



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ
ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ, ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



Εργαστήριο Γνώσης και Αβεβαιότητας

Μεταπτυχιακή Εργασία

Ομομορφικό φιλτράρισμα έγχρωμων εικόνων

Φωτεινή Παχτίτη
202220182025

Επιβλέπων:

Μανόλης Γουάλλες
Αναπληρωτής Καθηγητής

Τρίπολη, Μάρτιος, 2022

Εγκρίθηκε από την εξεταστική επιτροπή την 15η Μαρτίου 2022.

Μανόλης Γουάλλες
Αναπληρωτής Καθηγητής

Κωνσταντίνος Βασιλάκης
Καθηγητής

Γεώργιος Λέπουρας
Καθηγητής

.....

Φωτεινή Παχτίτη
Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Πελοποννήσου

Copyright © ,Φωτεινή Παχτίτη 2022
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευτεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πελοποννήσου.

Η παρούσα εργασία εκπονήθηκε σε συνεργασία με το Εργαστήριο Γνώσης και Αβεβαιότητας (ΓΑΒ LAB).

*Αφιερώνεται στην οικογένειά μου και σε όσους συνέβαλαν για την εκπόνηση της
παρούσης εργασίας.*

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον διδάσκοντα και επιβλέποντα της διπλωματικής μου εργασίας κ. Γουάλλες Μανόλη για την ευκαιρία που μου έδωσε να εργαστούμε για ακόμη μια φορά μαζί ειδικότερα πάνω σε αυτή την ενδιαφέρουσα έρευνα, αλλά και για την πάντα άμεση και πολύτιμη καθοδήγησή του σε όλα τα εμπόδια που προέκυψαν.

Επίσης, θα ήθελα να ευχαριστήσω τον Γιαννακάκη Κωνσταντίνο για τη βοήθεια που πρόσφερε στο debugging του κώδικα που αναπτύχθηκε.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου που είναι πάντα δίπλα μου και με στηρίζει στις επιλογές μου.

Περίληψη

Η επεξεργασία εικόνας αποτελεί ένα συχνό φαινόμενο τόσο για την επιστήμη των υπολογιστών, όσο και για την καθημερινή χρήση από έναν άνθρωπο. Στην επιστήμη των υπολογιστών χρησιμοποιείται ένα ευρύ φάσμα αλγορίθμων και μαθηματικών μοντέλων με σκοπό την επίτευξη της επεξεργασίας εικόνας. Αντίστοιχα, στην καθημερινή χρήση φωτογραφικών μηχανών ή κινητών από πολλούς χρήστες που φωτογραφίζουν και επιθυμούν την επεξεργασία και βελτίωση της εικόνας.

Στην παρούσα διπλωματική εργασία πρόκειται να δούμε την ανάλυση και την εφαρμογή του ομομορφικού φίλτρου σε μια εικόνα RPG και μια εικόνα Greyscale. Υπάρχουν αρκετά φίλτρα, τα οποία μπορούν να βοηθήσουν στην διαδικασία της επεξεργασίας μιας εικόνας, αλλά η επιλογή και η έρευνα έγινε για το ομομορφικό φιλτράρισμα που είναι μια τεχνική, η οποία αποσκοπεί στην επεξεργασία της εικόνας και του σήματος που εφαρμόζονται τεχνικές γραμμικού φίλτρου. Η υλοποίηση αναπτύχθηκε με Python.

Τέλος, αναλύεται η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη του κώδικα με χρήση ενός framework, ώστε να πραγματοποιηθεί με επιτυχία η εφαρμογή του ομομορφικού φίλτρου σε μία εικόνα.

Abstract

Image processing is a common phenomenon both in computer science and in everyday use. A wide range of algorithms and mathematical models are used in computer science to achieve image processing. Respectively, in the daily use of cameras or mobile phones by many users who take photos and want to edit and improve the image.

In the present study we are going to look at the analysis and application of the Homomorphic filter to an RPG image and a Greyscale image. There are several filters that can help in the process of processing an image, but the selection and the research was done for homomorphic filtering which is a technique that aims to process the image and signal using linear filtering techniques. The implementation was developed with Python.

Finally, the methodology followed for the development of the code is analyzed using the OpenCV framework, in order to successfully implement the homomorphic filter in an image.

Πίνακας περιεχομένων

Ευχαριστίες	iii
1 Εισαγωγή	1
2 Θεωρητικό υπόβαθρο	3
2.1 Εικόνα	3
2.2 Ψηφιακή επεξεργασία εικόνας	5
2.3 Χωρικό φιλτράρισμα	7
2.4 Φιλτράρισμα στο πεδίο της συχνότητας	8
2.5 Μετασχηματισμός Fourier	8
2.6 Εφαρμογή φίλτρων	11
2.7 Ομομορφικό φιλτράρισμα	13
2.8 Υπολογιστικά εργαλεία Python και OpenCV	14
3 Υπολογιστικό μέρος	17
3.1 Δομή υπολογιστικού κώδικα	17
3.2 Εισαγωγή βιβλιοθηκών και εικόνας	18
3.3 Ανάλυση και κανονικοποίηση εικόνας	18
3.4 Λογαρίθμηση και επιστροφή στην αρχική κλίμακα	21
3.5 Μετασχηματισμός Fourier και αντίστροφος μετασχηματισμός Fourier	22
3.6 Μεταφορά στοιχείων με μηδενικές συχνότητες στο κέντρο και επιστροφή	23
3.7 Εφαρμογή φίλτρου	24
3.8 Δημιουργία παραθύρου με τις μπάρες εργασιών	29
3.9 Αποθήκευση νέας εικόνας και έξοδος	31
3.10 Συνολική διαδικασία	31
4 Αποτελέσματα	33
4.1 Εκτέλεση κώδικα	33
4.2 Ακραίες τιμές παραμέτρων	34
4.3 Ομομορφικό φιλτράρισμα σε ασπρόμαυρη εικόνα	36
4.4 Ομομορφικό φιλτράρισμα σε έγχρωμη εικόνα	37
5 Συμπεράσματα	41
6 Παραρτήματα	43
6.1 Homomorphic Filtering Main Code	43
6.2 Homomorphic Filtering Utilities	44
Βιβλιογραφία	47

Κεφάλαιο 1

Εισαγωγή

Η εισαγωγή και η εκτεταμένη χρήση των ψηφιακών εικόνων στη βιομηχανία και στην καθημερινότητα των ανθρώπων αποτέλεσε την απαρχή της επεξεργασίας εικόνων ως επιστήμη. Η έννοια της επεξεργασίας σημαίνει την μετατροπή μιας αρχικής μορφής εικόνας σε μια τελική με σκοπό την αφαίρεση μη επιθυμητών στοιχείων, τα οποία ονομάζονται θόρυβοι. Ήδη από τις αρχές του 20ου αιώνα κατέστη φανερή η ανάγκη για την βελτίωση και τη διόρθωση της ποιότητας των εικόνων τόσο για λόγους αισθητικής όσο και για επαγγελματικούς σκοπούς όπου απαιτούνταν ψηφιακές εικόνες υψηλής ευκρίνειας [Sto72]. Θεμέλιος λίθος στην επεξεργασία εικόνων συνιστά η κατανόηση και η εμβάθυνση στην οπτική και την ανάλυση της ίδιας της εικόνας.

Σε μια από τις πιο βασικές μεθόδους κατάταξης, οι εικόνες κατηγοριοποιούνται με βάση την πηγή τους [Nam21]. Η πιο κοινή πηγή είναι το ηλεκτρομαγνητικό φάσμα και πιο συγκεκριμένα η ακτινοβολία X και το οπτικό φάσμα. Πέρα από την ηλεκτρομαγνητική ακτινοβολία, οι εικόνες μπορούν να δημιουργηθούν με βάση ηλεκτρονικές, ακουστικές ή υπέρηχες πηγές ενέργειας. Οι διαφορετικές πηγές εικόνων έχουν ως αποτέλεσμα τις διαφορετικές εφαρμογές των δημιουργούμενων προϊόντων. Για παράδειγμα, εικόνες που προέρχονται από ακτίνες Γ βρίσκουν εφαρμογή στην πυρηνική φυσική και στην αστρονομία. Οι ακτίνες X αποτελούν μια από τις παλαιότερες πηγές εικόνων, με σημαντική χρήση στη βιομηχανία και στην ιατρική, όπως η μέθοδος ακτινολογικής εξέτασης της αξονικής τομογραφίας. Η υπεριώδης ακτινοβολία χρησιμοποιείται στη βιολογία, στην κατασκευή laser και στην λιθογραφία, ενώ τα ραδιοκύματα είναι απαραίτητα στην διεξαγωγή της μαγνητικής τομογραφίας. Είναι καταφανές ότι η ολοένα και αυξανόμενη χρήση των εικόνων ενισχύει τις απαιτήσεις για υψηλότερη ακρίβεια και δυνατότητες επεξεργασίας. Πρακτικά η επεξεργασία εικόνων έγινε δυνατή με την ανάπτυξη της τεχνολογίας και συγκεκριμένα της επιστήμης υπολογιστών.

Στο πεδίο της πληροφορικής, η επεξεργασία εικόνων ανήκει στην ευρύτερη κατηγορία της επεξεργασίας σημάτων, η οποία εξετάζει την ανάλυση, την μορφοποίηση και την σύνθεση σημάτων όπως εικόνες, ήχοι και βίντεο. Το οπτικό αποτέλεσμα βασίζεται σε, μη ορατούς από το χρήστη, υπολογιστικούς αλγόριθμους και μαθηματικά μοντέλα που καθιστούν δυνατές τις απαραίτητες μετατροπές για τη μορφοποίηση μιας εικόνας.

Ο τομέας της επεξεργασίας εικόνων αποσκοπεί στην επίτευξη τριών βασικών στόχων [Στρ15]:

- Την ψηφιοποίηση αναλογικών εικόνων, την συμπίεση και την αποσυμπίεση,
- την διόρθωση, την βελτίωση, την αποκατάσταση και
- την ανάλυση, όπως την ανάκτηση βάθους ή την μεγέθυνση

Με βάση αυτούς τους άξονες αναπτύσσονται οι δυνατότητες που έχουμε σήμερα να χειριζόμαστε και να μεταβάλλουμε τις ψηφιακές εικόνες γύρω μας.

Κίνητρο για την συγκεκριμένη εργασία αποτέλεσε η μεγάλη σημασία της επεξεργασίας εικόνων στην σύγχρονη ψηφιακή κοινωνία και στις ανάγκες που απαντά η επιστήμη των υπολογιστών. Η μέθοδος που επιλέχτηκε να εξεταστεί είναι η επεξεργασία εικόνων με εφαρμογή φίλτρων ή μασκών και πιο συγκεκριμένα του ομομορφικού φιλτραρίσματος. Το ομομορφικό φιλτράρισμα

ανήκει στην κατηγορία των μορφοποιήσεων στο πεδίο της συχνότητας βασιζόμενο σε μη γραμμικούς μετασχηματισμούς. Η ιδιαιτερότητα που καθιστά μοναδικό το συγκεκριμένο φίλτρο είναι η ικανότητα του να εξομαλύνει την φωτεινότητα και να αυξάνει την αντίθεση ταυτόχρονα, στη χωρική διάσταση της εικόνας.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η δημιουργία του υπολογιστικού κώδικα που θα καθιστά δυνατό το ομομορφικό φιλτράρισμα RGB και Greyscale εικόνων. Ο κώδικας υλοποιήθηκε με τη χρήση της γλώσσας προγραμματισμού Python και η εφαρμογή του φίλτρου πραγματοποιήθηκε μέσω της βιβλιοθήκης OpenCV. Μέσω του κώδικα που αναλύεται, οι εικόνες που επιλέχθηκαν μπορούν να υποστούν την επιθυμητή επεξεργασία, μεταβάλλοντας τις τέσσερις παραμέτρους του ομομορφικού φιλτραρίσματος. Ως αποτέλεσμα η φωτεινότητα και η αντίθεση της εικόνας αλλάζουν όπως επιθυμεί ο χρήστης και, τέλος, η πλέον μορφοποιημένη εικόνα μπορεί να αποθηκευτεί.

Η δομή της διπλωματικής ακολουθεί την εξής αλληλουχία: Αρχικά, στο κεφάλαιο 2 αναλύεται το θεωρητικό υπόβαθρο που σχετίζεται με την επεξεργασία εικόνων και είναι απαραίτητο για την κατανόηση του υπολογιστικού μέρους. Στην συνέχεια, στο 3ο κεφάλαιο, περιγράφεται η μαθηματική ανάλυση και επεξηγείται ενδελεχώς ο κώδικας που υλοποιήθηκε. Στο κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα μετά τη χρήση του κώδικα. Τέλος, συνοψίζονται τα συμπεράσματα και οι συνολικές παρατηρήσεις σχετικά με την εργασία.

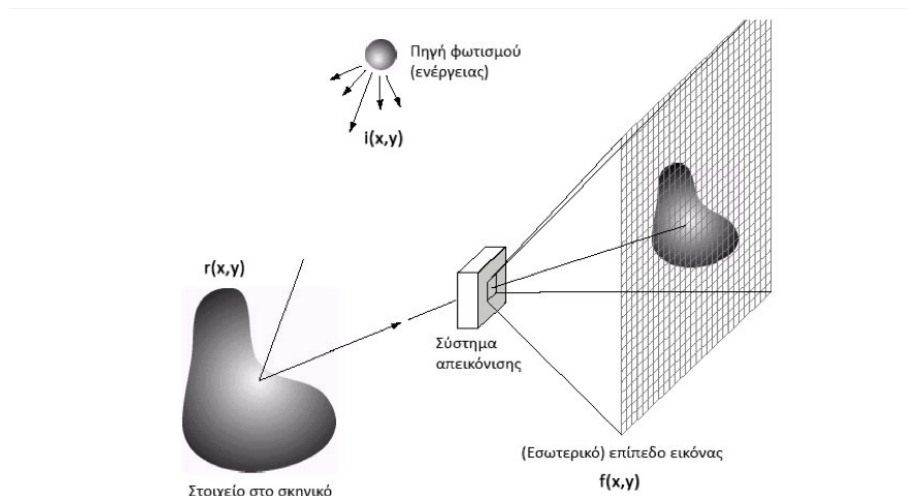
Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Εικόνα

Κατά την ανάλυση του σήματος που ονομάζεται εικόνα, παρατηρούνται οι ιδιαιτερότητες της μορφής της που έγκεινται σε δύο διαφορετικά χαρακτηριστικά [Ελλ21]. Αρχικά, οι εικόνες είναι σήματα μέσω των οποίων μετρούνται οι χωρικές διαστάσεις μιας παραμέτρου, σε αντίθεση με την πλειονότητα των σημάτων που βασίζονται στη διάσταση του χρόνου. Δεύτερον, οι εικόνες έχουν τη δυνατότητα αποθήκευσης μεγάλου όγκου πληροφορίας συγκριτικά με τα ηχητικά σήματα.

Μια δισδιάστατη εικόνα μπορεί να περιγραφεί από μια εξίσωση $f(x, y)$, όπου x, y είναι οι χωρικές συντεταγμένες στο δισδιάστατο επίπεδο [Nam21]. Η συνάρτηση f είναι ένα βαθμωτό μέγεθος, το οποίο αντιπροσωπεύει το μέγεθος της μεταβλητής που μετριέται και εξαρτάται από την φύση της εικόνας, συνήθως είναι το μέτρο της φωτεινότητας ή κάποιου χρώματος. Η εξίσωση $f(x, y)$ αναλύεται σε δύο στοιχεία, όπως παρουσιάζεται στην εικόνα 2.1.



Εικόνα 2.1. Συστατικά στοιχεία εικόνας [Nam21]

Τα στοιχεία της εικόνας είναι τα εξής:

- η φωτεινότητα (illumination) $i(x, y)$, που περιγράφει την ποσότητα της πηγής φωτισμού στο σκηνικό και
- η ανακλαστικότητα (reflectance) $r(x, y)$, που περιγράφει την ποσότητα του φωτισμού που ανακλάται από τα παρόντα αντικείμενα στο σκηνικό.

Μαθηματικά η εξίσωση $f(x, y)$ ορίζεται ως:

$$f(x, y) = i(x, y)r(x, y)$$

Όπου ισχύει,

$$0 < i(x, y) < \infty \text{ και } 0 < r(x, y) < 1$$

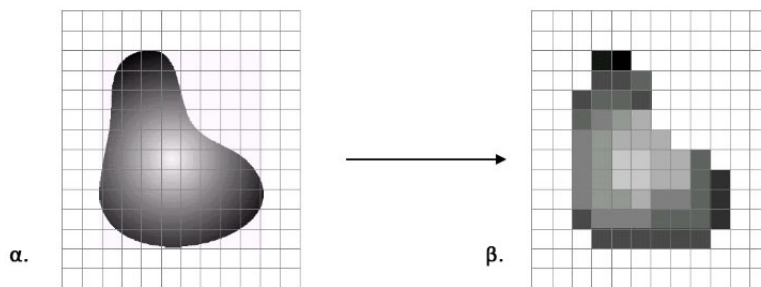
Η ανακλαστικότητα $r(x, y)$ εξαρτάται από τις ιδιότητες των αντικειμένων [Rao21]. Όταν λαμβάνει την τιμή μηδέν (0) σημαίνει πλήρη απορρόφηση του φωτός, ενώ στην τιμή ένα (1) υπάρχει πλήρης ανάκλαση. Ως χαρακτηριστικές τιμές φωτισμού μπορούν να θεωρηθούν οι 90,000 Im/mm² μια μέρα με ήλιο και 10,000 Im/mm² μια συννεφιασμένη μέρα. Ενώ τυπικές τιμές ανακλαστικότητας σε μπλε βελούδο είναι 0.01 και στο χιόνι 0.93.

Στην αναλογική εικόνα η εξίσωση $f(x, y)$ είναι συνεχής, σε αντίθεση με την ψηφιακή όπου οι μεταβλητές x, y και η παράμετρος f λαμβάνουν διακριτές, πεπερασμένες τιμές [GW21]. Η ψηφιακή εικόνα αποτελείται από ένα διακριτό αριθμό στοιχείων, κάθε ένα από τα οποία χαρακτηρίζεται από την τιμή και την θέση του στο καρτεσιανό επίπεδο. Τα στοιχεία αυτά ονομάζονται εικονοστοιχεία ή pixel. Ο αριθμός των εικονοστοιχείων είναι σημαντικός τόσο για το μέγεθος της εικόνας όσο και για την ποιότητά της. Για συγκριτικούς σκοπούς στην εικόνα 2.2. απεικονίζεται μια ψηφιακή εικόνα με διαφορετικούς αριθμούς εικονοστοιχείων.



Εικόνα 2.2. Σύγκριση εικόνων με διαφορετικό αριθμό εικονοστοιχείων [GW21]

Η απόκτηση των εικόνων σε αναλογική μορφή είναι συχνό φαινόμενο στην επεξεργασία εικόνων. Ο λόγος έγκειται στο ότι πολλοί αισθητήρες από τους οποίους προέρχεται το σήμα της εικόνας παράγουν ένα κύμα συνεχούς τάσης, του οποίου το πλάτος και η χωρική συμπεριφορά σχετίζονται με το φυσικό φαινόμενο που αναπαρίσταται, δηλαδή το περιεχόμενο που απεικονίζεται. Για την δημιουργία μιας ψηφιακής εικόνας είναι απαραίτητη η μετατροπή των συνεχών δεδομένων που έχουν μετρηθεί σε μια ψηφιακή μορφή. Η μετατροπή αυτή περιλαμβάνει τις διεργασίες της δειγματοληψίας και κβάντωσης σήματος (2.3).



Εικόνα 2.3. Συνεχής εικόνα (α), εικόνα μετά την δειγματοληψία και την κβάντωση σήματος (β) [Nam21]

Κατά την δειγματοληψία λαμβάνει χώρα η ψηφιοποίηση των συντεταγμένων της εικόνας (x, y), ενώ κατά την κβάντωση σήματος πραγματοποιείται η ψηφιοποίηση των τιμών του πλάτους

της εικόνας ($f(x, y)$). Η ποιότητα της ψηφιακής εικόνας σχετίζεται άμεσα από τον αριθμό των δειγμάτων και το βαθμό διακριτοποίησης του καρτεσιανού επιπέδου κατά την δειγματοληψία και την κβάντωση σήματος.

Ανάλογα με τα χρώματα που περιλαμβάνουν, οι εικόνες μπορούν να κατηγοριοποιηθούν σε τρία είδη όπως φαίνεται στην εικόνα 2.4 [Στρ15]:

- τις δυαδικές εικόνες (Binaryimages), οι οποίες περιλαμβάνουν μόνο δύο διαφορετικά χρώματα και το πιο κοινό ζεύγος χρωμάτων είναι λευκό – μαύρο,
- τις μονοχρωματικές με αποχρώσεις του γκρι (Greyscaleimages), στις οποίες το κάθε εικονοστοιχείο μπορεί να λάβει μια από τις 256 αποχρώσεις, με ακραίες τιμές το μαύρο και το λευκό και
- τις έγχρωμες εικόνες (Colorimages), οι οποίες αποτελούνται από το συνδυασμό τριών κύριων χρωμάτων, του κόκκινου, του μπλε και του πράσινου, γι' αυτό και ονομάζονται RGB (RedGreenBlue). Το κάθε χρώμα έχει 256 αποχρώσεις και το κάθε εικονοστοιχείο μπορεί να χρωματιστεί με έναν οποιοδήποτε συνδυασμό των αποχρώσεων των τριών χρωμάτων.



