



Πολυτεχνική Σχολή
Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

Κατσαρός Ανδρέας

A.M. 1084522

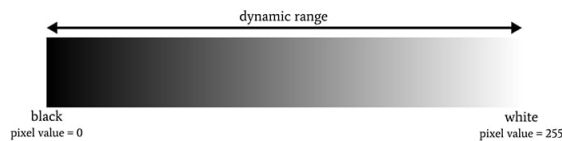
Πάτρα, 2024-25

ΕΡΩΤΗΜΑΤΑ

Σημείωση: Ο πλήρης κώδικας έχει αποσταλεί σε ξεχωριστά αρχεία τύπου .mlx (ex 1.mlx, ex 2.mlx, κ.λπ.), με σχόλια και αναλυτικές εξηγήσεις για κάθε ερώτημα. Κάθε αρχείο περιλαμβάνει αναλυτική παρουσίαση της λογικής και της υλοποίησης του κώδικα. Επιπλέον, η μορφοποίηση στα .mlx έχει γίνει με τρόπο ώστε κάθε ζητούμενο να παρουσιάζεται καθαρά και οπτικά οργανωμένα.

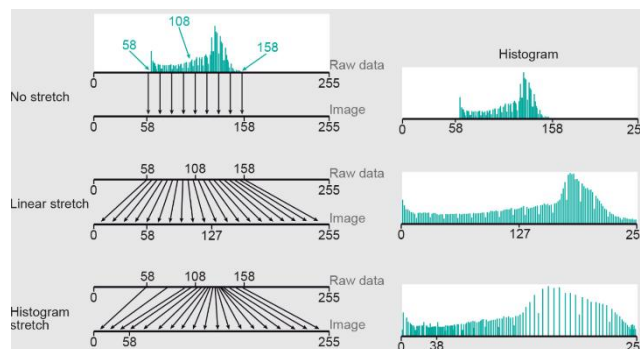
1) Προεπεξεργασία: Αρχικά εφαρμόστε γραμμικό μετασχηματισμό της περιοχής τιμών της εικόνας, ώστε να καλύπτει πλήρως τη δυναμική περιοχή [0:255]. Στη συνέχεια, με χρήση κατάλληλης ιδιότητας του μετασχηματισμού DFT, μεταφέρετε το συχνοτικό σημείο (0,0) στο κέντρο του πεδίου.

Στο στάδιο της προεπεξεργασίας επιδιώκουμε αρχικά να ανοίξουμε πλήρως τη γκάμα των τιμών της εικόνας, ώστε να αξιοποιήσουμε στο έπακρο ολόκληρο το δυναμικό εύρος [0, 255]. Αυτό το κάνουμε προκειμένου να αυξηθεί η οπτική διάκριση μεταξύ των features.



Στη φυσική της πλευρά, η ψηφιακή εικόνα απαρτίζεται από pixels που έχουν τιμές φωτεινότητας, συνήθως ακέραιες μεταξύ κάποιου ελάχιστου και μέγιστου. Αν οι τρέχουσες τιμές καταλαμβάνουν μόνο ένα υποσύνολο αυτού του διαστήματος, τότε η εικόνα φαίνεται υποτονική, με φτωχή αντίθεση.

Υπάρχουν διάφορες τεχνικές να επιτευχθεί αυτό το contrast enhancement (*Linear contrast stretch, histogram equalization.*)



Εδώ εφαρμόζουμε τον γραμμικό μετασχηματισμό (*linear stretching*) παίρνουμε κάθε τιμή της αρχικής εικόνας, την αναλογούμε σε ποσοστιαία κλίμακα σε σχέση με το εύρος [ελάχιστο, μέγιστο], και την επεκτείνουμε ώστε το ελάχιστο να γίνει 0 και το μέγιστο να γίνει 255:.

Στο στάδιο της προεπεξεργασίας επιδιώκουμε να επεκτείνουμε τη δυναμική περιοχή της εικόνας ώστε να καλύπτει πλήρως το διάστημα [0, 255], ενισχύοντας την αντίθεση και την ενίσχυση των λεπτομερειών (contrast enhancement).

