", 1);
52                              imshow("Original Image", image);
53
54                              //convert image to [0-255] in order to save
55                              med.convertTo(med, CV_64F, 255.0 / 1.0);
56                              imwrite("saved_image.png", med);
57
58                              waitKey();
59                  }
60                  else
61                  {
62                              printf("Wrong input! Try again!\n");
63                  }
64
65          }
66          return 0;
67 }
```

2.14.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15: Κλωνοποίηση εικόνας

Γραμμές 18 έως 22: Έλεγχος για ορθό διάβασμα της εικόνας

Γραμμή 26: Ορισμός μεταβλητής

Γραμμή 28: Εμφάνιση μηνύματος στον χρήστη

Γραμμή 29: Διάβασμα από το πληκτρολόγιο

Γραμμή 31: Έλεγχος εάν η τιμή από το πληκτρολόγιο είναι μεταξύ 0 και 1

Γραμμή 34: Δημιουργία εικόνας με το ίδιο μέγεθος με την αρχική

Γραμμή 35: Εξισορρόπηση εικόνας

Γραμμή 36: Δημιουργία ψευδοτυχαίου θορύβου

Γραμμή 38: Προσθήκη θορύβου στην εικόνα

Γραμμή 39: Εξισορρόπηση τελικής εικόνας με θόρυβο

Γραμμή 40: Εμφάνιση εικόνας με θόρυβο

Γραμμή 42: Κλωνοποίηση εικόνας

Γραμμή 44: Μετατροπή εικόνας στο διάστημα [0,255]

Γραμμή 46: Εφαρμογή φίλτρου median με μάσκα 5x5

Γραμμή 49: Εμφάνιση εικόνας έπειτα από την εφαρμογή του φίλτρου median

Γραμμή 52: Εμφάνιση αρχικής εικόνας

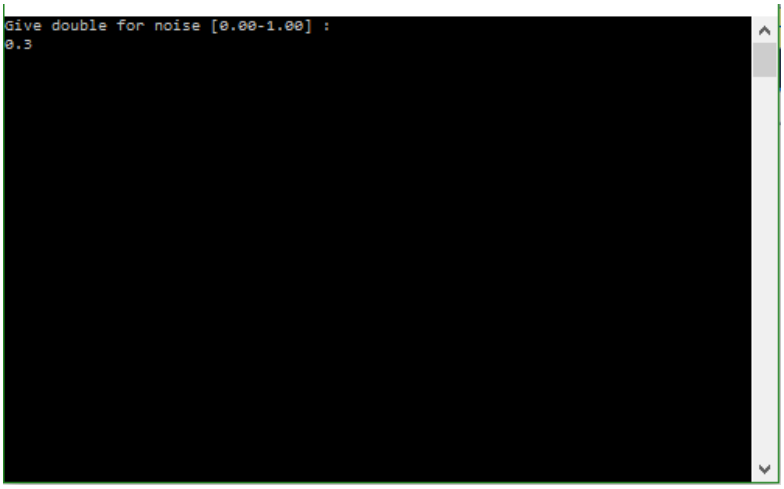
Γραμμή 55: Μετατροπή εικόνας στο διάστημα [0,255]

Γραμμή 56: Αποθήκευση τελικής εικόνας

Γραμμή 58: Αναμονή για πάτημα οποιουδήποτε πλήκτρου για τερματισμό του προγράμματος

Γραμμή 62: Εφόσον η τιμή που έδωσε ο χρήστης δεν είναι στο [0,1], εμφάνιση μηνύματος λάθους

2.14.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.42: Κατά την εκτέλεση δίνουμε την τιμή 0.3 για την προσθήκη θορύβου στην εικόνα.

Εικόνα 2.43: Αρχική εικόνα.

Εικόνα 2.44: Εικόνα μετά την προσθήκη θορύβου.

Εικόνα 2.45: Εικόνα μετά την χρήση του φίλτρου μεσαίας τιμής.

2.15 ΕΠΕΚΤΑΣΗ ΚΑΙ ΣΥΡΡΙΚΝΩΣΗ ΕΙΚΟΝΑΣ (DILATE-ERODE)

2.15.1 Λειτουργία προγράμματος

Κατά την εκτέλεση του προγράμματος διαβάζεται μία εικόνα. Στη συνέχεια, για να επιτευχθεί η επέκταση της αρχικής εικόνας γίνεται χρήση της συνάρτησης dilate() και για την συρρίκνωση χρησιμοποιείται η συνάρτηση erode(). Τέλος, εμφανίζονται οι εικόνες μετά την επέκταση και τη συρρίκνωση και αποθηκεύονται στο δίσκο.