Εικόνα 2.4. Τυπικά παραδείγματα κατηγοριών εικόνων: έγχρωμη (α), μονοχρωματική με αποχρώσεις του γκρι (β) και δυαδική (γ) [Ελλ21]

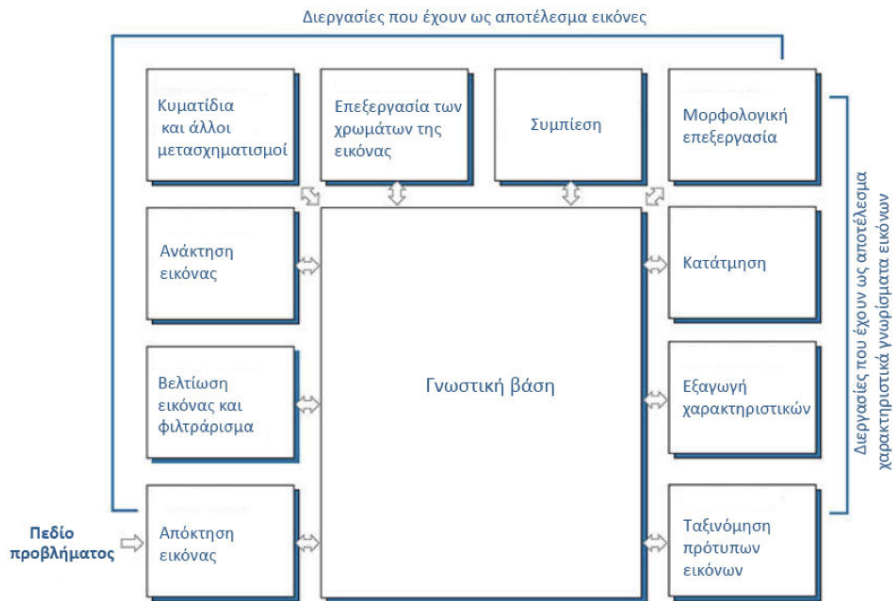
2.2 Ψηφιακή επεξεργασία εικόνας

Η θεμελιώδης αρχή της επεξεργασία εικόνων είναι η διεργασία που παρουσιάζεται στην εικόνα 2.5. Η υπό εξέταση εικόνα επιλέγεται, υφίσταται μια ή παραπάνω από τις δυνατές μορφοποιήσεις και λαμβάνεται το τελικό αποτέλεσμα. Το αποτέλεσμα της επεξεργασίας μπορεί να είναι είτε μια νέα εικόνα είτε ένα χαρακτηριστικό γνώρισμα της εικόνας και η όλη διαδικασία εξαρτάται από τη γνωστική βάση του συστήματος [GW18].

Η αρχική εικόνα μπορεί να προέρχεται από διαφορετικές πηγές ή να βρίσκεται σε μη ψηφιακή μορφή. Η μορφή της αρχικής εικόνας εξαρτάται άμεσα από την πηγή της, δηλαδή το είδος της ακτινοβολίας ή της μορφής ενέργειας από την οποία προέρχεται. Για αυτόν το λόγο το στάδιο της απόκτησης της εικόνας μπορεί να περιλαμβάνει ένα στάδιο προεξεργασίας, όπως ψηφιοποίηση ή αλλαγή κλίμακας, με σκοπό η αρχική εικόνα να εισαχθεί στην διαδικασία στην επιθυμητή μορφή.

Τα είδη της επεξεργασίας που μπορεί να υποστεί η εικόνα ποικίλουν και εξαρτώνται από το σκοπό της επεξεργασίας και τις δυνατότητες του χρήστη. Η βελτίωση της εικόνας και η χρήση φίλτρων στοχεύει στην αλλαγή κάποιων χαρακτηριστικών για την καλύτερη χρήση σε μια συγκεκριμένη εφαρμογή. Το είδος της επεξεργασίας προσανατολίζεται και εξειδικεύεται για τη συγκεκριμένη αυτή εφαρμογή. Ως αποτέλεσμα, μια μέθοδος που είναι βέλτιστη για τις ακτίνες X δεν είναι απαραίτητα κατάλληλη για εικόνες προερχόμενες από ραδιοκύματα ή για δορυφορικές εικόνες.

Η αποκατάσταση εικόνας, όπως και η βελτίωση, αποσκοπεί στην αύξηση της ποιότητας της εμφάνισης της, παρουσιάζοντας όμως μια σημαντική διαφορά. Η αποκατάσταση στηρίζεται σε



Εικόνα 2.5. Θεμελιώδεις στάδια στην επεξεργασία εικόνων [GW18]

αντικειμενικά κριτήρια και τεχνικές που σχετίζονται με την ποιότητα των εικόνων, υποβάθμιση ή αναβάθμιση, μέσα από μαθηματικά μοντέλα και πιθανότητες. Αντίθετα, η βελτίωση λαμβάνει χώρα βασιζόμενη σε υποκειμενικά κριτήρια αναφορικά με τι θεωρείται όμορφο από τον χρήστη ή από τον εκάστοτε θεατή.

Τα κυματίδια και άλλοι μετασχηματισμοί εικόνας αποτελούν θεμέλιο στη δημιουργία και στη διάκριση διαφορετικών βαθμών ευκρίνειας και ανάλυσης, ενώ η συμπίεση σχετίζεται με τεχνικές μείωσης αποθηκευτικού χώρου ή με το απαραίτητο εύρος ζώνης για την μετάδοσή της. Η μορφολογική επεξεργασία αναφέρεται σε εργαλεία για την εξαγωγή στοιχείων της εικόνας που θα είναι χρήσιμα στην περιγραφή και την αντιπροσώπευση του σχήματος ή του περιεχομένου της. Η κατάτμηση διαιρεί την εικόνα σε επιμέρους κομμάτια ή αντικείμενα και η εξαγωγή χαρακτηριστικών, που σχεδόν πάντα ακολουθεί το στάδιο της κατάτμησης, αναπαριστά συγκεκριμένα εικονοστοιχεία που βρίσκονται είτε στα όρια είτε στο εσωτερικό μιας συγκεκριμένης περιοχής. Τα όρια μιας περιοχής μπορεί να είναι τα συνοριακά εικονοστοιχεία που χωρίζουν ένα αντικείμενο της εικόνας από ένα άλλο. Η ταξινόμηση πρότυπων εικόνων ορίζει τίτλους και ταμπέλες στα αντικείμενα μιας εικόνας με βάση την εξαγωγή χαρακτηριστικών που έχει προηγηθεί.

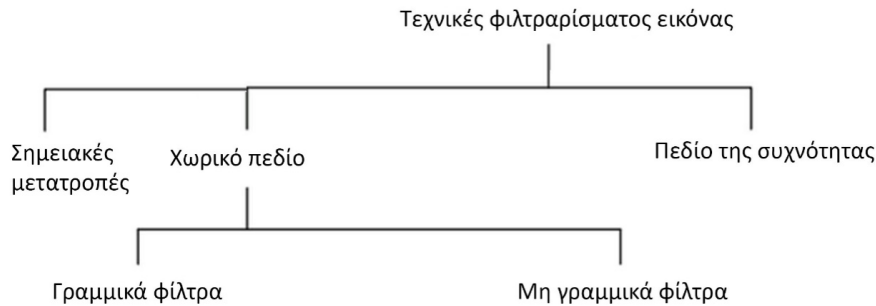
Το συνδετικό στοιχείο των διαφορετικών τεχνικών επεξεργασίας είναι η γνωστική βάση, που έχει τη μορφή μιας βάσης δεδομένων. Δηλαδή ενός συστήματος επεξεργασίας εικόνας στο οποίο βρίσκονται κωδικοποιημένες όλες οι πληροφορίες που απαιτούνται για την πραγματοποίηση των διεργασιών. Μέσα στη γνωστική βάση υπάρχουν όλες οι διαθέσιμες πληροφορίες από απλά δεδομένα, έως και πιο περίπλοκα. Ένα παράδειγμα απλού δεδομένου είναι οι αναλυτικές περιγραφές κατηγοριών για να ταξινομούνται τα αντικείμενα της εικόνας γρήγορα, ενώ ένα παράδειγμα σύνθετου είναι οι αλληλοσυνδεδεμένες λίστες ενδεχόμενων ελαττωμάτων σε διαφορετικά υλικά, ώστε στις εικόνες αυτοψίας υλικών να προσδιορίζονται αυτόματα τα ελαττώματα που υπάρχουν. Μια σημαντική λειτουργία της γνωστικής βάσης είναι ότι καθιστά δυνατή την αλληλεπίδραση των διάφορων διεργασιών μεταξύ τους.

Η επεξεργασία μιας εικόνας δεν είναι απαραίτητο να χρησιμοποιεί όλες τις διαθέσιμες διεργασίες, όμως σε γενικές γραμμές όσο αυξάνεται η περιπλοκότητα μιας επεξεργασίας τόσο περισσότερες διεργασίες απαιτούνται. Η παρούσα διπλωματική εργασία επικεντρώνεται στη μέθοδο της βελτίωσης μιας εικόνας με τη χρήση φίλτρων. Οι μορφοποιήσεις που μπορούν να επιτευχθούν με τη βοήθεια των φίλτρων είναι ποικίλες, οι κυριότερες εκ των οποίων είναι οι αλλαγές στην

φωτεινότητα, στην αντίθεση, στην οξύτητα και στον τονισμό.

2.3 Χωρικό φιλτράρισμα

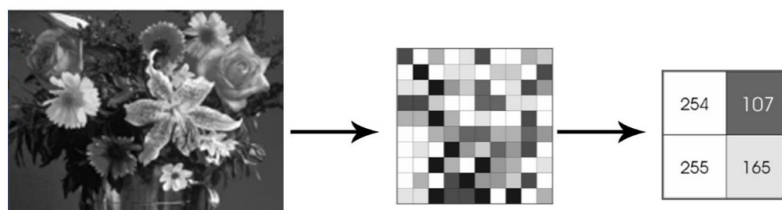
Οι τρεις βασικές κατηγορίες της βελτίωσης της εικόνας είναι οι σημειακές μετατροπές, το χωρικό φιλτράρισμα και το φιλτράρισμα στο πεδίο της συχνότητας (2.6).



Εικόνα 2.6. Τεχνικές φιλτραρίσματος εικόνας

Στις σημειακές μετατροπές μεμονωμένα εικονοστοιχεία μεταβάλλονται με σκοπό την αλλαγή της αντίθεσης και της κατωφλίωσης (thresholding). Αντίθετα, στο χωρικό φιλτράρισμα οι διεργασίες εκτελούνται σε ομάδες γειτονικών εικονοστοιχείων. Παραδείγματα χωρικού φιλτραρίσματος είναι η εξομάλυνση (smoothing) και η οξύτητα (sharpening) της εικόνας. Ανάλογα με τη μαθηματική βάση της διεργασίας, τα είδη των φίλτρων χωρίζονται σε γραμμικά και μη γραμμικά. Τα μη γραμμικά φίλτρα χρησιμοποιούνται για να εντοπίζουν τις ακμές της εικόνας, ενώ με τη χρήση γραμμικών φίλτρων οι λεπτομέρειες και οι ακμές της εικόνας τείνουν να θολώνουν. Παραδείγματα γραμμικών φίλτρων είναι τα Gaussian, Laplacian και NeighborhoodAverage (Mean), ενώ μη γραμμικών είναι το φίλτρο Median.

Το χωρικό φιλτράρισμά προϋποθέτει τον ορισμό του χωρικού πεδίου, που είναι το επίπεδο το οποίο περιλαμβάνει τα εικονοστοιχεία μιας ψηφιακής εικόνας. Κάθε ψηφιακή εικόνα μπορεί να αναπαρασταθεί με ένα δισδιάστατο πίνακα, όπου κάθε στοιχείο του αντιπροσωπεύει την ένταση ενός εικονοστοιχείου, δηλαδή την τιμή της απόχρωσης του. Η εικόνα 2.7 παρουσιάζει την αποδόμηση μιας μονόχρωμης εικόνας με γκρι αποχρώσεις σε ένα δισδιάστατο πίνακα.



Εικόνα 2.7. Μετατροπή ψηφιακής εικόνας σε πίνακα

Οι διεργασίες που λαμβάνουν χώρα στο χωρικό πεδίο περιγράφονται μαθηματικά από την παρακάτω εξίσωση:

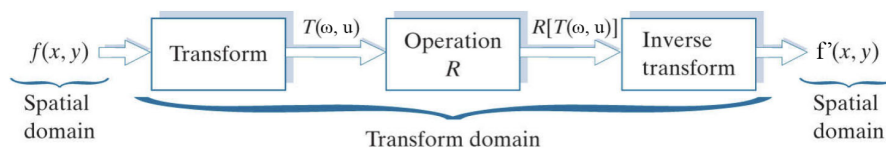
$$g(x,y) = T[f(x,y)]$$

όπου, η $f(x,y)$ είναι η αρχική εικόνα και η $g(x,y)$ η τελική. T είναι ο τελεστής της συνάρτησης f που ορίζεται σε ένα διπλανό σημείο (x,y) .

2.4 Φιλτράρισμα στο πεδίο της συχνότητας

Το χωρικό πεδίο εκτελεί τις διεργασίες στα εικονοστοιχεία της αρχικής εικόνας, όμως σε κάποιες περιπτώσεις η διεργασία δεν μπορεί να πραγματοποιηθεί σε αυτό το πεδίο και είναι απαραίτητος ο μετασχηματισμός του στο πεδίο της συχνότητας. Το πεδίο της συχνότητας αντιπροσωπεύει το ρυθμό αλλαγής των χωρικών εικονοστοιχείων και συνεπώς χρησιμοποιείται στα προβλήματα που σχετίζονται με αυτή τη μεταβλητή. Άλλο ένα σημαντικό πλεονεκτήματα του πεδίου της συχνότητας είναι η μικρότερη υπολογιστική πολυπλοκότητα που ενέχει.

Ως συχνότητα ορίζεται ο αριθμός των φορών που μια περιοδική συνάρτηση επαναλαμβάνεται με την ίδια αλληλουχία τιμών σε μια μονάδα χρόνου μιας ανεξάρτητης μεταβλητής. Όπως παρουσιάζεται στην εικόνα 2.8, η αρχική εικόνα από το πεδίο του χώρου μετασχηματίζεται και η διεργασία λαμβάνει χώρα στο μετασχηματισμένο πεδίο. Μετά την εκτέλεση της επιθυμητής διεργασίας εκτελείται ο αντίστροφος μετασχηματισμός και η τελική εικόνα γίνεται ορατή στο χωρικό πεδίο. Ένα σημαντικό χαρακτηριστικό των αντίστροφων μετασχηματισμών είναι ότι δεν χάνονται πληροφορίες, αλλά τα δεδομένα παραμένουν αυτούσια και αναλλοίωτα.



Εικόνα 2.8. Διαδικασία φιλτραρίσματος με μετασχηματισμό πεδίου [GW18]

Για να γίνει κατανοητή η ανάγκη του μετασχηματισμού θα αναλυθεί ένα παράδειγμα [GW18]. Στην εικόνα 2.9α μια περιοδική ημιτονοειδής συνάρτηση παρεμβάλλεται στην κυρίως εικόνα, λόγω ενός σφάλματος στο σύστημα απεικόνισης. Στο χωρικό πεδίο η παρεμβολή εμφανίζεται ως κύμα μιας ορισμένης έντασης. Στο πεδίο της συχνότητας γίνεται ορατή ως μια λάμψη ορισμένης έντασης, της οποίας η θέση μπορεί να καθοριστεί από τη συχνότητα της ημιτονοειδούς παρεμβολής (εικόνα 2.9b). Συνήθως, οι λάμπες είναι εύκολα παρατηρήσιμες σε μια εικόνα ως μια τάξη μεγέθους ενός μετασχηματισμού Fourier $|T(\omega, u)|$.

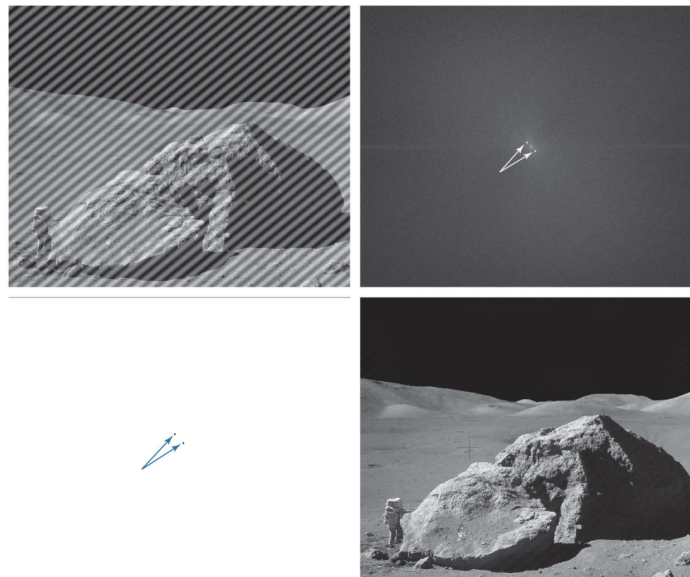
Αν υποθέσουμε ότι η αρχική εικόνα είναι η συνάρτηση $f(x, y)$, τότε ο μετασχηματισμός Fourier $|T(\omega, u)|$ απεικονίζεται στην εικόνα 2.9b. Η εικόνα 2.9c αντιπροσωπεύει την επιθυμητή διεργασία, δηλαδή την αφαίρεση της παρεμβολής. Η αφαίρεση γίνεται με τη χρήση μιας μάσκας ή αλλιώς ενός φίλτρου στις συχνότητες που παρατηρήθηκε η παρεμβολή από την εικόνα 2.9b. Δηλαδή, το φίλτρο είναι η συσκευή ή το υλικό που καταστέλλει ή μικραίνει τα κύματα ή τις ταλαντώσεις μιας συγκεκριμένης συχνότητας. Μαθηματικά η διεργασία εκφράζεται ως το γινόμενο του φίλτρου επί το μετασχηματισμό. Το τελικό αποτέλεσμα (εικόνα 2.9d) λαμβάνεται μετά τον αντίστροφο μετασχηματισμό, όπου η παρεμβολή δεν είναι πλέον ορατή και όλα τα στοιχεία της εικόνας είναι ευκρινή.

Ένα σήμα μπορεί να μετατραπεί από το πεδίου του χώρου στο πεδίο των συχνοτήτων μέσω διαφορετικών μαθηματικών μετασχηματισμών [Μπε15]. Παραδείγματα τέτοιων μετασχηματισμών είναι ο Διακριτός Μετασχηματισμός Fourier ή DFT (Discrete Fourier Transform), ο Διακριτός Μετασχηματισμός Συνημίτονου ή DCT (Discrete Cosine Transform), ο Μετασχηματισμός Wavelet. Από τους παραπάνω μετασχηματισμούς θα εξηγηθεί ο μετασχηματισμός Fourier που χρησιμοποιήθηκε για την εκπόνηση της συγκεκριμένης εργασίας.

2.5 Μετασχηματισμός Fourier

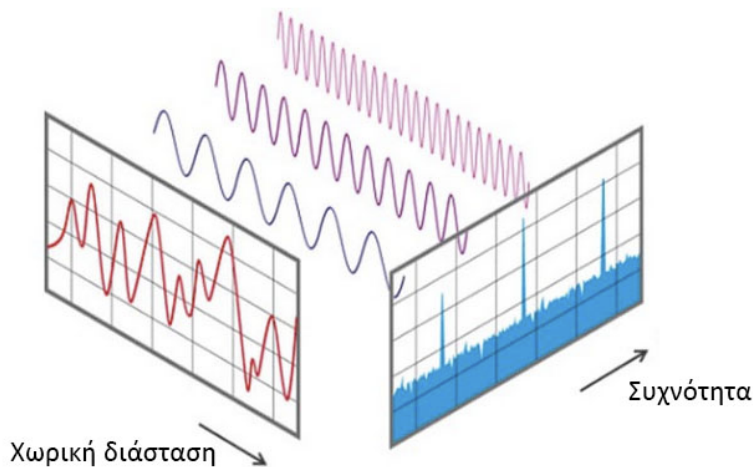
Ο μετασχηματισμός Fourier είναι μια μαθηματική διεργασία με την οποία μια συνάρτηση εκφρασμένη σε ένα πεδίο, π.χ. στο χώρο, μπορεί να εκφραστεί σε ένα άλλο πεδίο, π.χ. στη συ-

a b
c d



Εικόνα 2.9. Αφαίρεση θορύβου στο πεδίο του μετασχηματισμού [GW18]

χνότητα, το οποίο δρα ως εξαρτώμενη μεταβλητή. Η γραφική απεικόνιση του μετασχηματισμού παρατίθεται στην εικόνα 2.10.