Αρχικά, υπολογίζουμε τις ακραίες τιμές intensity φωτεινότητας της εισόδου,

$$I_{min} = \min_{x,y} I(x,y), \quad I_{max} = \max_{x,y} I(x,y)$$

και εφαρμόζουμε γραμμική κλιμάκωση (linear stretching):

$$I'(x,y) = \frac{I(x,y) - I_{min}}{I_{max} - I_{min}} \times 255.$$

Με αυτόν τον τρόπο ενισχύουμε την αντίθεση, τεντώνοντας τη συσχέτιση φωτεινότητας-pixel σε όλο το δυναμικό πεδίο [0, 255].

Όμως η επεξεργασία στο πεδίο των συχνοτήτων απαιτεί ένα ακόμα βήμα:

τη μετατόπιση της συνιστώσας DC στο κέντρο του φάσματος. Στην έξοδο ενός διακριτού μετασχηματισμού Fourier (DFT), αυτή η συνιστώσα εμφανίζεται στην κορυφή της γωνίας πάνω αριστερά του πίνακα του φάσματος.

Για λόγους οπτικοποίησης και ευκολότερου σχεδιασμού φίλτρων συχνοτήτων χαμηλής ή υψηλής διέλευσης, θέλουμε να μεταφέρουμε αυτή τη μέση τιμή ακριβώς στο κέντρο του πίνακα.

Το κεντράρισμα της μπορεί να επιτευχθεί με δύο τρόπους:

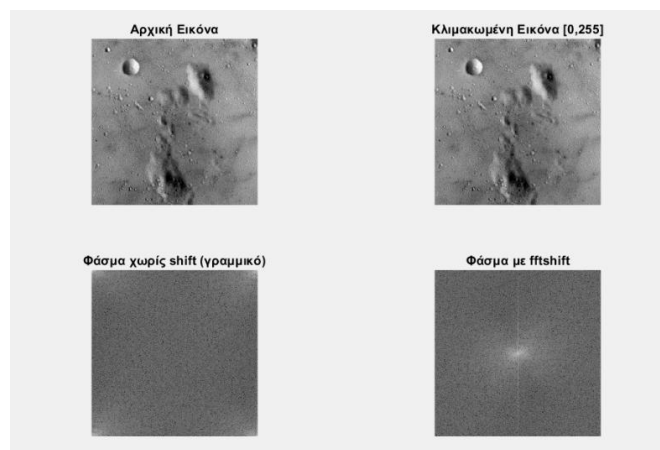
-DFT μετατόπιση στο χωρικό πεδίο, πολλαπλασιάζοντας κάθε pixel της εικόνας με $(-1)^{x+y}$. Αυτό προκαλεί εναλλαγή προσήμου με τέτοιο μοτίβο ώστε, όταν εφαρμόσουμε τον μετασχηματισμό Fourier, η μηδενική συχνότητα να βρεθεί στη μέση.

- Μετά-DFT μετατόπιση στο φάσμα, χρησιμοποιώντας απευθείας τη συνάρτηση matlab fftshift.

fftshift

Shift zero-frequency component to center of spectrum

Έτσι, διευκολύνουμε την οπτική και αλγοριθμική διάκριση χαμηλών και υψηλών συχνοτήτων.



Αποτελέσματα Προεπεξεργασίας:

- Η εικόνα πλέον εκτείνεται πλήρως σε [0–255] (πάνω δεξιά).
- Στο φάσμα χωρίς shift (κάτω αριστερά) η πηγή (DC) είναι στην πάνω-αριστερή γωνία.
- Με fftshift (κάτω δεξιά) η DC συνιστώσα βρίσκεται στο κέντρο, έτοιμη για εφαρμογή φίλτρων.

2) Να υλοποιηθεί ο δισδιάστατος διακριτός μετασχηματισμός Fourier με τη μέθοδο γραμμών-στηλών και χρησιμοποιώντας το μονοδιάστατο μετασχηματισμό DFT. Να γίνει γραμμική και λογαριθμική απεικόνιση του πλάτους του μετασχηματισμού της εικόνας.

Ο 2D-DFT είναι ουσιαστικά η γενίκευση του μονοδιάστατου μετασχηματισμού σε δύο διαστάσεις: ορίζει για κάθε συχνοτικό (u,v) πόση ενέργεια (πλάτος) υπάρχει στην εικόνα σε εκείνη τη συχνότητα οριζόντια και κάθετα.