2.15.2 Κώδικας προγράμματος

```
1 #include "opencv2/objdetect/objdetect.hpp"
2 #include "opencv2/highgui/highgui.hpp"
3 #include "opencv2/imgproc/imgproc.hpp"
4 #include "opencv2/core/core.hpp"
5 #include <opencv/cv.h>
6 #include <iostream>
7 #include <Windows.h>
8
9 using namespace std;
10using namespace cv;
11
12int main(int, char**)
13{
14	Mat image = imread("rice.png", 0);
15	Mat dilate_im = image.clone();
16	Mat erode_im = image.clone();
17
18	if (image.empty())//check if empty
19	{
20		printf("Image is not read! File is probably missing! Press any key to exit program");//message for error
21		waitKey(); //10 second delay for showing message to user
22	}
23	else
24	{
25
26		namedWindow("Original", 1);
27		imshow("Original", image);
28
29		dilate(image, dilate_im, Mat());
30		erode(image, erode_im, Mat());
31
32		namedWindow("Dilate Image", 1);
```

```
33                  imshow("Dilate Image", dilate_im);
34
35                  namedWindow("Erode Image", 1);
36                  imshow("Erode Image", erode_im);
37
38                  //convert image to [0-255] in order to save
39                  dilate_im.convertTo(dilate_im, CV_64F, 255.0 / 1.0);
40                  imwrite("saved_image.png", dilate_im);
41
42                  erode_im.convertTo(erode_im, CV_64F, 255.0 / 1.0);
43                  imwrite("saved_image1.png", erode_im);
44
45                  waitKey();
46          }
47          return 0;
48 }
```

2.15.3 Ανάλυση κώδικα προγράμματος

Γραμμές 1 έως 7: Εισαγωγή βιβλιοθηκών.

Γραμμές 9 και 10: Χρήση cv και std namespaces

Γραμμή 14: Διάβασμα εικόνας

Γραμμή 15 και 16: Κλωνοποίηση εικόνας

Γραμμή 18: Έλεγχος για ορθό διάβασμα εικόνας

Γραμμή 20: Εάν η εικόνα δεν διαβάστηκε σωστά, εμφάνιση αντίστοιχου μηνύματος

Γραμμή 21: Αναμονή για πίεση οποιουδήποτε πλήκτρου για τερματισμό προγράμματος

Γραμμή 27: Εμφάνιση αρχικής εικόνας

Γραμμή 29: Χρήση συνάρτησης dilate (επέκταση φωτεινότητας)

Γραμμή 30: Χρήση συνάρτησης erode (συρρίκνωση φωτεινότητας)

Γραμμή 33: Εμφάνιση εικόνας έπειτα από τη χρήση της dilate

Γραμμή 36: Εμφάνιση εικόνας έπειτα από τη χρήση της erode

Γραμμή 39: Μετατροπή εικόνας dilate για αποθήκευση

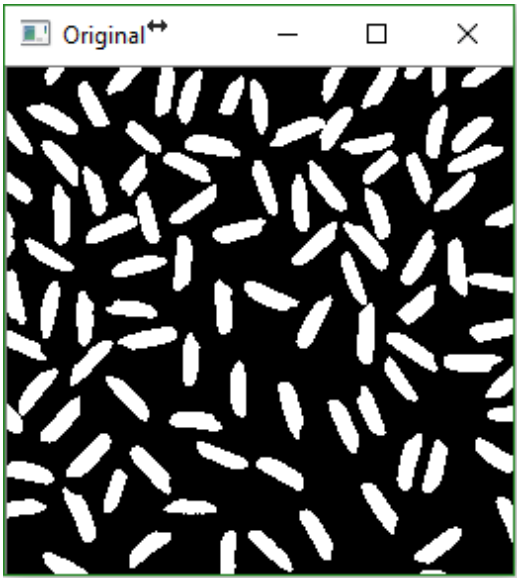
Γραμμή 40: Αποθήκευση εικόνας

Γραμμή 42: Μετατροπή εικόνας erode για αποθήκευση

Γραμμή 43: Αποθήκευση εικόνας

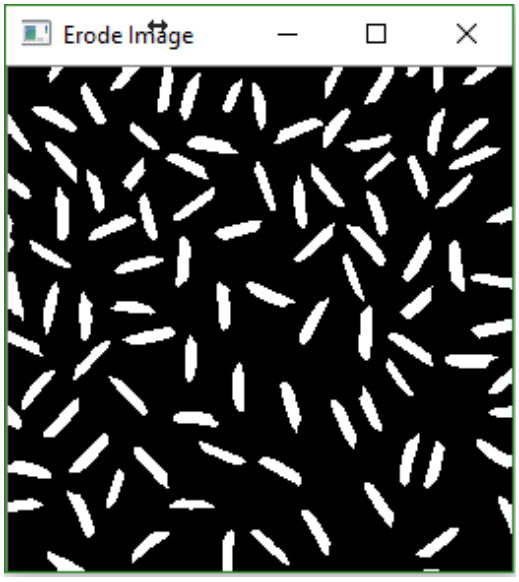
Γραμμή 45: Αναμονή για πίεση οποιουδήποτε πλήκτρου για τερματισμό προγράμματος

2.15.4 Εκτέλεση-Αποτελέσματα προγράμματος



Εικόνα 2.46: Αρχική εικόνα.

Εικόνα 2.47: Εικόνα μετά την επέκταση.



Εικόνα 2.48: Εικόνα μετά την συρρίκνωση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Ιωάννης Ν. Έλληνας, «Ψηφιακή Επεξεργασία Εικόνας και Βίντεο Από την Θεωρία στην Πράξη», Αυτοέκδοση Αθήνα 2010

[2] OpenCV για Windows:
http://opencv.org/downloads.html

[3] stack overflow:
https://stackoverflow.com/

[4] opencv cheat sheet:
http://docs.opencv.org/2.4/opencv_cheatsheet.pdf

[5] opencv structures:
http://docs.opencv.org/2.4/modules/core/doc/basic_structures.html