Εικόνα 2.10. Μετασχηματισμός Fourier [Nau21]

Οι μετασχηματισμοί Fourier προέρχονται από τις σειρές Fourier, οι οποίες χρησιμοποιούνται ευρέως σαν εργαλείο στις θετικές επιστήμες. Διατυπώνουν μαθηματικά ότι κάθε περιοδική και ολοκληρώσιμη συνάρτηση μπορεί να γραφτεί ως άθροισμα ημιτόνων και συνημίτονων διαφορετικών συχνοτήτων, πολλαπλασιασμένα με διαφορετικούς συντελεστές. Η σειρά Fourier μιας συνεχούς εξίσωσης $f(x)$, με περίοδο T έχει τη γενική μορφή [Smi07]:

$$f(\omega_n) = \frac{1}{T} \int_0^T f(x) e^{-j\omega_n x} dx$$

όπου $n = 0, \pm 1, \pm 2$.

Η νέα μεταβλητή, ω , που εμφανίζεται αντιπροσωπεύει τη συχνότητα και οι μονάδες μέτρησής της εξαρτώνται από τις μονάδες μέτρησης του χώρου, x . Ως αποτέλεσμα η ω_x είναι η ν-ιοστή αρμονική συχνότητα. Με βάση τη σχέση του Euler ο όρος $e^{-j\omega_x x}$ μπορεί να μετατραπεί σε όρους ημιτόνων και συνημίτωνων.

Συγκεκριμένα, η εξίσωση του Euler για μια γωνία θ διατυπώνεται ως εξής:

$$e^{j\theta} = \eta\mu(\theta) + j\sigma\upsilon\nu(\theta)$$

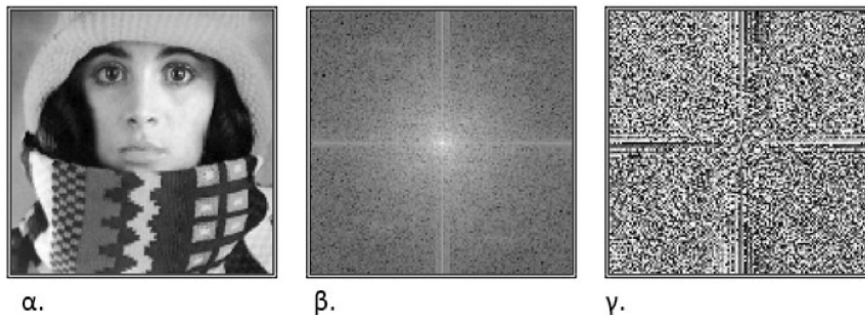
Ο μετασχηματισμός Fourier, $J\{f(x)\}$, στηρίζεται στην ίδια βάση με τις σειρές Fourier, αλλά εφαρμόζεται σε μη περιοδικές συναρτήσεις. Μια μη περιοδική συνάρτηση, το εμβαδόν της οποίας ορίζεται από μια πεπερασμένη καμπύλη, μπορεί να εκφραστεί ως το ολοκλήρωμα ημιτόνων και συνημίτωνων πολλαπλασιασμένων με μια συνάρτηση σταθμίσεως. Μαθηματικά, αυτή η έκφραση μπορεί να περιγραφεί ως εξής:

$$T(\omega) = J\{f(x)\} = \int_{-\infty}^{\infty} f(x)e^{-j\omega x} dx$$

Αντίστοιχα, όταν οι μεταβολές λαμβάνουν χώρα στο διδιάστατο επίπεδο παίρνει τη μορφή:

$$T(\omega, u) = J\{f(x, y)\} = \int_{-\infty}^{\infty} f(x, y)e^{-j(\omega x + uy)} dx dy$$

Η μεταβλητές x, y περιγράφουν τις καρτεσιανές συντεταγμένες στο χωρικό πεδίο, ενώ οι μεταβλητές ω, u τη συχνότητα και τη φάση του κύματος στο πεδίο της συχνότητας, αντίστοιχα. Το τελικό αριθμητικό αποτέλεσμα του μετασχηματισμού είναι ένας μιγαδικός αριθμός που περιλαμβάνει ένα πραγματικό μέρος, το πλάτος, και ένα φανταστικό, τη φάση. Στην εικόνα 2.11α παρουσιάζεται μια εικόνα στο πεδίο του χώρου και στα συστατικά της στοιχεία στο πεδίο των συχνοτήτων, πλάτος (εικόνα 2.11β) και φάση (εικόνα 2.11γ) σε όλες τις συχνότητες.



Εικόνα 2.11. Εικόνα στο χωρικό πεδίο (α), ως πλάτος (β) και ως φάση (γ) σε όλες τις συχνότητες [YV21]

Είναι προφανές ότι τόσο η συχνότητα όσο και φάση είναι απαραίτητα για την πλήρη ανακατασκευή του χωρικού πεδίου κατά τον αντίστροφο μετασχηματισμό [YV21]. Ο μετασχηματισμός Fourier είναι πλήρως αναστρέψιμος και ο αντίστροφος μετασχηματισμός Fourier δίνεται από την εξίσωση:

$$f(x, y) = \int_{-\infty}^{\infty} T(\omega, u)e^{j(\omega x + uy)} d\omega du$$

Καθώς ο μετασχηματισμός Fourier που αναλύθηκε αναφέρεται σε συνεχείς συναρτήσεις θα πρέπει να γραφτεί σε μια μορφή που να εφαρμόζεται σε σήματα που παίρνουν διακριτές τιμές,

όπως είναι οι εικόνες. Για μια εικόνα διαστάσεων $N \times M$ εικονοστοιχεία, η μορφή που χρησιμοποιείται ονομάζεται Διακριτός Μετασχηματισμός Fourier και μαθηματικά εκφράζεται ως:

$$T(\omega, u) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-j(\frac{\omega x}{N} + \frac{u y}{M})}$$

Στην περίπτωση που οι αριθμοί N και M είναι μεγάλοι, η διαδικασία των μετασχηματισμών Fourier κατά τη χρήση τους στην επεξεργασία εικόνων αποδεικνύεται ιδιαίτερα χρονοβόρα και πολύπλοκη, διότι απαιτούνται οι υπολογισμοί αθροισμάτων και γινομένων της τάξης του $(NM)^2$ [Fer19]. Για το λόγο αυτό, αντί του Διακριτού Μετασχηματισμού Fourier χρησιμοποιείται ο Γρήγορος Μετασχηματισμός Fourier (FFT), που πραγματοποιεί τους υπολογισμούς μέσω λογαρίθμων, και συγκεκριμένα μετατρέπει το γινόμενο $(NM)^2$ σε $(MN \log_2 MN)$. Ως αποτέλεσμα ο υπολογιστικός χρόνος μειώνεται και είναι δυνατή η επεξεργασία εικόνων μεγάλου μεγέθους.

2.6 Εφαρμογή φίλτρων

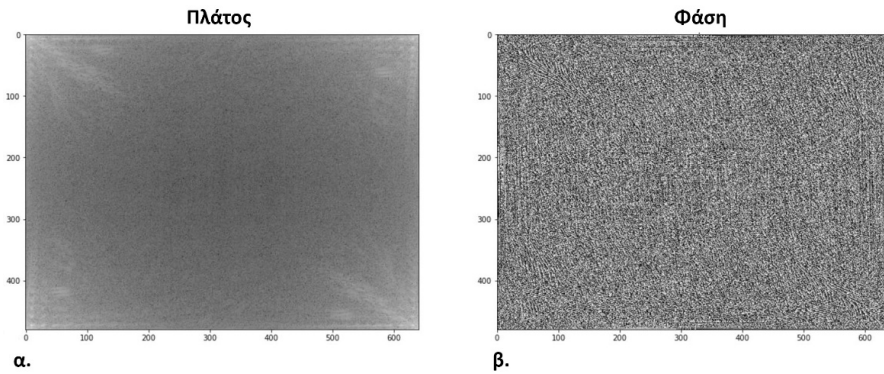
Συνολικά, η χρήση φίλτρων με σκοπό τη βελτίωση μιας εικόνας μπορεί να συνοψιστεί σε πέντε βασικά στάδια. Για την εφαρμογή του κατάλληλου φίλτρου στην επεξεργασία μιας εικόνας Greyscale ακολουθούνται τα εξής βήματα [Nau21]:

- Η εφαρμογή του μετασχηματισμού για τη μετατροπή του χωρικού πεδίου σε πεδίο της συχνότητας.
- Η κατανόηση της εικόνας και η μετακίνηση στοιχείων με βάση τη συχνότητά τους.
- Η εφαρμογή φίλτρων με την κατάλληλη συχνότητα.
- Η επαναφορά των στοιχείων στην αρχική τους θέση.
- Η εφαρμογή του αντίστροφου μετασχηματισμού.

Όπως ήδη αναλύθηκε, το πρώτο βήμα πραγματοποιείται με τη βοήθεια του Γρήγορου Μετασχηματισμού Fourier και ο μιγαδικός αριθμός που προκύπτει αναλύεται στις μεταβλητές του πλάτους και της φάσης ως συνάρτηση της συχνότητας. Η εικόνα στο σύνολό της εξαρτάται από τις συχνότητες των ημιτονοειδών της στοιχείων και συνεπώς μπορούμε να συσχετίσουμε τις διαφορετικές συχνότητες με τα επιμέρους στοιχεία της εικόνας. Σε αυτό το σημείο είναι ιδιαίτερα σημαντική η κατανόηση των μεταβλητών του πλάτους, της φάσης και της γραφικής τους απεικόνισης.

Στην εικόνα 2.12α το λευκό χρώμα αναπαριστά υψηλή συγκέντρωση συχνοτήτων και οι γωνίες της εικόνας αντιπροσωπεύουν χαμηλές συχνότητες [Che20]. Συνεπώς, οι λευκές περιοχές στις γωνίες υποδεικνύουν υψηλή συγκέντρωση χαμηλών συχνοτήτων, που είναι λογικό αποτέλεσμα στην ανάλυση μιας εικόνας. Γενικά, οι υψηλές συχνότητες αντιστοιχούν στα άκρα και στα περιγράμματα των στοιχείων μιας εικόνας, ενώ ημίτονα χαμηλών συχνοτήτων σε ομαλές περιοχές και εντάσεις που μεταβάλλονται αργά, όπως για παράδειγμα οι μονόχρωμοι τοίχοι ενός δωματίου σε μια εικόνα. Με μια πρώτη ματιά, από την εικόνα 2.12β, δεν μπορούμε να αντλήσουμε κάποια σημαντική πληροφορία, όμως είναι γνωστό ότι η μεταβλητή της φάσης είναι υπεύθυνη για τη διατήρηση των σχημάτων των αντικειμένων που περιέχει η εικόνα.

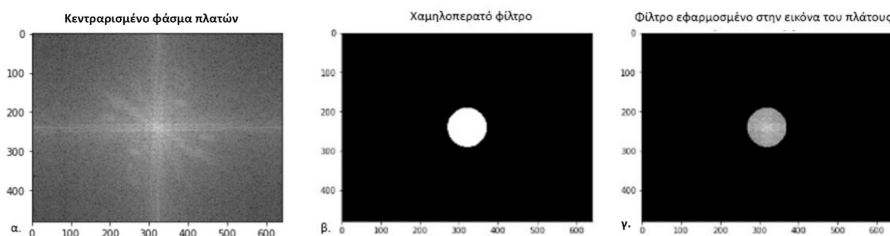
Το δεύτερο βήμα της διαδικασίας αφορά την μετακίνηση στοιχείων της εικόνας με συγκεκριμένες συχνότητες. Παραδείγματος χάρη, η μετακίνηση στοιχείων στο κέντρο με σκοπό την καλύτερη ορατότητά τους από το ανθρώπινο μάτι και συνεπώς την διευκόλυνση της επεξεργασίας. Η δυνατότητα αυτή βασίζεται στις ιδιότητες του Γρήγορου Μετασχηματισμού Fourier που επιτρέπει την περιστροφή των στοιχείων χωρίς την αλλοίωση της πληροφορίας που περιέχουν. Το



Εικόνα 2.12. Πλάτος (α) και φάση (β) στις συχνότητες της εικόνας [Che20]

αποτέλεσμα είναι η μετακίνηση των χαμηλών συχνοτήτων από τα άκρα στο κέντρο της εικόνας και παρουσιάζεται γραφικά στην εικόνα 2.13α.

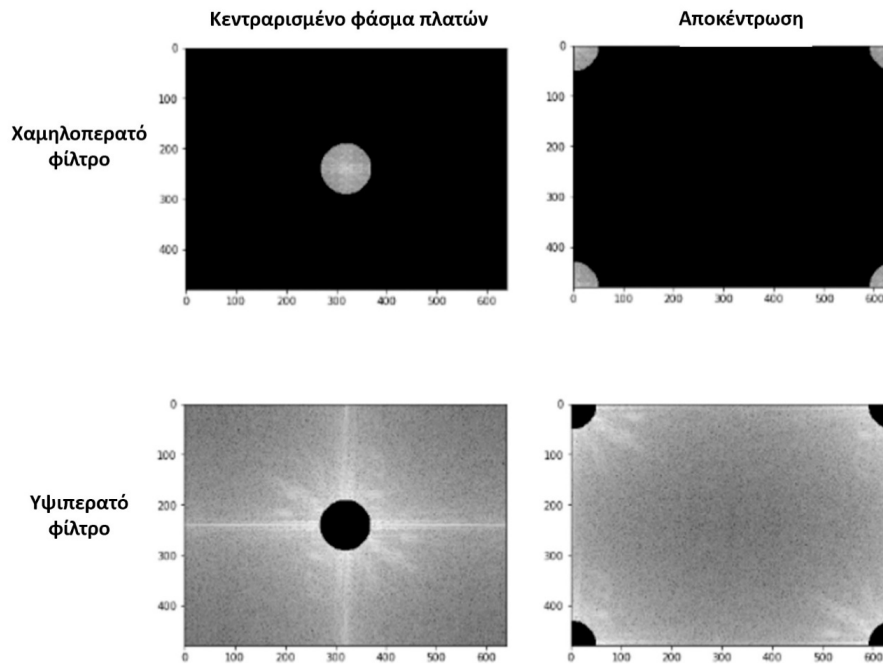
Το τρίτο βήμα είναι η εφαρμογή του κατάλληλου φίλτρου για την επεξεργασία, ανάλογα με την ζητούμενη μορφοποίηση. Δύο από τα απλούστερα φίλτρα είναι το χαμηλοπερατό και το υπερπερατό φίλτρο. Ο τρόπος λειτουργίας τους είναι η δημιουργία ενός κύκλου μιας συγκεκριμένης διαμέτρου, μέσα στον οποίο θα επιτρέπεται η διέλευση ορισμένων συχνοτήτων (εικόνα 2.13β). Τα χαμηλοπερατά φίλτρα επιτρέπουν μόνο στις χαμηλές συχνότητες να περάσουν, ενώ οι υψηλές αποκλείονται. Ως αποτέλεσμα, με την εφαρμογή ενός χαμηλοπερατού φίλτρου οι θόρυβοι και οι παρεμβολές υψηλών συχνοτήτων αφαιρούνται και η εικόνα τείνει να θολώνει. Στην εικόνα 2.13γ παρουσιάζεται η εφαρμογή ενός χαμηλοπερατού φίλτρου στην εικόνα 2.13α. Αντίστοιχα, τα υπερπερατά φίλτρα επιτρέπουν στις υψηλές συχνότητες να εμφανίζονται, δηλαδή στις περιοχές της εικόνας με έντονες χρωματικές αλλαγές. Σκοπός είναι να ενισχυθεί η οξύτητα και να ενταθούν οι διαφορές των χρωμάτων.



Εικόνα 2.13. Εφαρμογή χαμηλοπερατού φίλτρου [Che20]

Υπάρχουν διαφορετικά είδη χαμηλοπερατών και υπερπερατών φίλτρων ανάλογα με τον τρόπο που επιλέγονται οι επιτρεπτές συχνότητες. Παραδείγματος χάρη, στο Ιδανικό φίλτρο (Ideal filter) οι συχνότητες επιλέγονται με βάση μια οριακή τιμή μέσω μιας διακλαδισμένης εξίσωσης, ενώ το φίλτρο Gauss εκφράζεται από μια εκθετική συνάρτηση. Ως αποτέλεσμα στο φίλτρο Gauss οι αλλαγές θα είναι ομαλότερες και θα μειώνονται οι παραμορφώσεις [Che20].

Στο τέταρτο και πέμπτο βήμα πραγματοποιούνται οι αντίστροφες διαδικασίες του δεύτερου και του πρώτου βήματος, αντίστοιχα. Η μεταφορά στον κέντρο της εικόνας των χαμηλών συχνοτήτων ονομάζεται αποκέντρωση. Δηλαδή, τα στοιχεία που είχαν μεταφερθεί στον κέντρο επιστρέφουν στα άκρα της εικόνας. Στην εικόνα 2.14 παρουσιάζονται συγκριτικά οι περιπτώσεις χαμηλοπερατών και υπερπερατών φίλτρων και τα αποτελέσματα μετά τη διαδικασία της αποκέντρωσης. Τέλος, πραγματοποιείται ο αντίστροφος μετασχηματισμός και λαμβάνεται η εικόνα μετά την επεξεργασίας της.



Εικόνα 2.14. Σύγκριση χαμηλοπερατού και υψιπερατού φίλτρου [Che20]

2.7 Ομομορφικό φιλτράρισμα

Μια απλή απόδειξη ότι μια εικόνα χρειάζεται επεξεργασία είναι η μη ομοιόμορφη φωτεινότητα. Δηλαδή, η συνύπαρξη περιοχών με χαμηλή φωτεινότητα και σκούρες αποχρώσεις μαζί με έντονα ανοιχτά χρώματα μειώνουν την ποιότητα της εικόνας και καθιστούν δύσκολη την κατανόηση του περιεχομένου της [Fer19].

Μια τεχνική που στοχεύει στη διόρθωση αυτού του φαινομένου είναι το ομομορφικό φιλτράρισμα, που πετυχαίνει την αφαίρεση του πολλαπλασιαζόμενου θορύβου με συγκεκριμένα χαρακτηριστικά. Η διεργασία συμβαίνει στο πεδίο της συχνότητας με τη χρήση υψιπερατών φίλτρων για τη μείωση των στοιχείων με χαμηλές συχνότητες. Παραδείγματος χάρη, στις εικόνες ακτινών X μειώνεται η επίδραση των γκρι αποχρώσεων που υπάρχουν στις μαύρες περιοχές.