Θεωρητικά, ορίζεται ως

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)},$$

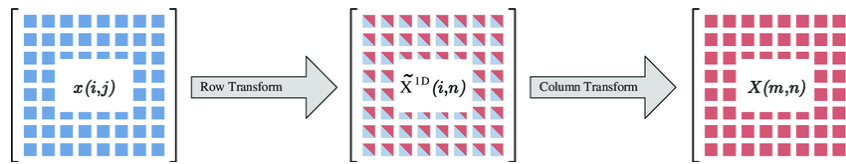
όπου $I(x, y)$ είναι η ένταση (intensity) στο pixel (x,y) και M,N οι διαστάσεις της εικόνας.

Η άμεση υλοποίηση αυτού του διπλού αθροίσματος μπορεί να είναι υπολογιστικά βαρεια για μεγάλα M,N.

Για αυτόν τον λόγο συχνά αξιοποιούμε την ιδιότητα αποσύνθεσης σε δύο μονοδιάστατους μετασχηματισμούς (μέθοδος γραμμών-στηλών):

- Πρώτα εφαρμόζουμε DFT σε καθέναν από τους M οριζόντιους (rows) της εικόνας σαν να ήταν ανεξάρτητες μονοδιάστατες ακολουθίες μήκους N.

- Παίρνουμε το αποτέλεσμα και εφαρμόζουμε DFT σε κάθε μία από τις N στήλες των μετασχηματισμένων σειρών.



Row-column decomposition of a 2-D transform.

Αφού αποκτήσουμε το σύνθετο αποτέλεσμα $F(u,v)$, το **μέτρο** του:

Μας δείχνει πόση ενέργεια υπάρχει σε κάθε συχνότητα και σε ποια περιοχή της εικόνας αντιστοιχεί. Για να το απεικονίσουμε προχωράμε σε δύο βασικούς τρόπους:

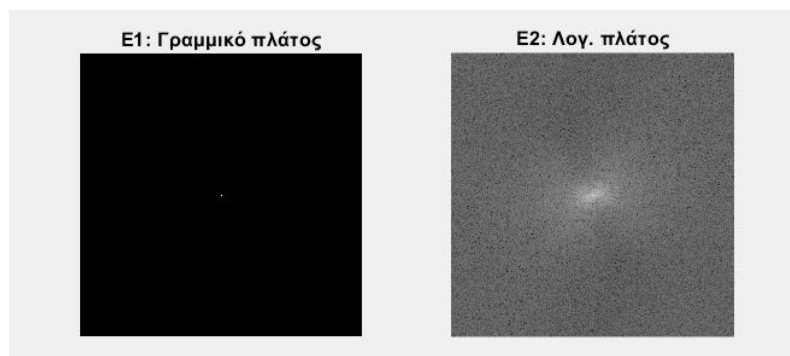
- **Γραμμική απεικόνιση:**

Υπολογίζουμε $|F(u,v)|$ και το κανονικοποιούμε (π.χ. μέσω κλιμάκωσης στο $[0,255]$ ή απλώς διαιρώντας με τη μέγιστη τιμή). Ωστόσο, επειδή οι τιμές πλάτους τείνουν να είναι πολύ μεγαλύτερες στις χαμηλές συχνότητες (κοντά στο κέντρο), οι υψηλές συχνότητες εμφανίζονται πολύ σκουρότερες και χάνουν λεπτομέρεια.

- **Λογαριθμική απεικόνιση:**

Για να αναδείξουμε και τις πιο μη φανερές, (υψηλές) συχνότητες, παίρνουμε $\log(1 + |F(u,v)|)$. Ο λογάριθμος συμπιέζει το δυναμικό εύρος του φάσματος, κάνοντας πιο ορατές τις λεπτομέρειες στις εξασθενημένες συχνότητες ενώ διατηρεί τις ισχυρότερες.

Οι δύο τρόποι απεικόνισης (linear & log) μας δίνουν διαφορετικές οπτικές γωνίες στο ίδιο φάσμα, αποκαλύπτοντας είτε τη γενική κατανομή ενέργειας είτε τις λεπτομέρειες στις υψηλές συχνότητες.