Η βάση του ομομορφικού φιλτραρίσματος είναι το μοντέλο φωτεινότητας – ανακλαστικότητας που μαθηματικά περιγράφεται με την εξίσωση $f(x, y) = i(x, y)r(x, y)$ [Cha14] [Sal12]. Η εικόνα αναλύεται σε δύο συνιστώσες, την φωτεινότητα και την ανακλαστικότητα. Οι δύο όροι συνδέονται μεταξύ τους μέσω ενός γινομένου, που όμως, σε αντίθεση με το άθροισμα, όταν υποστεί Μετασχηματισμού Fourier λειτουργεί ως ένας ενιαίος όρος. Γι' αυτόν το λόγο το πρώτο βήμα είναι ο διαχωρισμός των δύο μεταβλητών με τη βοήθεια λογαρίθμων, όπως παρουσιάζεται παρακάτω:

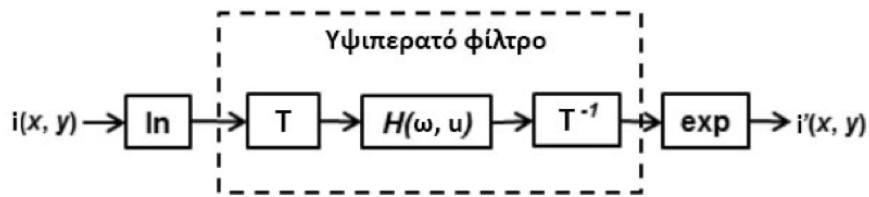
$$f(x, y) = i(x, y)r(x, y) \Rightarrow$$

$$\ln(f(x, y)) = \ln(i(x, y)r(x, y)) \Rightarrow$$

$$\ln(f(x, y)) = \ln(i(x, y)) + \ln(r(x, y))$$

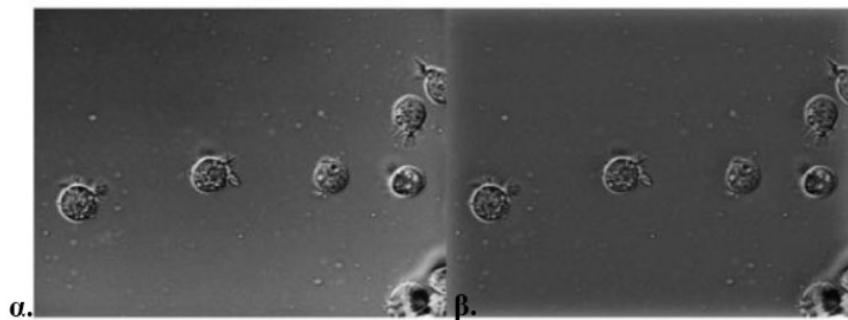
Ο όρος της φωτεινότητας είναι υπεύθυνος για τις αργές μεταβολές στα εικονοστοιχεία μιας εικόνας και συνεπώς σχετίζεται με τις χαμηλές συχνότητες. Αντίθετα, ο όρος της ανακλαστικότητας αναπαριστά τις λεπτομέρειες στα αντικείμενα μιας εικόνας, μεταβάλλεται ταχέως και επομένως συνδέεται με την τοπική αντίθεση και τις υψηλές συχνότητες. Παραδείγματος χάρη, για να μειωθεί η σκίαση σε μια περιοχή της εικόνας απαιτείται να αυξηθεί η φωτεινότητα και να τονιστούν οι

άκρες της περιοχής αυξάνοντας παράλληλα την ανακλαστικότητα. Ο στόχος του ομομορφικού φιλτραρίσματος είναι να μειωθεί η φωτεινότητα, μειώνοντας τα στοιχεία των χαμηλών συχνοτήτων της εικόνας και γι' αυτό το λόγο χρησιμοποιούνται υψιπερατά φίλτρα. Έπειτα, τα αποτελέσματα των δύο φίλτρων μεταφέρονται στην επιθυμητή κλίμακα ανάλογα με το ζητούμενο βαθμό επιρροής του κάθε φίλτρου και συνδυάζονται με την αντίστροφη μετατροπή των λογαριθμικών συναρτήσεων, δηλαδή τη χρήση εκθετικών όρων. Η συνολική διεργασία συνοψίζεται στην εικόνα 2.15, όπου φαίνονται οι διαδικασίες της λογαριθμικής μετατροπής (\ln), του μετασχηματισμού Fourier (T), της εφαρμογής του φίλτρου στο πεδίο των συχνοτήτων ($H(\omega, u)$), της επιστροφής στο χωρικό πεδίο μέσω του αντίστροφου μετασχηματισμού Fourier (T^{-1}) και της αφαίρεσης των λογαρίθμων μέσω εκθετικών συναρτήσεων (\exp).



Εικόνα 2.15. Συνολική διεργασία ομομορφικού φιλτραρίσματος [Edi13]

Το αναμενόμενο αποτέλεσμα είναι η βελτίωση της φωτεινότητας χωρίς την αλλοίωση των στοιχείων της εικόνας. Ένα τυπικό παράδειγμα παρατίθεται στην εικόνα 2.16, όπου συγκρίνεται μια εικόνα πριν την επεξεργασία με ομομορφικό φιλτράρισμα και μετά τη μορφοποίησή της. Είναι εμφανές ότι οι ανομοιομορφίες του φωτισμού έχουν εξαλειφθεί και τα χρώματα της εικόνας μεταβάλλονται ομαλότερα με την ταυτόχρονα διατήρηση των λεπτομερειών των επιμέρους αντικειμένων.



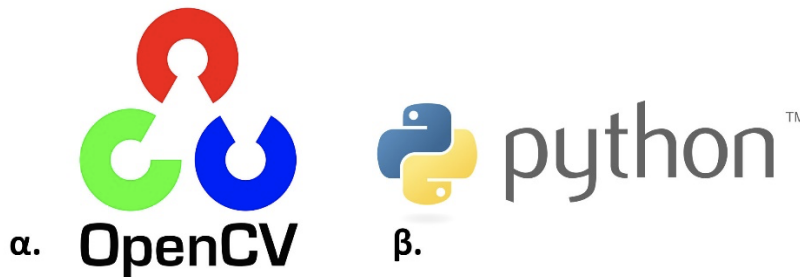
Εικόνα 2.16. Σύγκριση εικόνας πριν (α) και μετά (β) το ομομορφικό φιλτράρισμα [Edi13]

2.8 Υπολογιστικά εργαλεία Python και OpenCV

Η πρακτική υλοποίηση της επεξεργασίας εικόνων απαιτεί τη δημιουργία ενός υπολογιστικού κώδικα και τη χρήση μιας βιβλιοθήκης που παρέχει τη δυνατότητα ανάλυσης και κατανόησης των εικόνων προς επεξεργασία (εικόνα 2.17). Ο υπολογιστικός κώδικας περιγράφει τις απαιτούμενες εντολές με τη χρήση μαθηματικών μοντέλων και αλγορίθμων και η βιβλιοθήκη καθιστά δυνατή την αντίληψη και τη διάκριση των επιμέρους στοιχείων της εικόνας. Μια από τις βασικότερες γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν για αυτό το σκοπό είναι η γλώσσα Python. Η Python είναι μια γλώσσα υψηλού επιπέδου, όπως η Fortran, η Java και γενικής χρήσης, δηλαδή

έχει ευρεία εφαρμογή σε πολλά πεδία και είναι κατάλληλη για όλα τα επίπεδα χρηστών [Pyt21]. Η μετατροπή του κώδικα σε εντολές υπολογιστή πραγματοποιείται με τη βοήθεια διερμηνευτή και είναι δυνατή η χρήση της σε διαφορετικά λειτουργικά συστήματα (cross-platformsoftware) όπως Windows, Linux. Η προσέγγιση που στηρίζει είναι αντικειμενοστρεφής (object-oriented), δηλαδή το λογισμικό οργανώνεται γύρω από συγκεκριμένα αντικείμενα ή δεδομένα αντί για συναρτήσεις ή λογικά μοντέλα. Βασίζεται σε λογισμικό με άδεια χρήσης ανοικτού κώδικα, που σημαίνει ότι μπορεί να χρησιμοποιηθεί και να διανεμηθεί ελεύθερα και αναπτύσσεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Η εκμάθηση και η χρήση της διευκολύνεται από τις έτοιμες βιβλιοθήκες που παρέχονται με τη δυνατότητα επέκτασής τους σε γλώσσα C, C++ και ο ίδιος ο κώδικα δομείται σε αρθρώματα (modules) και πακέτα (packages) [Αγγ21]. Η ορθή και αποτελεσματική χρήση της Python ήταν βασικό στοιχείο για την πραγματοποίηση της παρούσας εργασίας.

Το δεύτερο υπολογιστικό εργαλείο που χρησιμοποιήθηκε ήταν η βιβλιοθήκη Open Source Computer Vision Library (OpenCV). Η OpenCV αποτελείται από πληθώρα υπολογιστικών αλγορίθμων και χρησιμοποιείται κατά κόρον στο πεδίο της τεχνητής νοημοσύνης και συγκεκριμένα της τεχνητής όρασης [Mag20]. Αποτελεί ένα από τα δημοφιλέστερα εργαλεία τόσο στην αναγνώριση κίνησης, προσώπων ή αντικειμένων σε μια εικόνα ή σε ένα βίντεο όσο και στην επεξεργασία εικόνων, διότι μπορεί να αναγνωρίζει και να χειρίζεται το περιεχόμενο των εικονοστοιχείων. Όπως και η Python μπορεί να χρησιμοποιηθεί σε πολλά λειτουργικά συστήματα και αποτελεί λογισμικό ανοικτού κώδικα. Οι βιβλιοθήκες της OpenCV είναι γραμμένες σε γλώσσα C++. Τα δύο υπολογιστικά εργαλεία μπορούν να συνεργαστούν με τη βοήθεια της Python και συνδυαστικά παράγουν την επιθυμητή μορφοποίηση στην εικόνα.



Εικόνα 2.17. Τα υπολογιστικά εργαλεία OpenCV (α), Python (β)

Κεφάλαιο 3

Υπολογιστικό μέρος

3.1 Δομή υπολογιστικού κώδικα

Μετά την ανάλυση των απαραίτητων βασικών θεωρητικών εννοιών κατασκευάζεται ο υπολογιστικός κώδικας που καθιστά δυνατή τη βελτίωση της εικόνας με τη μέθοδο του ομομορφικό φιλτραρίσματος. Ο κώδικας δομείται σε μαθηματικές συναρτήσεις και σχέσεις που αποτελούν τις εντολές στην υπολογιστική γλώσσα Python, οι οποίες είναι υπεύθυνες για την πραγματοποίηση των επιθυμητών αλλαγών. Για την πρακτική διευκόλυνση στον χειρισμό του κώδικα, οι εντολές αποθηκεύονται σε δύο χωριστά αρχεία. Το αρχείο `hy` το οποίο καλείται για την εκτέλεση του αλγορίθμου, περιέχει τα βήματα που ακολουθούνται για την εφαρμογή του φίλτρου με τη μορφή εντολών. Το αρχείο `Utilities` το οποίο συνδέεται με το αρχείο `hy`, περιγράφει το ακριβές περιεχόμενο του κώδικα, δηλαδή κάθε εντολής και συνάρτησης που περιέχεται στο αρχείο `hy`. Ο αλγόριθμος στηρίζεται στο σχεδιάγραμμα της εικόνας 2.15 και στις συμπληρωματικές λειτουργίες που πρέπει οριστούν για να ολοκληρωθεί η διαδικασία. Επιγραμματικά, τα βήματα που ακολουθούνται μπορούν να συνοψιστούν ως εξής:

- Εισαγωγή της επιθυμητής εικόνας που πρόκειται να μορφοποιηθεί.
- Ανάλυση και κανονικοποίηση (normalization) εικόνας.
- Λογαρίθμηση του πίνακα της εικόνας.
- Μετασχηματισμός Fourier για την μετάβαση από το χωρικό πεδίο στο πεδίο της συχνότητας.
- Μεταφορά των στοιχείων με μηδενικές συχνότητες στο κέντρο της εικόνας.
- Εφαρμογή φίλτρου.
- Δημιουργία παραθύρου με τις μπάρες εργασιών (trackbar) μέσω των οποίων ο χρήστης θα επεξεργάζεται την εικόνα οπτικά.
- Αντίστροφος Μετασχηματισμός Fourier.
- Επιστροφή στην αρχική κλίμακα τιμών.
- Επανάληψη κανονικοποίησης της εικόνας.
- Αποθήκευση και έξοδος.

Παρακάτω θα επεξηγηθεί τόσο ο λόγος που απαιτείται το κάθε βήμα όσο και το περιεχόμενό του αναλυτικά. Μαζί με την επεξήγηση κάθε εντολής παρατίθεται και ο τρόπος εισαγωγής της στον κώδικα.

3.2 Εισαγωγή βιβλιοθηκών και εικόνας

Το πρώτο σημαντικό βήμα στην υλοποίηση του κώδικα πριν τον ορισμό των απαραίτητων εντολών, είναι η εισαγωγή των βιβλιοθηκών που θα καταστήσουν τις εντολές κατανοητές από τον υπολογιστή. Το πρώτο άρθρωμα που εισάγεται είναι το SYS (SYstem-Specific parameters and functions), που επιτρέπει την πρόσβαση ορισμένων μεταβλητών και εξισώσεων στον διερμηνέα της Python. Στη συνέχεια απαιτείται η βιβλιοθήκη cv (OpenCV), η οποία ευθύνεται για την αναγνώριση ποικίλων στοιχείων της εικόνας, όπως αναλύθηκε στην ενότητα 2.8. Έπειτα, καλείται το πακέτο NumPy (Numerical Python), μια επιστημονική βιβλιοθήκη που παρέχει πολυδιάστατα και παράγωγα αντικείμενα όπως μάσκες και πίνακες. Η βιβλιοθήκη NumPy περιέχει, επίσης, ρουτίνες για τον γρήγορο υπολογισμό των σειρών των αντικειμένων που βασίζονται σε ανώτερα μαθηματικά, Διακριτούς Μετασχηματισμούς Fourier, βασική γραμμική άλγεβρα, στατιστική, χειρισμό σχημάτων, προσομοιώσεις κτλ. [Num22]. Άλλη μια απαραίτητη βιβλιοθήκη είναι η Python Imaging Library (PIL) από την οποία εξαρτάται άμεσα η δυνατότητα εισαγωγής των εικόνων. Το τελευταίο πακέτο που εισάγεται ονομάζεται EasyGUI και αναφέρεται στην ικανότητα προγραμματισμού με τη χρήση γραφικού περιβάλλοντος χρήστη (Graphical User Interface). Τα πακέτα GUI είναι η δίοδος επικοινωνίας του χρήστη με τον υπολογιστή μέσω συμβόλων, εικόνων, σημείων ή συσκευών και συνήθως διαμεσολαβούν αυτή την επικοινωνία με μια επιφάνεια αλληλεπίδρασης ορατή και οικεία για το χρήστη πάνω στην οποία κάθε απλή ενέργεια, όπως η επιλογή ενός σημείου στην εικόνα, μεταφράζονται σε μια εντολή. Τα προγράμματα GUI πρέπει να αποκρίνονται σε κάθε εντολή, συνεπώς λειτουργούν υπό τον έλεγχο του χρήστη και όχι του προγράμματος και συνήθως ανήκουν στην κατηγορία του προγραμματισμού χειρισμού γεγονότων (event-driven programming). Το πακέτο EasyGUI διαφέρει ως προς το σημείο αυτό, καθώς όλες οι αλληλεπιδράσεις συμβαίνουν με συναρτήσεις αντί για γεγονότα [Eas22]. Η εισαγωγή των πακέτων στον κώδικα πραγματοποιείται με την εντολή `import` και κατόπιν εισάγεται η εικόνα με την εντολή `easygui.fileopenbox`, όπως φαίνεται στην εικόνα 3.1.

```
import cv2 as cv
import numpy as np
from PIL import Image
import easygui

def load_image():
    image_path = easygui.fileopenbox()
    return image_path
```

Εικόνα 3.1. Εισαγωγή των απαραίτητων βιβλιοθηκών στον κώδικα

3.3 Ανάλυση και κανονικοποίηση εικόνας

Η ανάλυση της εικόνας λαμβάνει χώρα με την κατάταξη της σε έγχρωμη ή σε ασπρόμαυρη με γκρι αποχρώσεις. Αρχικά, η εικόνα ανοίγεται με την εντολή `Image.open` και δημιουργείται ένα PIL αντικείμενο, το οποίο μετατρέπεται σε RGB χώρο χρωμάτων με την εντολή `Image.convert`. Οι διαστάσεις τις εικόνας σε εικονοστοιχεία εισάγονται σε ένα βρόγχο και για κάθε εικονοστοιχείο λαμβάνεται η απόχρωσή με βάση τα τρία κύρια χρώματα κόκκινο (R), πράσινο (G) ή μπλε (B) με την εντολή `Image.getpixel`. Δηλαδή από τις συντεταγμένες του κάθε εικονοστοιχείου εξάγεται η

Πίνακας 3.1. Χαρακτηριστικοί τύποι κανονικοποίησης

Τύπος κανονικοποίησης	Μαθηματική εξίσωση
norm_INF	$\ dst\ = \max_i dst_i = alpha$
norm_L1	$\ dst\ = \sum_i dst_i = alpha$
norm_L2	$\ dst\ = \sqrt{\sum_i (dst_i)^2} = alpha$
norm_MINMAX	$[a, b] = [f_{newmin}, f_{newmax}]$

αριθμητική τιμή της απόχρωσής του και η εικόνα κατηγοριοποιείται σε έγχρωμη ή ασπρόμαυρη. Η επόμενη διαδικασία είναι η κανονικοποίηση (normalization) της εικόνας που αποσκοπεί στην μετατροπή της από το αρχικό εύρος εικονοστοιχείων σε ένα τελικό πιο οικείο για τις ανθρώπινες αισθήσεις. Με άλλα λόγια, η εικόνα θα καταστεί πιο «κανονική» για το ανθρώπινο μάτι. Η κανονικοποίηση μεταβάλλει την ένταση των τιμών των εικονοστοιχείων, αυξάνει δηλαδή την αντίθεση της συνολικής εικόνας. Αν υποθέσουμε μια εικόνα με διαστάσεις στο σύνολο των πραγματικών αριθμών και αρχικές τιμές της έντασης, f , στο διάστημα $[f_{min}, f_{max}]$, τότε οι τελικές τιμές, f_{new} , μετά την κανονικοποίηση είναι $[f_{newmin}, f_{newmax}]$. Μαθηματικά, η διεργασία της κανονικοποίησης περιγράφεται από την εξίσωση:

$$f_{new} = (f - f_{min}) \frac{f_{newmax} - f_{newmin}}{f_{max} - f_{min}} + f_{newmin}$$

Η παραπάνω εξίσωση εκφράζει την γραμμική κανονικοποίηση μιας ψηφιακής εικόνας με γκρι αποχρώσεις. Η υπολογιστική εντολή για την κανονικοποίηση έχει τη γενική μορφή [Doc22]:

```
normalize(source_array, destination_array, alpha, beta, normalization_type, depth_type)
```

Όπου,

source_array ή πίνακας εισόδου, είναι η αρχική εικόνα πριν την κανονικοποίηση,

destination_array (dst) ή πίνακας εξόδου, είναι η τελική εικόνα μετά την κανονικοποίηση,

alpha ή f_{newmin} , είναι η κατώτατη τιμή του εύρους τιμών της έντασης,

beta ή f_{newmax} , είναι η ανώτερη τιμή του εύρους τιμών της έντασης,

normalization_type ή τύπος κανονικοποίησης, είναι η μαθηματική μέθοδος που ακολουθείται για την πραγματοποίηση της κανονικοποίησης και

depth_type ή τύπος βάθους είναι η μορφή του πίνακα εξόδου.

Εάν είναι αρνητικός αριθμός τότε ο πίνακας εξόδου έχει την ίδια μορφή με τον πίνακα εισόδου, αντίθετα εάν δεν είναι αρνητικός έχει τον ίδιο αριθμό καναλιών (channels) με τον πίνακα εισόδου.

Ο τύπος κανονικοποίησης μπορεί να ποικίλει ανάλογα με τον τρόπο που υπολογίζονται οι νέες τιμές της έντασης και κάποιοι χαρακτηριστικοί τύποι παρουσιάζονται στον πίνακα 3.1 [Doc22].

Ο τύπος κανονικοποίησης που χρησιμοποιείται στην παρούσα εργασία είναι ο norm_MINMAX που αντιστοιχεί στην γραμμική κανονικοποίηση της εξίσωσης παραπάνω. Αξίζει να σημειωθεί ότι η διαδικασία της κανονικοποίησης πραγματοποιείται δύο φορές, μια φορά στην αρχή της διαδικασίας μαζί με την ανάλυση της εικόνας και επαναλαμβάνεται μετά την εφαρμογή του φίλτρου, και ειδικότερα μετά τη διαδικασία της επιστροφής στην αρχική κλίμακα τιμών. Ο λόγος επανάληψης της διαδικασίας συνίσταται στο ότι η εικόνα χρειάζεται να αποκτήσει τα εικονοστοιχεία στο αρχικό εύρος τιμών τους ώστε να μην παραμορφωθούν οι πληροφορίες που περιέχει.

Στην περίπτωση που η εικόνα είναι έγχρωμη, πριν την διαδικασία της κανονικοποίησης εισάγονται δύο εντολές που αφορούν την ανάλυση της εικόνας στις χρωματικές της συνιστώσες. Η εντολή cv.split επιτρέπει τον διαχωρισμό της εικόνας στους RGB χρωματικούς άξονες και η εντολή cv.cvtColor μετατρέπει ένα χώρο χρωμάτων σε έναν άλλο. Η τελευταία εντολή περιέχει πάνω από 150 χώρους χρωμάτων και η γενική της μορφή είναι η εξής [Doc22]:

cv2.cvtColor (scr, code)

Όπου,

scr ή πηγή, είναι η αρχική εικόνα και

code ή κώδικας, είναι οι χρωματικοί χώροι μεταξύ των οποίων συμβαίνει η μετατροπή.

Παραδείγματα χρωματικών χώρων είναι οι RGB, YUV, HLS, HSV, $YC_B C_R$. Η μετατροπή που λαμβάνει χώρα στην παρούσα διπλωματική εργασία είναι από τον χώρο RGB στον χώρο $YC_B C_R$.