Σχολιασμός:

E1: Γραμμικό Πλάτος

Η απεικόνιση αυτή βασίζεται στην απλή κανονικοποίηση του πλάτους του μετασχηματισμού Fourier $|F(u,v)|$ στο εύρος $[0, 1]$. Καθώς στις περισσότερες εικόνες η φασματική ενέργεια συγκεντρώνεται έντονα στις χαμηλές συχνότητες, το κέντρο

του φάσματος (όπου βρίσκεται η DC συνιστώσα) εμφανίζεται ως σχεδόν μοναδική φωτεινή περιοχή. Αντίθετα, οι υψηλές συχνότητες, που έχουν κατά πολύ μικρότερα πλάτη, αποδίδονται με πολύ χαμηλή φωτεινότητα, καταλήγοντας πρακτικά μαύρες. Ως αποτέλεσμα, η απεικόνιση δεν αποτυπώνει ουσιαστικά καμία πληροφορία για τη δομή του φάσματος πέρα από την ύπαρξη της κύριας ενέργειας στο κέντρο.

E2: Λογαριθμικό Πλάτος

Με την εφαρμογή του λογαριθμικού μετασχηματισμού $\log(1+|F(u,v)|)$, το φάσμα συμπιέζεται δυναμικά, επιτρέποντας την αποκάλυψη λεπτομερειών που είναι αόρατες στη γραμμική απεικόνιση. Οι περιοχές χαμηλής ενέργειας (υψηλές συχνότητες) εμφανίζονται πλέον πιο ευδιάκριτες, με αποτέλεσμα να αποκαλύπτονται σταδιακοί κυκλικοί θόρυβοι, παλμοί ή γραμμικά μοτίβα. Η DC συνιστώσα στο κέντρο εξακολουθεί να είναι η πιο φωτεινή.

3) Να φιλτράρετε την εικόνα στο πεδίο συχνοτήτων (u,v) με χρήση ενός κατωπερατού φίλτρου (low pass filter) $H(u,v)$ με ζώνη διάβασης της επιλογής σας.

Έχουμε ήδη περάσει το σήμα στο χωρικό πεδίο και το έχουμε μεταφέρει στο φασματικό με τη DC στο κέντρο.

Τώρα θέλουμε να αφήσουμε να περάσουν μόνο οι χαμηλές συχνότητες, δηλαδή τις ομαλές διαβαθμίσεις της εικόνας και να απορρίψουμε τις υψηλότερες συχνότητες, που συνήθως αντιστοιχούν σε θόρυβο ή πολύ έντονες ακμές.

Για να καθαρίσουμε την εικόνα από τις πολύ γρήγορες μεταβολές φωτεινότητας (που συνήθως αντιστοιχούν σε θόρυβο ή ανεπιθύμητες λεπτομέρειες, ερώτημα 2), ορίζουμε ένα **κατωπερατό φίλτρο** $H(u,v)$. Το H είναι ένας πίνακας ίδιων διαστάσεων με το φάσμα, όπου επιλέγουμε μια **ζώνη διάβασης** και βάζουμε:

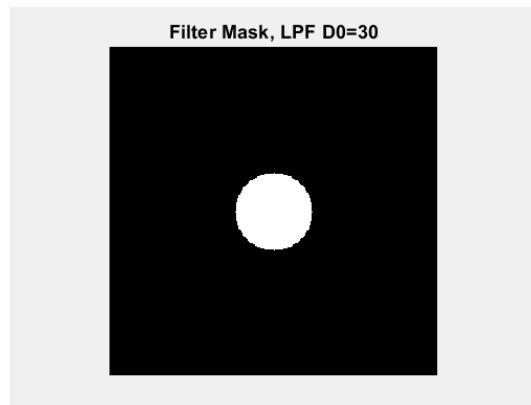
- $H(u,v)=1$ για όλες τις συχνότητες που βρίσκονται **εντός** αυτής της ζώνης (δηλαδή οι χαμηλές συχνότητες που θέλουμε να περάσουν ανεμπόδιστα),
- $H(u,v)=0$ για όλες τις συχνότητες **εκτός** της ζώνης (όπου οι υψηλές συχνότητες αποκόπτονται).

Με αυτόν τον τρόπο, στη πράξη πολλαπλασιάζουμε απλώς το κεντραρισμένο φάσμα:

$$F_{\text{filtered}}(u,v) = H(u,v) [F_{\text{shifted}}(u,v)]$$

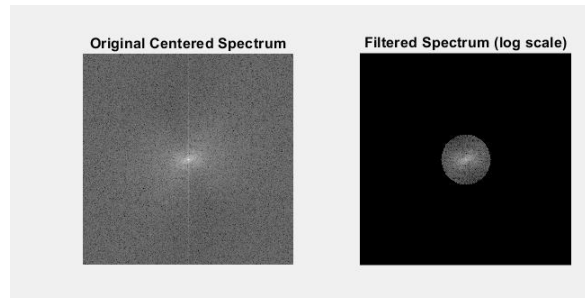
Έχοντας σβήσει όλες τις ανεπιθύμητες υψηλές συχνότητες εκτός του επιλεγμένου κύκλου, διατηρώντας μόνο τις χαμηλές, ομαλές μεταβολές της εικόνας.

- Ορίζουμε ένα **κατωπερατό φίλτρο** $H(u,v)$, δηλαδή έναν πίνακα διαστάσεων $M \times N$ (ίσων με το φάσμα), όπου κρατάμε ποιες συχνότητες θέλουμε να περάσουν (τιμή 1) και ποιες να κόψουμε (τιμή 0).
- Επιλέγουμε μια **ζώνη διάβασης** γύρω από το κέντρο — συνήθως έναν κύκλο ακτίνας $D0=30$.



Επιλεγμένο φίλτρο

Το φίλτρο έχει εφαρμοστεί σωστά στο πεδίο των συχνοτήτων, σβήνοντας όλες τις ανεπιθύμητες υψηλές συνιστώσες:



- Το subplot με τίτλο **Original Centered Spectrum** είναι απλώς η απεικόνιση του κεντραρισμένου φάσματος **πριν** το φιλτράρισμα (βήμα 1.2/1.2-visualization).
- Το subplot με τίτλο **Filtered Spectrum (log scale)** είναι η έξοδος του **βήματος 1.3**, δηλαδή το φάσμα **μετά** την εφαρμογή του κατωπερατού φίλτρου $H(u,v)$. Μετά την εφαρμογή του κυκλικού LPF $D0=30$, το φάσμα κόβεται έξω από τον κεντρικό κύκλο. Το εσωτερικό του κύκλου παραμένει φωτεινό, αφού οι χαμηλές συχνότητες περνούν ανεμπόδιστα, ενώ το υπόλοιπο φάσμα εκτός κύκλου έχει γίνει εντελώς μαύρο (μηδενικές συνιστώσες). Η λογαριθμική απεικόνιση αναδεικνύει καθαρά το όριο της ζώνης διάβασης και πόσο προστατεύονται οι συχνότητες μέσα στην ακτίνα $D0$.

4) Να εφαρμοστεί ο δισδιάστατος αντίστροφος διακριτός μετασχηματισμός IDFT για την επαναφορά στο χωρικό πεδίο.

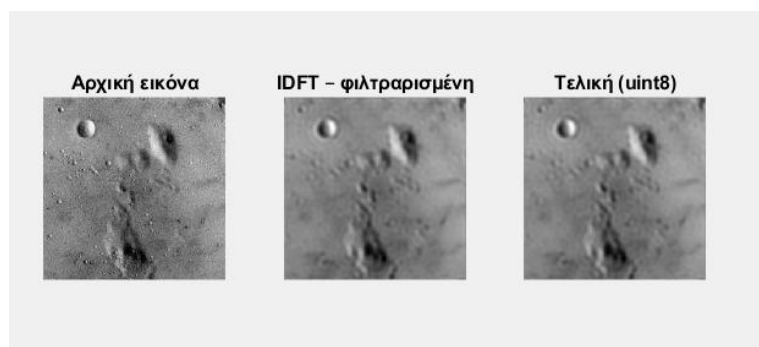
Μετά, αρκεί να εφαρμόσουμε αντίστροφο μετασχηματισμό Fourier (IFFT) — αφού πρώτα επαναφέρουμε το DC πίσω στην αρχική του θέση με `ifftshift` — για να δούμε στην εικόνα πώς φιλτράρισε η διαδικασία.