Η $YC_B C_R$ είναι μια κατηγορία χρωματικών χώρων που αναλύουν την εικόνα σε τρεις συνιστώσες. Την συνιστώσα Y ή luma, που αντιπροσωπεύει την φωτεινότητα και τις συνιστώσες C_B, C_R ή chroma, που αντιστοιχούν στα μπλε και κόκκινα χρωματικά στοιχεία της εικόνας, αντίστοιχα. Κατά τον μετασχηματισμό της εικόνας από RGB σε $YC_B C_R$ τα χαρακτηριστικά των συντεταγμένων της εικόνας διατηρούνται, π.χ. αν ο αρχικός χώρος είναι απόλυτος (absolute) τότε και ο τελικός θα είναι επίσης απόλυτος. Το κύριο μειονέκτημα του RGB, λόγω του οποίου καθίσταται αναγκαία η μετατροπή, είναι η ανεπάρκεια της συγκεκριμένης απεικόνισης κατά την αποθήκευση και τη μετάδοση της εικόνας. Αντίθετα στον $YC_B C_R$ χώρο, η Y συνιστώσα μπορεί να αποθηκεύσει την πληροφορία με εξαιρετική ακρίβεια και να τη μεταδώσει σε μεγάλο εύρος ζώνης, ενώ οι συνιστώσες C_B, C_R μπορούν να υποστούν μείωση του εύρους ζώνης τους, να συμπιεστούν ή να μορφοποιηθούν ανεξάρτητα χωρίς να επηρεάζουν την φωτεινότητα της εικόνας. Μαθηματικά, η μετατροπή των τιμών των εικονοστοιχείων εκφράζεται ως εξής:

$$Y = 0.299R + 0.587G + 0.114B$$

$$C_r = 0.713(R - Y) + delta$$

$$C_b = 0.564(B - Y) + delta$$

Και αντίστροφα,

$$R = Y + 1.403(C_r - delta)$$

$$G = Y - 0.714(C_r - delta) + 0.344(C_b - delta)$$

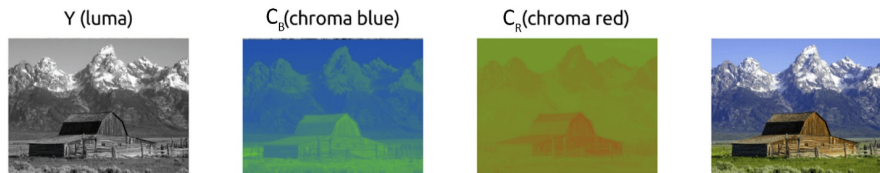
$$B = Y + 1.733(C_r - delta)$$

Όπου,

	128	8bit
delta =	32768	32bit
	0.5.	floating point

Στην εικόνα 3.2 παρουσιάζεται μια εικόνα αναλυμένη στις συνιστώσες του $YC_B C_R$ χρωματικού χώρου.

Ο λογάριθμός που χρησιμοποιήθηκε για την ανάλυση και την κανονικοποίηση της εικόνας παρατίθεται στην εικόνα 3.3.



Εικόνα 3.2. Ανάλυση εικόνας στις συνιστώσες του $YC_B C_R$ χρωματικού χώρου

```
def is_grey_scale(img_path):
    img = Image.open(img_path).convert('RGB')
    w, h = img.size
    for i in range(w):
        for j in range(h):
            r, g, b = img.getpixel((i,j))
            if r != g != b: //!<= means if not
                return False
    return True

def read_img(imgpath):
    Cr = 0
    Cb = 0
    if(is_grey_scale(imgpath)):
        image = cv.imread(imgpath, 0)
        image = cv.normalize(image, None, alpha=0, beta=1, norm_type=cv.NORM_MINMAX, dtype=cv.CV_32F)
        color = "gray"
    else:
        image = cv.imread(imgpath)
        image = cv.cvtColor(image, cv.COLOR_BGR2YCrCb)
        channels = cv.split(image)
        image = cv.normalize(channels[0], None, alpha=0, beta=1, norm_type=cv.NORM_MINMAX, dtype=cv.CV_32F)
        Cr = channels[1]
        Cb = channels[2]
        color = "rgb"
    return image, color, Cr, Cb

def normalize(exp_image, color, Cr, Cb):
    if (color == "gray"):
        return cv.normalize(exp_image, None, alpha=0, beta=255, norm_type=cv.NORM_MINMAX, dtype=cv.CV_8U)
    else:
        image = cv.normalize(exp_image, None, alpha=0, beta=255, norm_type=cv.NORM_MINMAX, dtype=cv.CV_8U)
        image = cv.merge((image, Cr, Cb))
        return cv.cvtColor(image, cv.COLOR_YCrCb2BGR)
```

Εικόνα 3.3. Ανάλυση και κανονικοποίηση εικόνας

3.4 Λογαρίθμηση και επιστροφή στην αρχική κλίμακα

Όπως έχει εξηγηθεί και στην ενότητα 2.7 η λογαρίθμηση της εξίσωσης της εικόνας είναι απαραίτητη για το διαχωρισμό των όρων της φωτεινότητας και της ανακλαστικότητας που συνδέονται με ένα γινόμενο. Η λογαρίθμηση λαμβάνει χώρα αμέσως μετά την ανάλυση της εικόνας και πριν το Μετασχηματισμό Fourier, ενώ μετά την εφαρμογή του φίλτρου απαιτείται η επιστροφή στην αρχική κλίμακα τιμών. Η τελευταία εκτελείται αμέσως μετά τον αντίστροφο Μετασχηματισμό Fourier. Η τοποθέτηση της λογαρίθμησης και της αντίστροφης διαδικασίας στις συγκεκριμένες θέσεις είναι κρίσιμης σημασίας. Δηλαδή, δεν μπορεί να αντικατασταθεί από άλλα βήματα ή να υπάρξει η λογαρίθμηση χωρίς της αντιστροφή της, διότι θα χαθούν δεδομένα της εικόνας. Η μαθηματική έκφραση που περιγράφει την διαδικασία της λογαρίθμησης και την επιστροφή στην αρχική κλίμακα σε μια ψηφιακή εικόνα δίνεται από την εξίσωση:

$$s = \Lambda(f) = \kappa \ln(f + 1)$$

$$e^s = (f + 1)^\kappa$$

Όπου,

s, είναι η τελική εικόνα μετά τη λογαρίθμηση,

f, είναι η αρχική εικόνα πριν τη λογαρίθμηση και

κ, είναι ο συντελεστής λογαρίθμησης.

Κατά την διαδικασία της λογαρίθμησης, η φωτεινότητα των σκούρων περιοχών αυξάνεται και των ανοιχτών αποχρώσεων μειώνεται. Συνεπώς αυξάνεται ο βαθμός ορατότητας στις λεπτομέρειες που βρίσκονται σε σκουρόχρωμες περιοχές της εικόνας. Η διακύμανση της φωτεινότητας στις σκούρες και στις ανοιχτές περιοχές γίνεται με μη ανάλογο τρόπο, με βάση ένα συντελεστή, κ . Εφόσον, το εύρος τιμών των αποχρώσεων σε κάθε ζεύγος χρωμάτων πριν την λογαρίθμηση είναι $[0, 255]$, τότε κατά τη λογαρίθμηση το εύρος θα γίνει $[0, 5.55]$. Επομένως, η λειτουργία του λογαρίθμου είναι να αυξάνει την κλίμακα μέτρησης της έντασης και ο συντελεστής που αντιπροσωπεύει την αλλαγή της κλίμακας, κ , εκφράζεται από τη σχέση:

$$\kappa = \frac{255}{\ln(1+maxpixel)}$$

Όπου,

$maxpixel$, είναι η μέγιστη τιμή εικονοστοιχείου της αρχικής εικόνας.

Η εκτέλεση της λογαρίθμησης και της επιστροφής από τον λογάριθμο στην αρχική κλίμακα τιμών εισάγεται στον κώδικα όπως φαίνεται στην εικόνα 3.4.

```
def log_transform(image):  
    log_image = np.log1p(image)  
    return log_image  
  
def exp(image):  
    exp_image = np.expm1(image)  
    return exp_image
```

Εικόνα 3.4. Λογαρίθμηση και επιστροφή στην αρχική κλίμακα τιμών

3.5 Μετασχηματισμός Fourier και αντίστροφος μετασχηματισμός Fourier

Σύμφωνα με την εικόνα 2.15, η λογαρίθμηση της εξίσωσης της εικόνας ακολουθείται από τη μετατροπή του χωρικού πεδίου στο πεδίο συχνότητας με τη βοήθεια του Μετασχηματισμού Fourier. Αντίστοιχα, πριν την επιστροφή από την λογαριθμική κλίμακα στην αρχική κλίμακα της εικόνας λαμβάνει χώρα ο αντίστροφος Μετασχηματισμός Fourier και συνεπώς η μετάβαση από το πεδίο της συχνότητας στο χωρικό πεδίο. Όπως αναλύθηκε στην ενότητα 2.4, το ομομορφικό φιλτράρισμα πραγματοποιείται στο πεδίο της συχνότητας διότι οι διεργασίες απλουστεύονται και ο υπολογιστικός χρόνος μειώνεται σε σχέση με το χωρικό πεδίο. Συγκεκριμένα, από την οικογένεια των Μετασχηματισμών Fourier θα υλοποιηθεί ο Γρήγορος Μετασχηματισμός Fourier (FFT) με την απαραίτητη θεωρητική μαθηματική ανάλυση όπως έχει περιγραφεί στην ενότητα 2.5. Οι εντολές που καλούνται υπολογίζουν τόσο το Διακριτό Μετασχηματισμό Fourier σε δύο διαστάσεις πάνω σε οποιοδήποτε άξονα του εξεταζόμενου πίνακα με τη βοήθεια του FFT όσο και τον αντίστροφό του. Η γενική μορφή των εντολών για τον FFT και τον αντίστροφο FFT, αντίστοιχα, είναι οι εξής [Num22]:

`np.fft.fft2(a, s, axes)`

`np.fft.ifft2(a, s, axes)`

Πίνακας 3.2

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & -4 \\ -3 & -2 & -1 \end{bmatrix}$$

Πίνακας 3.3

$$\begin{bmatrix} 2 & 0 & 1 \\ -4 & 3 & 4 \\ -1 & -3 & -2 \end{bmatrix}$$

Όπου,

a, είναι ο πίνακας εισόδου που μπορεί να είναι πραγματικός ή μιγαδικός,

s, είναι το σχήμα ή μήκος του κάθε μετασχηματισμένου άξονα. Αν οριστεί μικρότερο σχήμα από τις διαστάσεις του πίνακα εισόδου τότε το πλεόνασμα το τιμών εισόδου αφαιρείται, αν οριστεί μεγαλύτερο τότε στις επιπλέον θέσεις προστίθενται μηδενικά. Αν η συγκεκριμένη μεταβλητή δεν είναι ορισμένη το σχήμα του πίνακα εξόδου έχει τις ίδιες διαστάσεις με τον πίνακα εισόδου και axes, είναι οι άξονες πάνω στους οποίους γίνεται ο μετασχηματισμός.

Δεδομένου ότι το αποτέλεσμα του μετασχηματισμού μπορεί να είναι ένας μιγαδικός αριθμός, μετά τον αντίστροφο μετασχηματισμό, δηλαδή την επιστροφή στο χωρικό πεδίο, διατηρείται μόνο το πραγματικό μέρος του αποτελέσματος και αφαιρείται το φανταστικό. Η εντολή που καθιστά δυνατό αυτό τον υπολογισμό είναι η `np.real`. Στην εικόνα 3.5 παρουσιάζεται ο κώδικας που πραγματοποιεί την παραπάνω διαδικασία.

```
fft_image = np.fft.fft2(log_image)

inverse_fft_image = np.real(np.fft.ifft2(reverse_shift_image, axes=(0,1)))
```

Εικόνα 3.5. Μετασχηματισμός Fourier και αντίστροφος Μετασχηματισμός Fourier

3.6 Μεταφορά στοιχείων με μηδενικές συχνότητες στο κέντρο και επιστροφή

Η επόμενη διαδικασία αφορά την μεταφορά των στοιχείων από μια περιοχή της εικόνας σε μια άλλη. Σκοπός της διαδικασίας είναι ομοιόμορφη κατανομή των στοιχείων της εικόνας με βάση τη συχνότητά τους ξεκινώντας από το κέντρο της εικόνας, όπου τοποθετούνται τα στοιχεία με μηδενική συχνότητα. Απομακρυνόμενα από το κέντρο με ακτινική κατεύθυνση μεταφέρονται τα στοιχεία με αύξουσα σειρά στην απόλυτη τιμή της συχνότητάς τους. Κατά αυτόν τον τρόπο έχει επιτευχθεί μια οργανωμένη κατάταξη και μπορεί να εφαρμοστεί ένα φίλτρο που δρώντας σε ορισμένο μήκος ακτίνας θα επηρεάζει ομοιόμορφα τις συχνότητες της εικόνας. Για παράδειγμα, έστω ότι ο αρχικός πίνακας που υποθετικά αναπαριστά μια εικόνα είναι ο εξής (3.2):

Τότε, ο πίνακας μετά τη μεταφορά των μηδενικών στοιχείων στο κέντρο του φάσματος στον άξονα x θα είναι (3.3):

Η διαδικασία μεταφοράς των στοιχείων της εικόνας για την ομοιόμορφη επεξεργασία τους είναι μονόπλευρη, συνεπώς μετά την εφαρμογή του φίλτρου θα πρέπει να λάβει χώρα και η επι-

στροφή των στοιχείων στις αρχικές τους θέσεις. Εάν η επιστροφή των στοιχείων δεν συμβεί με ορθό τρόπο ή παραλειφθεί, η τελική εικόνα θα έχει μορφοποιηθεί αλλά τα κύρια στοιχεία της δεν θα ταυτίζονται με την αρχική, γιατί κομμάτια της εικόνας θα παραμένουν μετατοπισμένα σε τυχαία σημεία. Οι υπολογιστικές εντολές που ακολουθούνται για την εφαρμογή της μετατόπισης και της επιστροφής έχουν τη γενική μορφή [Num22]:

```
np.fft.fftshift(x, axes)
np.fft.ifftshift(x, axes)
```

Όπου,

x, είναι ο πίνακας εισόδου και

axes, είναι ο άξονας πάνω στον οποίο εφαρμόζεται η μετατόπιση. Αν δεν οριστεί η μεταβλητή του άξονα τότε η μεταφορά πραγματοποιείται σε όλους τους άξονες.

Ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση της μεταφοράς και την επιστροφή των στοιχείων της εικόνας στις αρχικές του θέσεις φαίνεται στην εικόνα 3.6.

```
shift_image = np.fft.fftshift(fft_image)

reverse_shift_image = np.fft.ifftshift(filtered_image)
```

Εικόνα 3.6. Μεταφορά στοιχείων με μηδενικές συχνότητες στο κέντρο της εικόνας και επιστροφή

3.7 Εφαρμογή φίλτρου

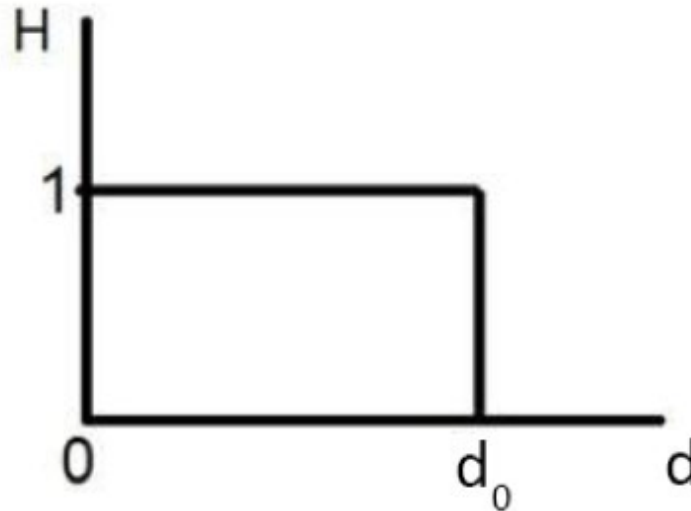
Το κεντρικό γεγονός στη διαδικασία του φιλτραρίσματος είναι η εφαρμογή του επιθυμητού φίλτρου. Τα είδη των φίλτρων διακρίνονται ανάλογα με τη μαθηματική εξίσωση που περιγράφει τον τρόπο που διενεργείται το φιλτράρισμα σε συγκεκριμένες συχνότητες. Ένα από τα πιο βασικά και απλά φίλτρα είναι το Ιδεατό φίλτρο (Ideal filter), που μαθηματικά εκφράζεται ακολουθώντας από την πρώτη διακλαδισμένη εξίσωση για τις περιπτώσεις των χαμηλοπερατών και την δεύτερη εξίσωση υψηλοπερατών συχνοτήτων [Che20]:

$$H(\omega, u) = \begin{cases} 1, & d(\omega, u) \leq d_0 \\ 0, & \text{if } d(\omega, u) > d_0 \end{cases}$$

$$H(\omega, u) = \begin{cases} 0, & d(\omega, u) \leq d_0 \\ 1, & \text{if } d(\omega, u) > d_0 \end{cases}$$

Πρακτικά, το Ιδεατό φίλτρο λειτουργεί επιτρέποντας ή απαγορεύοντας τη διέλευση συχνοτήτων από ένα κύκλο διαμέτρου d_0 . Η διάκριση των διαφορετικών συχνοτήτων με βάση μια καθορισμένη διάμετρο είναι δυνατή λόγω της παράταξης των στοιχείων με αύξουσα συχνότητα ακτινικά ως προς το κέντρο, όπως περιγράφηκε στην παραπάνω ενότητα. Για την περίπτωση του χαμηλοπερατού φίλτρου, εάν το εξεταζόμενο στοιχείο βρίσκεται σε απόσταση μικρότερη από d_0 , τότε

σημαίνει ότι η τιμή της συχνότητάς του ανήκει στο εύρος των επιτρεπτών συχνοτήτων και επιτρέπεται η διέλευσή του. Αντίστροφα, αν βρίσκεται σε απόσταση μεγαλύτερη από d_0 , τότε η τιμή της συχνότητας είναι μεγαλύτερη από το επιτρεπτό όριο, η εξίσωση $H(\omega, u)$ λαμβάνει την τιμή 0 και το φίλτρο αποκλείει την εξεταζόμενη συχνότητα. Η γραφική παράσταση του Ιδεατού φίλτρου είναι μια βηματική μεταβολή, όπως παρουσιάζεται στην εικόνα 3.7 για την περίπτωση του χαμηλοπερατού φίλτρου.



Εικόνα 3.7. Γραφική παράσταση χαμηλοπερατού Ιδεατού φίλτρου [Tut22]

Ένα πρόβλημα που υπεισέρχεται κατά τη χρήση του Ιδεατού φίλτρου είναι η κατάσταση που κυριαρχεί ακριβώς στην τιμή d_0 , δηλαδή στη μετάβαση από την τιμή 1 στην τιμή 0. Το φαινόμενο αυτό ονομάζεται κλυδωνισμός (ringing effect) ή φαινόμενο Gibbs και οπτικά εμφανίζεται σαν μια ατέλεια με τη μορφή κύματος στα αιχμηρά σημεία της εικόνας ή σαν μια ασαφή χρωματική μετάπτωση. (εικόνα 3.8) [Ima22]. Πρακτικά, οφείλεται στην παραμόρφωση ή την απώλεια στοιχείων της εικόνας με υψηλές συχνότητες και παρατηρείται συχνά σε συμπιεσμένες εικόνες, φωτογραφίες MRI, εικόνες μετά τη διάδοση από αναλογικά κανάλια κ.ά.

Το φαινόμενο του κλυδωνισμού μπορεί να περιοριστεί εφαρμόζοντας το φίλτρο Gauss, αντί του Ιδεατού φίλτρου. Το φίλτρο Gauss ορίζεται ως η παλμική απόκριση μιας συνάρτησης Gauss, που δεν παρουσιάζει υπέρβαση σε βηματική μεταβολή εισόδου και μειώνει του χρόνους ανόδου και καθόδου. Η συμπεριφορά αυτή σχετίζεται άμεσα με την ιδιότητα του συγκεκριμένου φίλτρου να έχει τη μικρότερη δυνατή χρονική καθυστέρηση στα πλάτη (group delay) [FE03]. Γενικά, το φίλτρο Gauss τείνει να θολώνει την εικόνα δημιουργώντας την εντύπωση ομαλότερης μετάβασης από μια περιοχή της εικόνας σε μια άλλη και ο βαθμός εξομάλυνσης εξαρτάται από την τυπική απόκλιση. Στην εικόνα 3.9 παρουσιάζεται η σύγκριση του Ιδεατού φίλτρου με το φίλτρο Gauss.

Το φίλτρο Gauss, $H(\omega, u)$, έχει συμμετρικό σχήμα με κέντρο τις συντεταγμένες (ω, u) του επιπέδου στο πεδίο της συχνότητας και η συνάρτηση μεταφοράς του μπορεί να περιγραφεί παρακάτω από την πρώτη εξίσωση για την περίπτωση του χαμηλοπερατού φίλτρου και αντίστοιχα από την δεύτερη εξίσωση για το υψιπερατό φίλτρο.

$$H(\omega, u) = e^{-c \frac{d^2(\omega, u)}{2d_0^2}}$$

$$H(\omega, u) = (\gamma_H - \gamma_L)(1 - e^{-c \frac{d^2(\omega, u)}{2d_0^2}}) + \gamma_L$$



Εικόνα 3.8. Το φαινόμενο του κλυδωνισμού ή ringing effect [Ima22]

Η εξίσωση που φαίνεται πιο κάτω δίνει το συσχετισμό των διαστάσεων της εικόνας $M \times N$ στο καρτεσιανό επίπεδο με το πεδίο της συχνότητας [Fer19].

$$d(\omega, u) = \sqrt{\left(\omega - \frac{M}{2}\right)^2 + \left(u - \frac{N}{2}\right)^2}$$

Όπου,
 $d(\omega, u)$, είναι η ακτίνα, δηλαδή η απόσταση των συντεταγμένων (ω, u) από την αρχή των αξόνων,

d_0 , είναι η συχνότητα αποκοπής,

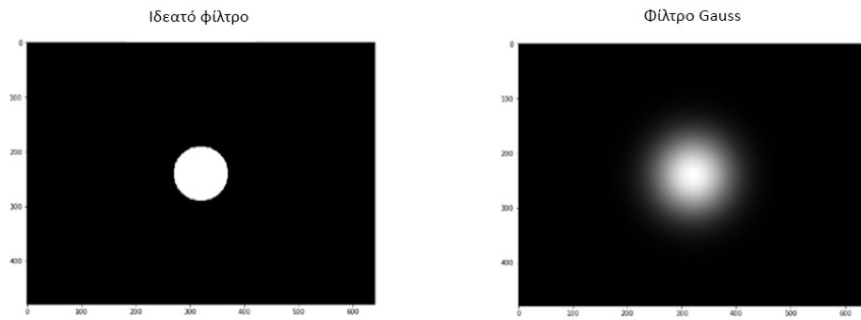
γ_H, γ_L , είναι η υπέρβαση υψηλών και χαμηλών συχνοτήτων, αντίστοιχα και ισχύει

$$\gamma_H > \gamma_L > 0$$

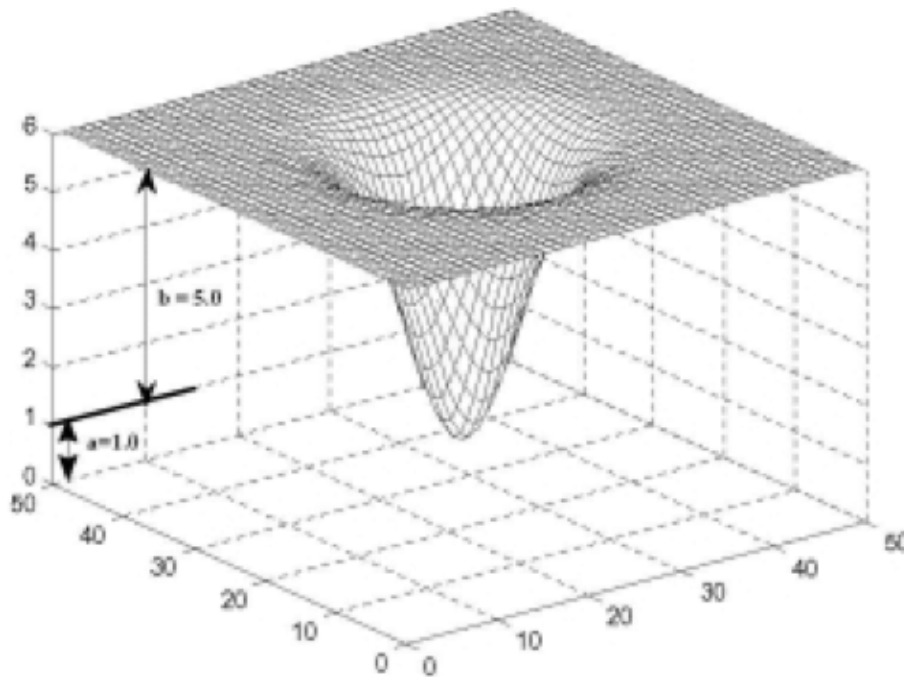
c , είναι μια σταθερή που καθορίζει την κλίση της καμπύλης.

Το φίλτρο που χρησιμοποιείται στην παρούσα διπλωματική εργασία είναι το υπερπαρατό φίλτρο Gauss που μειώνει τον όρο της φωτεινότητας και εντείνει την ανακλαστικότητα ταυτόχρονα. Με άλλα λόγια, τα στοιχεία της εικόνας φαίνονται λιγότερο φωτεινά και τονίζονται οι ακμές και οι λεπτομέρειες. Η γραφική απεικόνιση ενός υπερπαρατού φίλτρου Gauss παρουσιάζεται στην εικόνα 3.10.

Οι τέσσερις παράμετροι $c, d_0, \gamma_H, \gamma_L$ είναι οι μεταβλητές που ρυθμίζονται από το χρήστη κατά την εφαρμογή του ομομορφικού φιλτραρίσματος και συνεπώς η κατανόησή τους είναι κρίσιμη



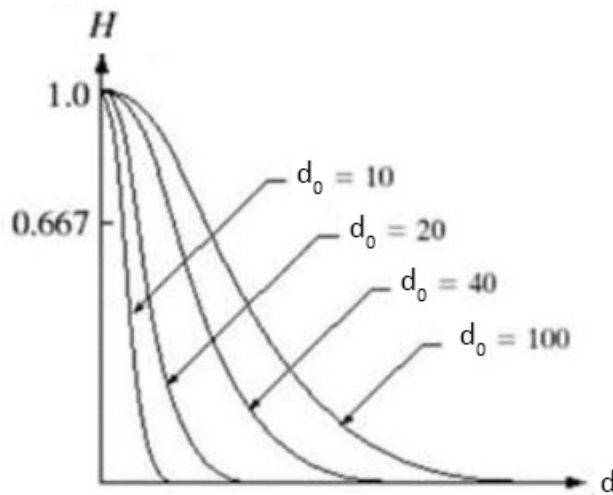
Εικόνα 3.9. Σύγκριση Ιδεατού φίλτρου και φίλτρου Gauss [Che20]



Εικόνα 3.10. Υπιπερατό φίλτρο Gauss [Raj07]

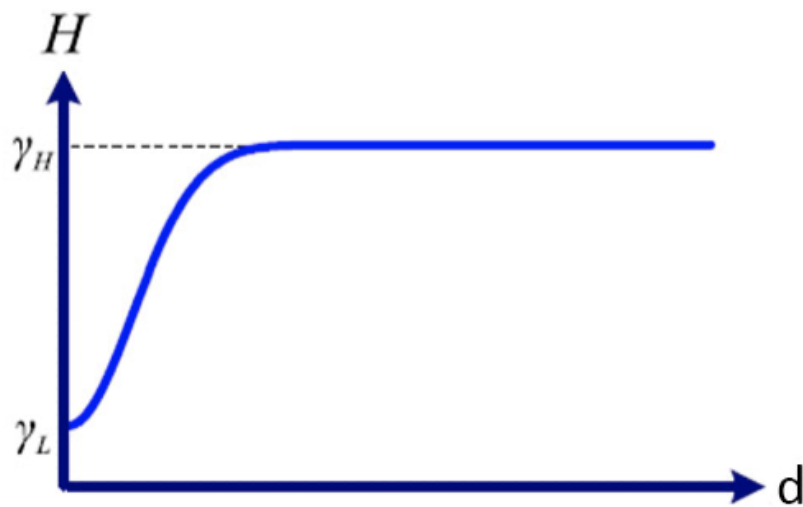
σημασίας για την εκτέλεση της επεξεργασίας της εικόνας. Η παράμετρος c καθορίζει την κλίση της καμπύλης, δηλαδή το σχήμα της καμπάνας Gauss. Η συχνότητα αποκοπής d_0 αναφέρεται στο οριακό σημείο στην απόκριση του συστήματος κατά το οποίο το σήμα δεν διαπερνάει το φίλτρο αλλά ανακλάται ή αποσβένεται. Στην περίπτωση του Ιδεατού φίλτρου η συχνότητα d_0 είναι το οριακό σημείο μεταξύ της διακλαδισμένης συνάρτησης, ενώ στο φίλτρο Gauss η d_0 έχει μια μοναδική τιμή. Στην εικόνα 3.11 συγκρίνονται για ένα χαμηλοπερατό φίλτρο Gauss διαφορετικές τιμές συχνοτήτων αποκοπής. Όσο υψηλότερη είναι η τιμή της συχνότητας αποκοπής τόσο θα αυξάνεται το όριο επιτρεπτής διέλευσης των συχνοτήτων δια μέσω του φίλτρου.

Οι παράμετροι γ_H , γ_L αφορούν το ύψος και το επίπεδο της καμπύλης, αντίστοιχα. Γραφικά, οι δύο μεταβλητές φαίνονται στην εικόνα 3.12, όπου παρουσιάζεται η τομή του φίλτρου Gauss. Εάν η γ_H οριστεί μεγαλύτερη της μονάδας και η γ_L μικρότερη, το φίλτρο τείνει να μειώνει τη συνεισφορά του όρου της φωτεινότητας, που αφορά κυρίως τα στοιχεία χαμηλών συχνοτήτων, και να αυξάνει την ανακλαστικότητα, που σχετίζεται με τα στοιχεία υψηλών συχνοτήτων [Sal12]. Γενικά, όσο μειώνεται η τιμή της γ_L , θα παρεμποδίζονται εντονότερα οι χαμηλές συχνότητες, συνεπώς η φωτεινότητα, και αντίστοιχα όσο αυξάνεται η γ_H θα εντείνεται η επίδραση της ανακλαστικότητας και των υψηλών συχνοτήτων. Όμως, το εύρος των τιμών στις οποίες μπορούν να κυμαίνονται οι γ_H , γ_L είναι περιορισμένο, καθώς σε πολύ χαμηλές τιμές γ_L θα υπάρξει απώλεια δεδομένων στις



Εικόνα 3.11. Σύγκριση διαφορετικών συχνοτήτων αποκοπής [Tut22]

ήπιες ακμές και αντίστοιχα σε υπερβολικά υψηλές τιμές της γH θα αμβλύνεται ο θόρυβος. Εν τέλει, η επιλογή των τιμών γ_H, γ_L είναι ένας συμβιβασμός μεταξύ των ορίων τους και κάθε διαφορετικός συνδυασμός τιμών επιφέρει διαφορετικά αποτελέσματα στο φίλτρο [FZ11].



Εικόνα 3.12. Η τομή του φίλτρου Gauss [FZ11]

Ο συσχετισμός του φίλτρου με την εικόνα γίνεται μέσω της πράξης της συνέλιξης της εξίσωσης του φίλτρου, $H(\omega, u)$, με την εξίσωση της εικόνας που έχει υποστεί Μετασχηματισμό Fourier, $T(\omega, u)$. Μαθηματικά η πράξη μπορεί να περιγραφεί ως:

$$S(\omega, u) = H(\omega, u) * T(\omega, u) = H(\omega, u)T_i(\omega, u) + H(\omega, u)T_r(\omega, u)$$

Όπου,

$T_i(\omega, u)$, $T_r(\omega, u)$ είναι η Μετασχηματισμοί Fourier των όρων της φωτεινότητας ($\ln i$) και της ανακλαστικότητας ($\ln r$) μετά τη λογαρίθμηση.

Υπολογιστικά η κατασκευή του φίλτρου αρχίζει με την εντολή `img.shape`, με την οποία λαμβάνονται οι διαστάσεις τις εικόνας στη μορφή ενός πίνακα. Η γενική μορφή της εντολής είναι [Num22]:

```
np.meshgrid (xi, sparse, indexing)
```

Όπου,

`xi`, είναι οι διαστάσεις του πλέγματος,

`sparse`, δηλώνει αν οι διαστάσεις του πλέγματος παραμένουν αμετάβλητες (επιλογή `False`) ή αλλάζουν (επιλογή `True`) και

`indexing` ή δείκτες, είναι το πεδίο του πλέγματος, που μπορεί να είναι το καρτεσιανό επίπεδο, `xy`, ή το επίπεδο των μιγαδικών αριθμών, `ij`.

Στη συνέχεια ορίζονται τα φίλτρα Gauss σύμφωνα με τις εξισώσεις που είδαμε στην αρχή της ενότητας και για την αποφυγή διαίρεσης με μηδενικές τιμές, ορίζεται η τιμή 1 όταν το d_0 ισούται με μηδέν. Αφού δημιουργηθεί το επιθυμητό φίλτρο, πραγματοποιείται η πράξη της συνέλιξης σύμφωνα με την εξίσωσή της και με την εντολή `np.multiply`. Η εντολή `np.multiply` υλοποιεί την συνέλιξη πολλαπλασιάζοντας δύο πίνακες στοιχείο προς στοιχείο. Στην εικόνα 3.13 παρατίθεται ο υπολογιστικός κώδικας για το στάδιο της εφαρμογής του υψιπερατού φίλτρου Gauss.

```
def gaussian_lp(img, c, D0):
    P,Q = img.shape
    centerX = P/2
    centerY = Q/2
    H = np.zeros(img.shape)
    U, V = np.meshgrid(range(int(P)), range(int(Q)), sparse=False, indexing='ij')
    Duv = (((U-centerX)**2+(V-centerY)**2)).astype(float)
    H = np.exp((-c*Duv/(D0**2))
    return H

def gaussian_hp(gammah,gammal,c,D0,img):
    lp = gaussian_lp(img, c, D0)
    base = (gammah - gammal)*(1-lp) + gammal
    return base, lp

#this check is to avoid division by zero
if d0 == 0:
    d0 = 1

# create filter
gauss, lp = utl.gaussian_hp(ghigh, glow, cc, d0, shift_image)

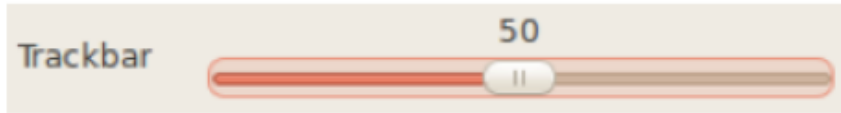
# apply filter on image
filtered_image = np.multiply(shift_image,gauss)
```

Εικόνα 3.13. Εφαρμογή υψιπερατού φίλτρου Gauss

3.8 Δημιουργία παραθύρου με τις μπάρες εργασιών

Για την διευκόλυνση της επεξεργασίας της εικόνας δημιουργείται ένα παράθυρο στο οποίο εμφανίζεται η επιλεγμένη εικόνας προς μορφοποίηση και τέσσερις μπάρες εργασιών (trackbars). Η κάθε μπάρα απεικονίζει μια παράμετρο του ομομορφικού φιλτραρίσματος, μέσω της οποίας ο χρήστης μπορεί να ρυθμίσει την συγκεκριμένη μεταβλητή. Συγκεκριμένα, οι μπάρες εργασιών θεωρούνται στοιχεία του γραφικού περιβάλλοντος του χρήστη (GUI) και οι άκρες τους αντιστοιχούν στο εύρος της παραμέτρου που αναπαρίσταται. Η μετατόπιση της τιμής ανάμεσα στις δύο ακραίες τιμές της μπάρας είναι γραμμικώς ανάλογη της μετατόπισης της παραμέτρου στο εύρος τιμών της. Η γραφική απεικόνιση μιας μπάρας εργασίας παρουσιάζεται στην εικόνα 3.14.

Οι εντολές που χρησιμοποιούνται για τη δημιουργία του παραθύρου και των μπαρών είναι οι `cv.namedWindow` και `cv.createTrackbar`, αντίστοιχα. Η γενική μορφή της εντολής `cv.createTrackbar` περιέχει τους εξής όρους [Doc22]:



Εικόνα 3.14. Η μπάρα εργασίας (trackbar)

Πίνακας 3.4. Παράμετροι ομομορφικού φιλτραρίσματος

Παράμετρος	Εύρος τιμών	Τυπικές τιμές φίλτρου
γ_L	[0, 100]	4
γ_H	[0, 200]	30
c	[0, 500]	50
d_0	[0 (1), 200]	20

`cv2.createTrackbar (TrackbarName, Window name, alpha_slider, alpha_slider_max, on_trackbar)`

Όπου,

Trackbar name ή όνομα μπάρας εργασιών, είναι η ονομασία που φαίνεται στο παράθυρο,

Window name ή όνομα παραθύρου, είναι η ονομασία του παραθύρου μέσα στο οποίο τοποθετείται η μπάρα,

alpha_slider ή αρχή σημείου κύλισης, είναι η αρχική τιμή της παραμέτρου που εμφανίζεται στην μπάρα εργασίας,

alpha_slider_max ή τέλος σημείου κύλισης, είναι η ανώτερη τιμή στην μπάρα εργασίας, δηλαδή η μεγαλύτερη τιμή που μπορεί να λάβει η εξεταζόμενη παράμετρος και

on_trackbar ή συνάρτηση παραμέτρου, είναι η συνάρτηση που υπολογίζει την νέα τιμή της παραμέτρου όταν μετακινείται το επιλεγμένο σημείο πάνω στη μπάρα εργασίας.

Οι τέσσερις παράμετροι του ομομορφικού φιλτραρίσματος που μπορεί να ρυθμίζει ο χρήστης με τα αντίστοιχα εύρη τιμών τους και τις τυπικές τιμές που λαμβάνουν παρουσιάζονται στον πίνακα 3.4.

Ο κώδικας που χρησιμοποιήθηκε για τη δημιουργία των τεσσάρων μπαρών εργασιών παρατίθεται στην εικόνα 3.15.

```
def gui(image):
    cv.namedWindow('Homomorphic Filter', cv.WINDOW_NORMAL)
    start_ghigh = 30 # slider start position
    max_ghigh = 200 # maximal slider position
    start_glow = 4 # slider start position
    max_glow = 100 # maximal slider position
    start_c = 50 # slider start position
    max_c = 500 # maximal slider position
    start_d0 = 20 # slider start
    max_d0 = 200 # maximal slider position
    cv.createTrackbar('ghigh (/10)', 'Homomorphic Filter', start_ghigh, max_ghigh, (lambda a: None))
    cv.createTrackbar('glow (/10)', 'Homomorphic Filter', start_glow, max_glow, (lambda a: None))
    cv.createTrackbar('cc (/10)', 'Homomorphic Filter', start_c, max_c, (lambda a: None))
    cv.createTrackbar('d0', 'Homomorphic Filter', start_d0, max_d0, (lambda a: None))
```

Εικόνα 3.15. Δημιουργία παραθύρου με τις μπάρες εργασιών

3.9 Αποθήκευση νέας εικόνας και έξοδος

Η αποθήκευση της νέας εικόνας και η έξοδος συμβαίνει με μια σειρά απλών εντολών που καταλαμβάνουν το τελευταίο κομμάτι του κώδικα. Αρχικά, χρησιμοποιείται η εντολή `cv.imshow`, με την οποία η εικόνα τοποθετείται στο παράθυρο που δημιουργήθηκε. Με αυτόν τον τρόπο, ο χρήστης έχει τη δυνατότητα να παρακολουθεί τις αλλαγές στην εικόνα την ώρα που ρυθμίζει τις παραμέτρους. Στη συνέχεια, ορίζεται η εντολή `cv.waitKey`, η οποία δίνει τη δυνατότητα αναμονής για 27 δευτερόλεπτα μέχρι ο χρήστης να επιλέξει ένα σημείο στο ανοικτό παράθυρο OpenCV.

Όταν λάβει τέλος η διαδικασία εφαρμογής του φίλτρου και η επιλογή των παραμέτρων, τότε με την εντολή `cv.imwrite`, η εικόνα μπορεί να αποθηκευτεί στην επιθυμητή μορφή στο φάκελο που διενεργείται η διαδικασία της επεξεργασίας. Η τελευταία εντολή είναι η `cv.destroyAllWindows`, με την οποία, αφού τελειώσουν όλες οι διεργασίες, κλείνει το παράθυρο OpenCV και τερματίζεται η επεξεργασία της εικόνας. Στην εικόνα 3.16 παρουσιάζεται ο σχετικός κώδικας για την πραγματοποίηση των παραπάνω εντολών.

```
cv.imshow("Homomorphic Filter", img_homomorphic)
k = cv.waitKey(1)
if k == 27:
    save_path = easygui.filesavebox()
    cv.imwrite(save_path, img_homomorphic)
    cv.destroyAllWindows()
    break
```

Εικόνα 3.16. Αποθήκευση νέας εικόνας και έξοδος

3.10 Συνολική διαδικασία

Στις προηγούμενες ενότητες αναλύθηκαν διεξοδικά και ανεξάρτητα τα βήματα 1-11 της ενότητας 3.1. Η συνολική διαδικασία της επεξεργασίας μιας εικόνας με ομομορφικό φιλτράρισμα απαιτεί την εκτέλεση όλων των βημάτων με την ορθή σειρά. Στην εικόνα 3.17 παρατίθεται το εκτελέσιμο αρχείο `hy` με τον αλγόριθμο που πραγματοποιεί αυτήν την εντολή.

```

import sys
import cv2 as cv
import numpy as np
import easygui

import Utilities as utl

imgpath = utl.load_image()

image, color, Cr, Cb = utl.read_img(imgpath)

log_image = utl.log_transform(image)

utl.gui(log_image)

while(True):
    ghigh = cv.getTrackbarPos('ghigh (/10)', 'Homomorphic Filter') / 10
    glow = cv.getTrackbarPos('glow (/10)', 'Homomorphic Filter') / 10
    cc = cv.getTrackbarPos('cc (/10)', 'Homomorphic Filter') / 10
    d0 = cv.getTrackbarPos('d0', 'Homomorphic Filter')

    if d0 == 0:
        d0 = 1

    fft_image = np.fft.fft2(log_image)

    shift_image = np.fft.fftshift(fft_image)

    gauss, lp = utl.gaussian_hp(ghigh, glow, cc, d0, shift_image)

    filtered_image = np.multiply(shift_image,gauss)

    reverse_shift_image = np.fft.ifftshift(filtered_image)

    inverse_fft_image = np.real(np.fft.ifft2(reverse_shift_image, axes=(0,1)))

    exp_image = utl.exp(inverse_fft_image)
    img_homomorphic = utl.normalize(exp_image, color, Cr, Cb)

    cv.imshow("Homomorphic Filter", img_homomorphic)
    k = cv.waitKey(1)
    if k == 27:
        save_path = easygui.filesavebox()
        cv.imwrite(save_path, img_homomorphic)
        cv.destroyAllWindows()
        break

```

Εικόνα 3.17. Συνολική διαδικασία ομομορφικού φιλτραρίσματος

Κεφάλαιο 4

Αποτελέσματα

4.1 Εκτέλεση κώδικα

Ο κώδικας για την επεξεργασία της εικόνας με τη μέθοδο του ομομορφικού φιλτραρίσματος είναι γραμμένος σε υπολογιστική γλώσσα Python, ως εκ τούτου μπορεί να εκτελεστεί σε συγκεκριμένα λειτουργικά συστήματα. Το κύριο λειτουργικό σύστημα που χρησιμοποιείται είναι τα Linux, το οποίο είτε υπάρχει ως το κύριο λειτουργικό πρόγραμμα του υπολογιστή είτε εγκαθίσταται ως δευτερεύον στο πρωταρχικό [Kan11]. Παραδείγματος χάρη, εάν το κύριο λειτουργικό σύστημα είναι τα Windows, τα Linux μπορούν να προσομοιαστούν μέσω ενός υπερεπόπτη ανοιχτού κώδικα, όπως το VirtualBox. Ο υπερεπόπτης δρα ως μια εικονική μηχανή που αναπαράγει όλες τις λειτουργίες του δευτερεύοντος λειτουργικού συστήματος μέσα στο πρώτο επιτρέποντας τη διαχείριση μεμονωμένων περιβαλλόντων χρήσης. Μια άλλη εναλλακτική για την λειτουργία των Linux ως δευτερεύον σύστημα είναι η χρήση επιπέδων συμβατότητας (compatibility layer) όπως το Windows Subsystem for Linux (WSL) και το Wine Is Not an Emulator (WINE), με τη βοήθεια των οποίων εκτελούνται δυαδικά αρχεία και εντολές των Linux μέσα στο περιβάλλον Windows χωρίς την προσομοίωση ολόκληρου του λειτουργικού συστήματος όπως στην περίπτωση των εικονικών μηχανών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε το λειτουργικό σύστημα Linux ως κύριο λειτουργικό σύστημα και συγκεκριμένα η διανομή Ubuntu.

Καταρχάς, μέσα στο περιβάλλον Ubuntu εγκαθίστανται τα τρία αρθρώματα που απαιτούνται για την εκτέλεση του κώδικα, Python, OpenCV και Tkinter. Η θεωρητική ανάλυση των υπολογιστικών εργαλείων και των πακέτων που περιέχουν αναλύθηκε στις ενότητες 2.8 και 3.2. Η γλώσσα Python είναι εγγενώς εγκατεστημένη στο λειτουργικό σύστημα, ενώ τα αρθρώματα OpenCV και Tkinter εισάγονται και επαληθεύεται η εγκατάστασή τους με τις υπολογιστικές εντολές που φαίνονται στην εικόνα 4.1.

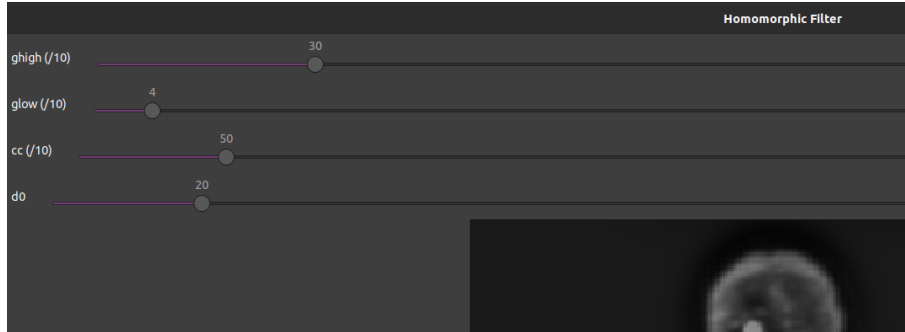
```
sudo apt install python3-opencv
sudo apt-get install python3-tk

python3 -c "import cv2; print(cv2.__version__)"
python3 -c "import tkinter; print(tkinter.TkVersion)"
```

Εικόνα 4.1. Εγκατάσταση των αρθρωμάτων OpenCV και Tkinter

Στη συνέχεια, πραγματοποιήθηκε η εκτέλεση του αλγορίθμου για το ομομορφικό φιλτράρισμα με την υπολογιστική εντολή `python3 hf.py path/to/image`, δηλαδή καλείται το εκτελέσιμο αρχείο `hf`. Αφού επιλεγθεί η εικόνα προς επεξεργασία, εμφανίζεται το παράθυρο με την εικόνα και τις ρυθμιζόμενες παραμέτρους του υπερερατού φίλτρου Gauss, όπως φαίνεται στην εικόνα 4.2. Οι τιμές των παραμέτρων που παρουσιάζονται στην εικόνα 4.2 αντιστοιχούν στις τυπικές τιμές του

πίνακα 3.4 για το ομομορφικό φιλτράρισμα. Παρόλα αυτά είναι δυνατή η μεταβολή τους μέσα στα επιτρεπτά όρια των μπαρών εργασίας και η επιλογή του κατάλληλου συνδυασμού τιμών. Μετά την επιλογή των βέλτιστων τιμών η νέα μορφοποιημένη εικόνα μπορεί να αποθηκευτεί και την εντολή Esc. Οι εικόνες που εισάχθηκαν και υπέστησαν μορφοποίηση ήταν συνολικά επτά, τρεις έγχρωμες και τέσσερις ασπρόμαυρες με γκρι αποχρώσεις.



Εικόνα 4.2. Παράθυρο επεξεργασίας με τις ρυθμιζόμενες παραμέτρους

4.2 Ακραίες τιμές παραμέτρων

Αρχικά εξετάζονται οι ακραίες τιμές που μπορούν να λάβουν οι παράμετροι εφαρμοσμένες σε μία εικόνα, με σκοπό την καλύτερη κατανόησή τους. Κάθε παράμετρος λαμβάνει την κατώτατη και την ανώτατη τιμή της, διατηρώντας σταθερές τις υπόλοιπες παραμέτρους στις τυπικές τιμές. Στην εικόνα 4.3 παρουσιάζεται η κατώτατη τιμή κάθε παραμέτρου εφαρμοσμένη σε μια ασπρόμαυρη εικόνα που αναπαριστά μια τομογραφία εκπομπής ποζιτρονίων (PET).

Οι κατώτατες τιμές των παραμέτρων γ_L , γ_H , c είναι μηδέν, ενώ της παραμέτρου d_0 ορίζεται στη μονάδα. Όταν η τιμή της παραμέτρου c είναι μηδενική, τότε ο εκθετικός όρος ισούται με ένα και η εξίσωση:

$$H(\omega, u) = (\gamma_H - \gamma_L)(1 - e^{-c \frac{d^2(\omega, u)}{2d_0^2}}) + \gamma_L$$

απλοποιείται ως εξής:

$$H(\omega, u) = (\gamma_H \gamma_L)(1 - 1) + \gamma_L \Rightarrow$$

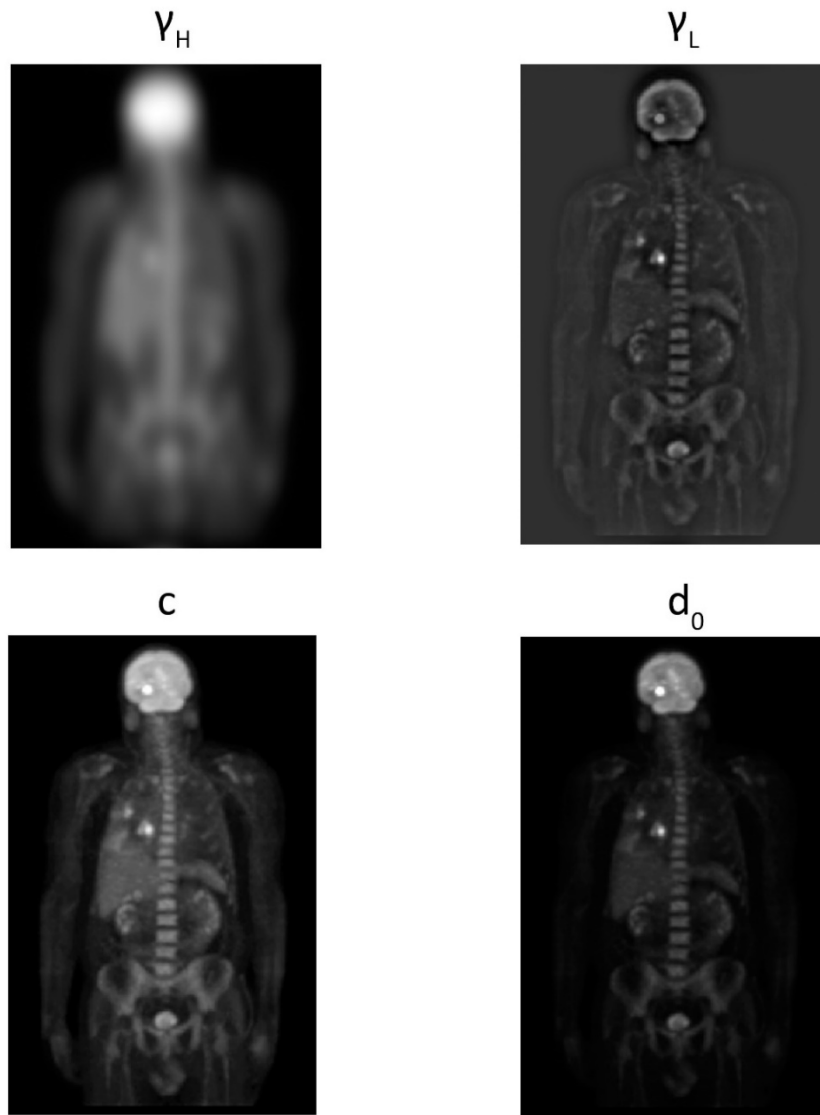
$$H(\omega, u) = \gamma_L$$

Η τελική εικόνα εξαρτάται μόνο από την τιμή της παραμέτρου γ_L και είναι ανεξάρτητη από τις άλλες παραμέτρους όπως αποδείχτηκε από την εξίσωση που φαίνεται παραπάνω. Όταν η τιμή γ_L λάβει την τιμή μηδέν, τότε το επίπεδο της καμπύλης Gauss ακουμπάει στον άξονα των τετμημένων και στο οπτικό αποτέλεσμα φαίνονται όλες οι λεπτομέρειες της εικόνας. Ενώ όταν η τιμή γ_H μηδενίζεται η καμπάνα Gauss αποκτά μικρότερο ύψος και η τελική εικόνα θολώνει. Η τιμή d_0 αναπαριστά το αριθμητικό όριο διέλευσης των συχνοτήτων, επομένως στην κατώτατη τιμή λιγότερες συχνότητες διαπερνούν το φίλτρο και οι λεπτομέρειες στην τελική εικόνα μειώνονται.

Όταν $\gamma_L = \gamma_H$, τότε η εξίσωση:

$$H(\omega, u) = (\gamma_H - \gamma_L)(1 - e^{-c \frac{d^2(\omega, u)}{2d_0^2}}) + \gamma_L$$

μετασχηματίζεται όπως φαίνεται παρακάτω:



Εικόνα 4.3. Κατώτατες τιμές ρυθμιζόμενων παραμέτρων

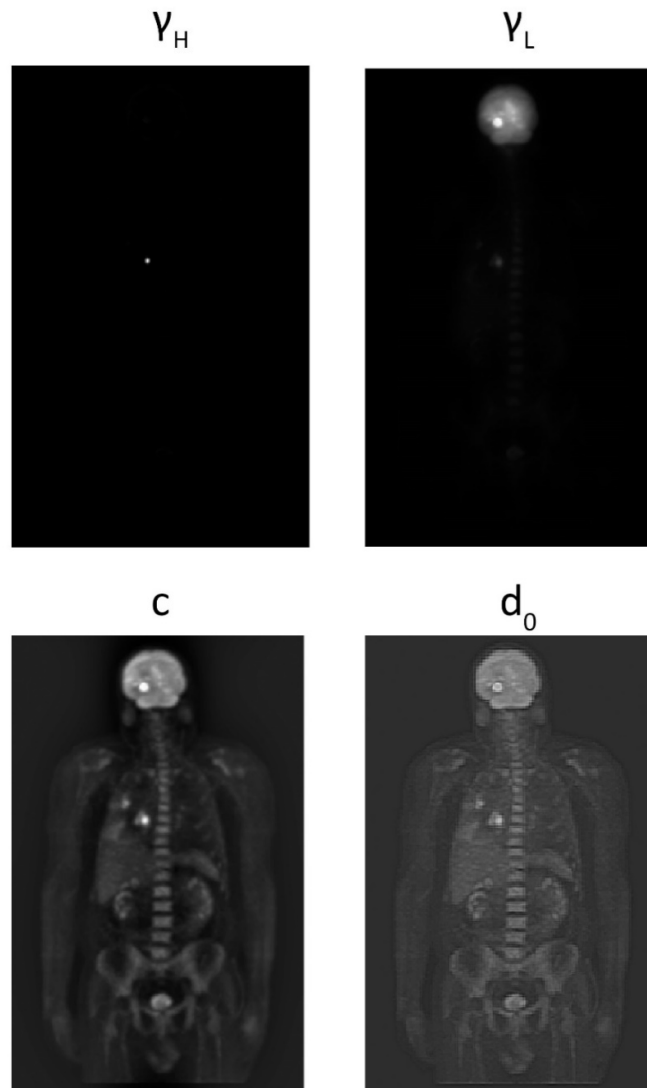
$$H(\omega, u) = (\gamma_H - \gamma_L)(1 - e^{-c \frac{d^2(\omega, u)}{2d_0^2}}) + \gamma_L \Rightarrow$$

$$H(\omega, u) = \gamma_L$$

Ομοίως με την πρώτη εξίσωση της ενότητας, η μόνη παράμετρος που επηρεάζει το φιλτράρισμα είναι η γ_L , ενώ οι υπόλοιπες μπορούν να μεταβάλλονται αφήνοντας ανεπηρέαστο το αποτέλεσμα.

Κατόπιν, εξετάζονται οι ανώτερες τιμές των ρυθμιζόμενων παραμέτρων 4.4 για συγκριτικούς λόγους με την 4.3.

Παρατηρείται ότι όταν οι παράμετροι γ_L, γ_H λάβουν τις ανώτερες τιμές του, το οπτικό αποτέλεσμα εμφανίζεται ως μια μαύρη εικόνα με ελάχιστα λευκά στοιχεία. Αυτό οφείλεται στο γεγονός ότι οι περισσότερες συχνότητες, στις οποίες οφείλονται οι λεπτομέρειες της εικόνας, δεν διαπερνούν το φίλτρο. Αντίθετα, όταν η παράμετρος d_0 έχει την ανώτερη τιμή της τότε το όριο διέλευσης των συχνοτήτων αμβλύνει και όλα τα στοιχεία της εικόνας είναι ορατά.



Εικόνα 4.4. Ανώτατες τιμές ρυθμιζόμενων παραμέτρων

4.3 Ομομορφικό φιλτράρισμα σε ασπρόμαυρη εικόνα

Το ομομορφικό φιλτράρισμα για την περίπτωση των ασπρόμαυρων εικόνων με γκρι αποχρώσεις εφαρμόζεται σε τέσσερις εικόνες και η αρχική και την τελική μορφή τους παρουσιάζεται στις εικόνες 4.5, 4.6.

Η πρώτη εικόνα που επεξεργάζεται είναι η φωτογραφία μιας κοπέλας στην οποία κυριαρχούν ανοικτές αποχρώσεις με απότομες μεταβάσεις στις μαύρες περιοχές της εικόνας που οφείλονται στην πηγή του φωτός στην φωτογραφία. Στην τελική εικόνα μετά την επεξεργασία, οι χρωματικές μεταβάσεις είναι ομαλές, οι επικρατέστερες αποχρώσεις είναι γκρι ενδιάμεσες στις ακραίες τιμές του λευκού και του μαύρου και συνολικά η εικόνα είναι πιο σκοτεινή από την αρχική.

Η δεύτερη εικόνα που υπόκειται μορφοποίηση αποτελεί μια τομογραφία εκπομπής ποζιτρονίων (PET), η οποία περιέχει σημαντικό θόρυβο και θολά περιγράμματα. Μετά την εφαρμογή του φίλτρου, τα στοιχεία είναι πιο διακριτά από το ανθρώπινο μάτι και οι λεπτομέρειες μπορούν να παρατηρηθούν ευκολότερα.

Στην τρίτη εικόνα που υφίσταται επεξεργασία με ομομορφικό φιλτράρισμα απεικονίζεται ένα σχέδιο με μαύρο χρώμα πάνω σε ανοιχτόχρωμο και άσπρο φόντο. Η επεξεργασία εξομαλύνει τη

Πριν την επεξεργασία



Μετά την επεξεργασία



Εικόνα 4.5. Ομομορφικό φιλτράρισμα σε ασπρόμαυρες εικόνες (μέρος πρώτο)

μετάβαση από το φόντο στο μαύρο σχέδιο, καθιστώντας πιο ανοιχτές τις σκούρες αποχρώσεις και, αντίστοιχα, πιο σκούρες τις ανοιχτές.

Η τελευταία ασπρόμαυρη εικόνα που μορφοποιείται αναπαριστά μια φωτογραφία στην οποία το φως έρχεται από το κέντρο ενώ στην εσωτερική περίμετρο τα χρώματα τείνουν στο μαύρο και στο σκούρο γκρι. Μετά την εφαρμογή του ομομορφικού φίλτρου, οι άσπρες και οι μαύρες αποχρώσεις έχουν απαλειφθεί εντελώς και έχουν αντικατασταθεί με ενδιάμεσα γκρι χρώματα που καθιστούν ευκρινέστερες τις λεπτομέρειες.

4.4 Ομομορφικό φιλτράρισμα σε έγχρωμη εικόνα

Η επεξεργασία έγχρωμων εικόνων υλοποιείται με όμοιο τρόπο όπως στην περίπτωση των ασπρόμαυρων εικόνων σε αποχρώσεις του γκρι. Οι αρχικές και οι τελικές μορφές των τεσσάρων εικόνων που επιλέχθηκαν να μορφοποιηθούν παρουσιάζονται στις εικόνες 4.7, 4.8. Όλες οι εικόνες που επιλέχθηκαν περιλαμβάνουν κυρίως μαύρο χρώμα με λίγα έντονα έγχρωμα στοιχεία διεσπαρμένα στο εσωτερικό της εικόνας. Οι έντονες χρωματικές διαφορές σε σκουρόχρωμες εικόνες, καθιστούν δυσδιάκριτες πολλές περιοχές με αποτέλεσμα να υπάρχει ανάγκη για χρωματική εξομάλυνση με τη βοήθεια του ομομορφικού φιλτραρίσματος.

Η πρώτη έγχρωμη εικόνα που επιλέχθηκε περιέχει μαύρες αποχρώσεις στα άκρα και σε μεμονωμένα σημεία εμφανίζονται αποχρώσεις του πράσινου, του πορτοκαλί και του άσπρου. Η διαδικασία της μορφοποίησης έχει ως αποτέλεσμα η εικόνα να αποκτήσει έγχρωμες αποχρώσεις λιγότερα κοντά στο μαύρο αυξάνοντας την συνολική φωτεινότητα και αντίθεση των χρωμάτων. Οι αποχρώσεις που συγκλίνουν μεταξύ τους επιτρέπουν στον θεατή να παρατηρήσει την εικόνα ολο-

Πριν την επεξεργασία



Μετά την επεξεργασία



Εικόνα 4.6. Ομομορφικό φιλτράρισμα σε ασπρόμαυρες εικόνες (μέρος δεύτερο)

κληρωμένα χωρίς να αποσπάται από τα στοιχεία με τις εντονότερες αποχρώσεις.

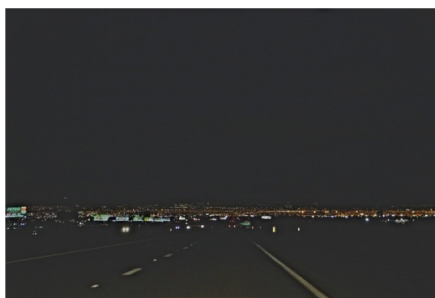
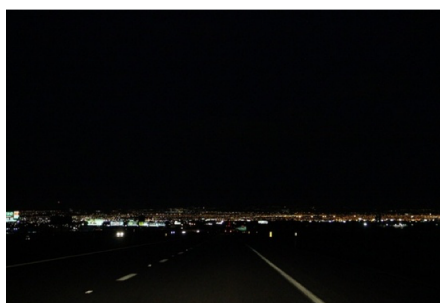
Η δεύτερη εικόνα που υφίσταται επεξεργασία απεικονίζει τα φώτα μιας πόλης στο βάθος της εικόνας με το μεγαλύτερο μέρος της να είναι μαύρο. Με μια πρώτη ματιά μετά την επεξεργασία η εικόνα δεν παρουσιάζει σημαντική διαφορά με την αρχική της μορφή, καθώς τα φωτεινά σημεία είναι ελάχιστα. Παρόλα αυτά σε ένα δεύτερο επίπεδο παρατήρησης, ο έντονος φωτισμός που συσσωρεύεται σε μια οριζόντια γραμμή στην εικόνα έχει μειωθεί σε μεγάλο βαθμό.

Η τρίτη εικόνα που μορφοποιείται αναπαριστά ένα διάδρομο με έντονη φωτεινότητα, αλλά σκοτάδι στο βάθος του διαδρόμου. Η επεξεργασία της εικόνας ισορροπεί τη φωτεινότητα, με αποτέλεσμα ο διάδρομος να προσεγγίζει τα πραγματικά χρώματα που απεικονίζεται και να είναι πιο ευδιάκριτο στο μάτι τα αντικείμενα που υπάρχουν στο τέλος του διαδρόμου.

Πριν την επεξεργασία



Μετά την επεξεργασία

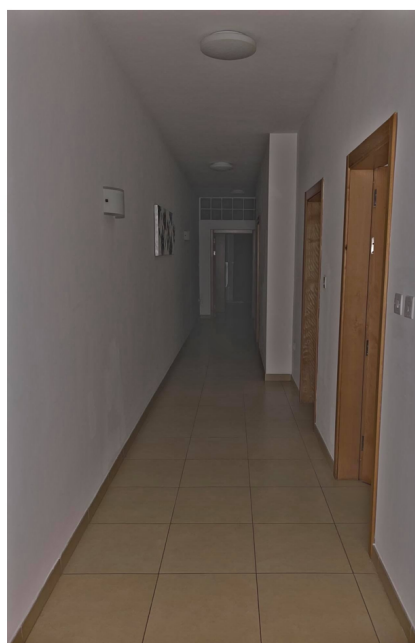


Εικόνα 4.7. Ομομορφικό φιλτράρισμα σε έγχρωμες εικόνες (μέρος πρώτο)

Πριν την επεξεργασία



Μετά την επεξεργασία



Εικόνα 4.8. Ομομορφικό φιλτράρισμα σε έγχρωμη εικόνα (μέρος δεύτερο)

Κεφάλαιο 5

Συμπεράσματα

Το θέμα της διπλωματικής εργασίας ήταν η συγγραφή και η εφαρμογή ενός κώδικα για την επεξεργασία εικόνας με την μέθοδο του ομομορφικού φιλτραρίσματος. Ο αλγόριθμος υλοποιήθηκε στην υπολογιστική γλώσσα Python σε λειτουργικό σύστημα Linux Ubuntu και η επεξεργασία των στοιχείων της εικόνας έγινε δυνατή με τη βοήθεια της βιβλιοθήκης OpenCV. Το φίλτρο που κατασκευάστηκε ήταν ένα υψιπερατό φίλτρο Gauss που λαμβάνει χώρα στο πεδίο των συχνοτήτων και χαρακτηρίζεται από τέσσερις παραμέτρους ρυθμιζόμενες από τον χρήστη. Οι ρυθμιζόμενες παράμετροι πέρα από τις τυπικές τιμές για την εξομάλυνση της φωτεινότητας και της ανακλαστικότητα της εικόνας, είχαν τη δυνατότητα να μεταβάλλονται σε ένα καθορισμένο εύρος τιμών. Η εικόνα προς μορφοποίηση και οι μεταβολές των τιμών των παραμέτρων ήταν ορατές μέσω ενός παραθύρου που λειτουργούσε ως επιφάνεια εργασίας κατά τη διάρκεια της επεξεργασίας.

Η υλοποίηση του κώδικα βασίστηκε στο θεωρητικό υπόβαθρο που αναλύθηκε και η αλληλουχία των βημάτων σε έναν αλγόριθμο που εξέταζε τη διαδικασία από την εισαγωγή και την ανάλυση της αρχικής εικόνας μέχρι την αποθήκευση της επεξεργασμένης εικόνας και την έξοδο. Η ανάλυση του αλγορίθμου μελέτησε μεμονωμένα όλα τα στάδια της διαδικασίας και την μαθηματική τους επεξήγηση, όπως τη λογαρίθμηση της εξίσωσης της εικόνας και το Μετασχηματισμό Fourier για τη μετάβαση από το χωρικό πεδίο στο πεδίο των συχνοτήτων.

Αρχικά, σε μια εικόνα επιβλήθηκε το φίλτρο με τις ακραίες τιμές των παραμέτρων και παρουσιάστηκαν οχτώ διαφορετικές περιπτώσεις, μια για κάθε ανώτερη και κατώτερη τιμή των τεσσάρων διαφορετικών μεταβλητών. Συμπεραίνεται ότι οι ακραίες τιμές συνεισφέρουν τόσο στην πλήρη κατανόηση όσο και στο βαθμό επίδρασης της κάθε μεταβλητής, διότι το οπτικό αποτέλεσμα μπορεί να εξηγηθεί με βάση τη μαθηματική εξίσωση που ορίζει το φίλτρο και την φυσική σημασία των παραμέτρων.

Στη συνέχεια, σε μια έγχρωμη και σε μια ασπρόμαυρη εικόνα με γκρι αποχρώσεις εφαρμόστηκε το ομομορφικό φιλτράρισμα τρεις φορές διαδοχικά, με σκοπό τη εξέταση της επιρροής του. Και στις δύο περιπτώσεις που μελετήθηκαν παρατηρείται ότι σταδιακά οι αποχρώσεις τις εικόνας συγκλίνουν όλο και περισσότερο και η φωτεινότητα μειώνεται. Μετά την τρίτη επεξεργασία, οι απότομες χρωματικές αλλαγές έχουν απαλειφθεί και το συνολικό χρώμα της εικόνας τείνει να γίνει μαύρο.

Έπειτα, εξετάστηκαν οι τυπικές τιμές του φίλτρου σε επτά διαφορετικές εικόνες, τέσσερις ασπρόμαυρες με γκρι αποχρώσεις και τρεις έγχρωμες. Σε όλες τις περιπτώσεις επιτυγχάνεται ο πρωταρχικός σκοπός της εφαρμογής του φίλτρου, που είναι η διόρθωση της ανομοιομορφης φωτεινότητας. Στο τελικό αποτέλεσμα οι αποχρώσεις της εικόνας σκουραίνουν καθιστώντας τις λεπτομέρειες και τα επιμέρους στοιχεία της εικόνας είναι πιο διακριτά και ευκρινή.

Η παρούσα εργασία μπορεί να αποτελέσει τη βάση για μια πιο διευρυμένη μελλοντική μελέτη σχετική με την επεξεργασία εικόνας και το ομομορφικό φιλτράρισμά. Η δημιουργία ενός αλγορίθμου που θα συγκρίνει το ομομορφικό φιλτράρισμα με διάφορα είδη φίλτρων όπως χωρικά, γραμμικά ή μη γραμμικά θα συνεισέφερε στην εξαγωγή συμπερασμάτων σχετικά με την εξειδί-

κευση του εκάστοτε φίλτρου, τα συγκεκριμένα πλεονεκτήματα που προσφέρει και συνεπώς στην κατάλληλη χρήση του ανάλογα με τον αρχικό σκοπό της επεξεργασίας ή την ιδιαιτερότητα της εικόνας που μορφοποιείται. Επίσης, μια κατεύθυνση επέκτασης της εργασίας μπορεί να στοχεύει στην εξέταση εναλλακτικών εργαλείων για την υλοποίηση του κώδικα, όπως η χρήση Matlab αντί για την υπολογιστική γλώσσα Python και τη συνεπαγόμενη σύγκρισή τους.

Κεφάλαιο 6

Παραρτήματα

6.1 Homomorphic Filtering Main Code

```
import sys
import cv2 as cv
import numpy as np
import easygui

# Custom Utilities module
import Utilities as utl

# dynamically ask for the image to open
# this is so the program works well on every OS
imgpath = utl.load_image()

# if image is grayscale continue, otherwise exit
# Cr, Cb values are gonna be used when we have a YCrCb image
# otherwise the value is 0
image, color, Cr, Cb = utl.read_img(imgpath)

# log the image
log_image = utl.log_transform(image)

# here we initialize the gui
utl.gui(log_image)

while(True):
    ghigh = cv.getTrackbarPos('ghigh (/10)', 'Homomorphic Filter') / 10
    glow = cv.getTrackbarPos('glow (/10)', 'Homomorphic Filter') / 10
    cc = cv.getTrackbarPos('cc (/10)', 'Homomorphic Filter') / 10
    d0 = cv.getTrackbarPos('d0', 'Homomorphic Filter')
    #this check is to avoid division by zero
    if d0 == 0:
        d0 = 1

    # fft image
    fft_image = np.fft.fft2(log_image)
```

```

# shift fft frequency
shift_image = np.fft.fftshift(fft_image)

# create filter
gauss, lp = utl.gaussian_hp(ghigh, glow, cc, d0, shift_image)

## apply filter on image
# using numpy's `multiply` we perform element-wise multiplication as
# the `.*` operator does in matlab
# so for example:
#
# a = np.array([[1,2],
#               [3,4]])
#
# b = np.array([[5,6],
#               [7,8]])
#
# np.multiply(a,b)
#
# Produces the following:
#
# array([[ 5, 12],
#        [21, 32]])
filtered_image = np.multiply(shift_image,gauss)

# reverse the frequency back to the corner
reverse_shift_image = np.fft.ifftshift(filtered_image)

# invert fourier
inverse_fft_image = np.real(np.fft.ifft2(reverse_shift_image, axes=(0,1)))

# exp image
exp_image = utl.exp(inverse_fft_image)

# normalize image back
img_homomorphic = utl.normalize(exp_image, color, Cr, Cb)

cv.imshow("Homomorphic Filter", img_homomorphic)
k = cv.waitKey(1)
if k == 27:
    save_path = easygui.filesavebox()
    cv.imwrite(save_path, img_homomorphic)
    cv.destroyAllWindows()
    break

```

6.2 Homomorphic Filtering Utilities

```

import cv2 as cv
import numpy as np

```

```

from PIL import Image
import easygui

def load_image():
    image_path = easygui.fileopenbox()
    return image_path

# check if the image is grayscale
def is_grey_scale(img_path):
    img = Image.open(img_path).convert('RGB')
    w, h = img.size
    for i in range(w):
        for j in range(h):
            r, g, b = img.getpixel((i,j))
            if r != g != b:
                return False
    return True

def read_img(imgpath):
    Cr = 0
    Cb = 0
    if(is_grey_scale(imgpath)):
        image = cv.imread(imgpath, 0)
        image = cv.normalize(image, None, alpha=0, beta=1,
                             norm_type=cv.NORM_MINMAX, dtype=cv.CV_32F)
        color = "gray"
    else:
        image = cv.imread(imgpath)
        image = cv.cvtColor(image, cv.COLOR_BGR2YCrCb)
        channels = cv.split(image)
        image = cv.normalize(channels[0], None, alpha=0, beta=1,
                             norm_type=cv.NORM_MINMAX, dtype=cv.CV_32F)
        Cr = channels[1]
        Cb = channels[2]
        color = "rgb"
    return image , color, Cr, Cb

# p.176 Gonzalez
# https://pythontic.com/image-processing/pillow/logarithmic%20transformation
def log_transform(image):
    log_image = np.log1p(image)
    return log_image

def gui(image):
    cv.namedWindow('Homomorphic Filter', cv.WINDOW_NORMAL)
    start_ghigh = 30 # slider start position
    max_ghigh = 200 # maximal slider position
    start_glow = 4 # slider start position
    max_glow = 100 # maximal slider position
    start_c = 50 # slider start position

```

```

max_c = 500 # maximal slider position
start_d0 = 20 # slider start
max_d0 = 200 # maximal slider position
cv.createTrackbar('ghigh (/10)', 'Homomorphic Filter',
                 start_ghigh, max_ghigh, (lambda a: None))
cv.createTrackbar('glow (/10)', 'Homomorphic Filter',
                 start_glow, max_glow, (lambda a: None))
cv.createTrackbar('cc (/10)', 'Homomorphic Filter',
                 start_c, max_c, (lambda a: None))
cv.createTrackbar('d0', 'Homomorphic Filter', start_d0,
                 max_d0, (lambda a: None))

# the filter we are using is a custom GHPF, DoG taken from here
#
# Homomorphic filtering based illumination normalization method
# for face recognition
# Chun-Nian Fan a,b
#, Fu-Yan Zhang a
def gaussian_lp(img, c, D0):
    P,Q = img.shape
    centerX = P/2
    centerY = Q/2
    H = np.zeros(img.shape)
    U, V = np.meshgrid(range(int(P)), range(int(Q)), sparse=False,
                       indexing='ij')
    Duv = (((U-centerX)**2+(V-centerY)**2)).astype(float)
    H = np.exp((-c*Duv/(D0**2)))
    return H

def gaussian_hp(gammah,gammal,c,D0,img):
    lp = gaussian_lp(img, c, D0)
    base = (gammah - gammal)*(1-lp) + gammal
    return base, lp

def exp(image):
    exp_image = np.expm1(image)
    return exp_image

def normalize(exp_image, color, Cr, Cb):
    if (color == "gray"):
        return cv.normalize(exp_image, None, alpha=0, beta=255,
                            norm_type=cv.NORM_MINMAX, dtype=cv.CV_8U)
    else:
        image = cv.normalize(exp_image, None, alpha=0, beta=255,
                             norm_type=cv.NORM_MINMAX, dtype=cv.CV_8U)
        image = cv.merge((image, Cr, Cb))
        return cv.cvtColor(image, cv.COLOR_YCrCb2BGR)

```

Βιβλιογραφία

- [Cha14] S. Characters. *Segmenting License Plate Characters*. Stack Overflow. 2014. URL: <https://stackoverflow.com/questions/24731810/segmenting-license-plate-characters>.
- [Che20] C. Chen. *Digital Image Processing using Fourier Transform in Python*. Medium. 2020. URL: <https://hicraigchen.medium.com/digital-image-processing-using-fourier-transform-in-python-bcb49424fd82>.
- [Doc22] Docs.opencv.org. *OpenCV documentation index*. 2022. URL: <https://docs.opencv.org/>.
- [Eas22] Easygui.sourceforge.net. *EasyGUI—easygui 0.97 (2014-12-20) documentation*. 2022. URL: <http://easygui.sourceforge.net/>.
- [Edi13] S. Edins. *Homomorphic filtering – part 1*. Mathworks. 2013. URL: https://blogs.mathworks.com/steve/2013/06/25/homomorphic-filtering-part-1/?doing_wp_cron=1640884670.4906480312347412109375.
- [FE03] Walker A. Fisher R. Perkins S. και Wolfart E. *Spatial Filters - Gaussian Smoothing*. Homepages.inf.ed.ac.uk. 2003. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>.
- [Fer19] E. Ferreira. *Introduction to Fourier Transform with openCV*. Medium. 2019. URL: <https://medium.com/@elvisdias/introduction-to-fourier-transform-with-opencv-922a79cddf36>.
- [FZ11] C. Fan και F. Zhang. *Homomorphic filtering based illumination normalization method for face recognition*. Pattern Recognition Letters, 32(10), pp.1468-1479. 2011.
- [GW18] R. Gonzalez και R. Woods. *Digital Image Processing*. 4th ed. New York: Pearson Education, Inc. Hoboken, pp.4-124, 365, 386-388, 409-412. 2018.
- [GW21] B. Girod και G. Wetzstein. *Digital Image Processing*. Imageprocessingplace.com. 2021. URL: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/Image%20Processing-Overview%20with%20Sample%20Applications.pdf.
- [Ima22] Imaging.cs.msu.ru. *Ringing analysis | Laboratory of Mathematical Methods of Image Processing*. Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. 2022. URL: <https://imaging.cs.msu.ru/en/research/ringing>.
- [Kan11] C. Kanchi. *Packaging a Python script on Linux into a Windows executable*. Stack Overflow. 2011. URL: <https://stackoverflow.com/questions/2950971/packaging-a-python-script-on-linux-into-a-windows-executable>.
- [Mag20] D. Magamedova. *TRENDS IN THE DEVELOPMENT OF SCIENCE AND EDUCATION*. OpenCV - Computer Vision Tool. 2020. URL: <https://opencv.org>.

- [Nam21] B. Namee. *Vernon's Machine Vision: Table of Contents*. Homepages.inf.ed.ac.uk. 2021. URL: <https://homepages.inf.ed.ac.uk/rbf/BOOKS/VERNON/>.
- [Nau21] R. Naushad. *Fourier Transform for Image Processing in Python from scratch | DataDrivenInvestor*. DataDrivenInvestor. 2021. URL: <https://www.datadriveninvestor.com/2020/10/23/fourier-transform-for-image-processing-in-python-from-scratch/>.
- [Num22] Numpy.org. *NumPy*. 2022. URL: <https://numpy.org/>.
- [Pyt21] Python.org. *Welcome to Python.org*. 2021. URL: <https://www.python.org/about/>.
- [Raj07] El-Benawy T. Alhazmi M. Rajab M. *Journal of X-Ray Science and Technology*. Application of frequency domain processing to X-ray radiographic images of welding defects, 5(3), pp.147-156. 2007.
- [Rao21] R. Rao. *Computer Vision*. Πανεπιστημιακές Διαλέξεις. CSE Vision Faculty, University of Washington Courses.cs.washington.edu. 2021. URL: <https://courses.cs.washington.edu/courses/cse455/09wi/Lects/>.
- [Sal12] Ibrahim H. Saleh S. *Mathematical Equations for Homomorphic Filtering in Frequency Domain: A Literature Survey*. International Conference on Information and Knowledge Management (ICIKM 2012), IACSIT Press, Singapore. 2012.
- [Smi07] J. Smith. *Mathematics of the Discrete Fourier Transform (Dft) with Audio Applications*. 2nd ed.: W3K Publishing. Ccrma.stanford.edu. 2007. URL: <https://ccrma.stanford.edu/~jos/mdft/>.
- [Sto72] T. Stockham. *Image Processing in the Context of a Visual Model*. Math.utah.edu. 1972. URL: https://www.math.utah.edu/~gustafso/s2014/3150/pdeNotes/Artigo_Stockham_01450712.pdf.
- [Tut22] Tutorialspoint.com. *High Pass vs Low Pass Filters*. 2022. URL: https://www.tutorialspoint.com/dip/high_pass_vs_low_pass_filters.htm.
- [YV21] Gerbrands J. Young I. και van Vliet L. *Fundamentals of Image Processing*. Imageprocessingplace.com. pp.2-12. 2021. URL: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/Image%20Processing%20Fundamentals--An%20overview.pdf.
- [Αγγ21] Ν. Αγγελιδάκης. *Εισαγωγή στον προγραμματισμό με την Python*. 1st ed. Ηράκλειο, Κρήτης, pp.12-16. 2021. URL: http://aggelid.mysch.gr/pythonbook/INTRODUCTION_TO_COMPUTER_PROGRAMMING_WITH_PYTHON.pdf.
- [Ελλ21] Ι. Έλληνας. *Επεξεργασία Εικόνας*. Πανεπιστημιακές Διαλέξεις. Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστικών Συστημάτων, Τεχνολογικός Τομέας, Α.Ε.Ι. Πειραιά. 2021. URL: <http://eclass.teipir.gr/opene/class/modules/units/?course=HYS107&id=145>.
- [Μπε15] Κ. Μπερμπερίδης. *Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας*. Πανεπιστημιακές Διαλέξεις. Σχολή Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Πατρών. 2015. URL: <https://eclass.upatras.gr/courses/CEID1112/>.
- [Στρ15] Χ. Στρουθόπουλος. *Ψηφιακή Επεξεργασία Εικόνας*. Πανεπιστημιακές Διαλέξεις. Τμήμα Μηχανικών Πληροφορικής, Σχολή Τεχνολογικών Εφαρμογών, Τ.Ε.Ι. Κεντρικής Μακεδονίας. 2015.