Διαδικασία κώδικα:

- **Ορισμός φίλτρου:** Επιλέγουμε το είδος και το μέγεθος της ζώνης (εμείς), καθορίζοντας την ακτίνα ή τα όρια πέρας της διέλευσης.
- **Εφαρμογή στο φάσμα:** Πολλαπλασιάζουμε το κεντραρισμένο φάσμα με το μάσκα $H(u,v)$.
- **Αντίστροφη μετατόπιση & IFFT:** Χρησιμοποιούμε τον `ifft2` (ή αναλυτικά `ifft` σειρές-στήλες) για να συναρμολογήσουμε ξανά την εικόνα. Ο IDFT παίρνει όλες τις φασματικές συνιστώσες, τόσο τις χαμηλές που αφήσαμε να περάσουν, όσο και τις μηδενισμένες, και τις συνδυάζει σε μία δισδιάστατη σύνθεση:

$$I(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F_{\text{filtered}}(u, v) e^{+j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)},$$

- **Αφαίρεση φανταστικών τμημάτων (real)**
Επειδή οι αριθμητικοί υπολογισμοί μπορεί να εισάγουν μικρές φανταστικές (imaginary) τιμές, παίρνουμε το πραγματικό μέρος της σύνθετης εξόδου με `real(...)`.
- **Αποτελέσματα και plots**



Με την επιλογή ακτίνας ζώνης $D0=30$ και IDFT , η φιλτραρισμένη εικόνα δείχνει, στο “Μετά IDFT”:

- **Μείωση θορύβου υψηλών συχνοτήτων:**

Οι κόκκοι και οι τυχαίες διακυμάνσεις φωτεινότητας στην επιφάνεια της σελήνης έχουν γίνει αισθητά πιο ομαλές, καθώς το φίλτρο έχει αποκόψει τις συχνότητες που αντιστοιχούν σε μικρές, αιχμηρές παραλλαγές.

- **Απώλεια λεπτομερειών:**

Κρατήθηκαν κρατήσεις, λόφοι, και το μεγάλο ανάγλυφο, αλλά οι μικρότερες υφές (όπως λεπτομερείς γραμμώσεις ή πολύ μικροί κρατήρες) έχουν σβηστεί ή θολώσει. Αυτό είναι αναμενόμενο, καθώς αποκόψαμε όλες τις συχνότητες πάνω από το κατώφλι $D0$.

- **Εξομάλυνση ακμών:**

Οι ακμές μεταξύ σκοτεινών και φωτεινών περιοχών είναι πλέον πιο απαλές. Αν και δεν χάνουμε τον συνολικό σκελετό της εικόνας, οι μεταβάσεις φωτεινότητας είναι λιγότερο απότομες.

- **Ισορροπία χαμηλών–υψηλών:**

Η εικόνα διατηρεί το γενικό κοντράστ και τις μεγάλες διαφορές φωτεινότητας (χαμηλές συχνότητες), ενώ καθαρίζει τον ανεπιθύμητο θόρυβο. Αν μειώσουμε περισσότερο την ακτίνα $D0$, θα δούμε ακόμα πιο έντονο θόλωμα· αν την αυξήσουμε, θα διατηρηθούν περισσότερες λεπτομέρειες.

Συνοψίζοντας, το κατωπερατό φίλτρο επιτυγχάνει: εξομάλυνση και μείωση θορύβου, με αναπόφευκτη , αλλά ελεγχόμενη απώλεια των πολύ λεπτών υφών.

5) Τέλος, εφαρμόστε την αντίστροφη διαδικασία του Βήματος 1, έτσι ώστε να επαναφέρετε το σημείο (0,0).

Στο βήμα 1.5 ολοκληρώνουμε τον κύκλο Fourier επιστρέφοντας ξανά στο χωρικό πεδίο με δύο κρίσιμες ενέργειες:

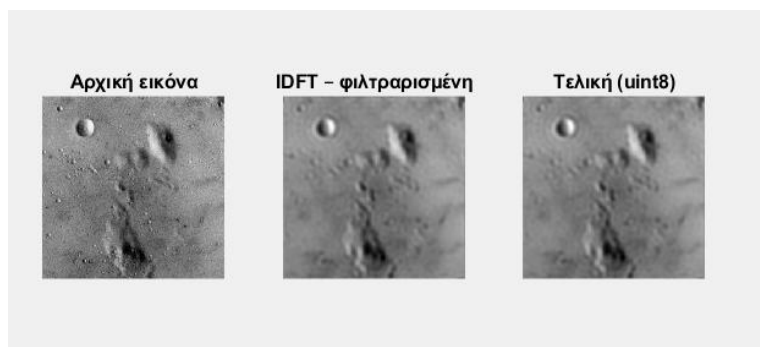
1. **Επαναφορά της συνιστώσας DC**

Κατά τη μεταφορά στο πεδίο των συχνοτήτων είχαμε κεντράρει το σημείο (0,0) στο κέντρο του φάσματος (με `fftshift`), διευκολύνοντας το φιλτράρισμα. Τώρα, πριν προχωρήσουμε σε τελική απεικόνιση, πρέπει να ξε-κεντράρουμε το φάσμα με `ifftshift` ή, αντιστοίχως, να πολλαπλασιάσουμε ξανά στο χωρικό πεδίο με $(-1)^{x+y}$. Έτσι η μέση τιμή φωτεινότητας (DC) επιστρέφει στην πάνω-αριστερή γωνία, όπως απαιτεί ο ορισμός του IDFT.

2. Τελική κλιμάκωση στο [0...255]

Μετά τον αντίστροφο μετασχηματισμό το αποτέλεσμα είναι ένας πίνακας πραγματικών τιμών φωτεινότητας. Στο τέλος μετατρέπουμε σε uint8, εξασφαλίζοντας ότι κάθε pixel έχει έγκυρη τιμή [0,255].

Με αυτές τις δύο κινήσεις, το φιλτραρισμένο σήμα αποκαθίσταται πλήρως ως εικόνα: η διακριτή συχνότητα μηδέν βρίσκεται ξανά στη θέση της, και οι τιμές φωτεινότητας εκτείνονται ομοιόμορφα στο πλήρες δυναμικό πεδίο.



- Το “Μετά IDFT” επιδεικνύει το ωμό, μη-κλιμακωμένο αποτέλεσμα του IDFT (χωρίς έλεγχο εύρους, 1.4), το φασματικό θόλωμα των υψηλών συχνοτήτων οδηγεί σε αισθητά πιο λεία υφή στην εικόνα και σε μείωση του υψηλής συχνότητας θορύβου. Οι έντονες ακμές και οι πολύ μικρές λεπτομέρειες έχουν μαλακώσει ή εξαλειφθεί, επιτυγχάνοντας την επιθυμητή εξομάλυνση. ενώ
- Το “Τελική με Re-scaling & DC” δείχνει την κανονικοποιημένη, τελική εικόνα που θα χρησιμοποιήσεις, με πλήρη αντίθεση και σωστή τοποθέτηση της συνιστώσας DC. Πριν την οπτική παρουσίαση και χρήση, η εικόνα μετατρέπεται σε ακέραιο τύπο (uint8), ακολουθώντας γραμμική ανακλιμάκωση ώστε οι φωτεινότητες να εκτείνονται στο εύρος [0,255]. Επιπλέον, εφαρμόζεται η αντίστροφη διαδικασία μετακίνησης της DC συνιστώσας (από το κέντρο πίσω στην πάνω αριστερή γωνία), ώστε η εικόνα να ευθυγραμμίζεται με τη φυσική διάταξη του πεδίου. Η τελική μορφή αποτυπώνει τις επιδράσεις της φασματικής επεξεργασίας με μέγιστη δυνατή ακρίβεια και οπτική σαφήνεια.
- Η εικόνα έχει πλέον μαζευτεί ώστε το σκοτεινότερο pixel να είναι αληθινό μαύρο (0) και το φωτεινότερο αληθινό άσπρο (255).
- Οι τόνοι είναι ομοιόμορφα διαδοχικοί, χωρίς clipping – όλη η δυναμική κλίμακα των πιξελ αξιοποιείται.

Στον κώδικα

```
I_final = I_filtered - min(I_filtered(:));  
I_final = I_final ./ max(I_final(:)) * 255;  
I_final_uint8 = uint8(I_final);
```

Σε αυτό το block υπολογίζουμε ρητά το νέο ελάχιστο και μέγιστο, εφαρμόζουμε το γραμμικό scaling, και μετά μετατρέπουμε σε uint8, εξασφαλίζοντας πλήρη χρήση του [0,255] χωρίς clipping.

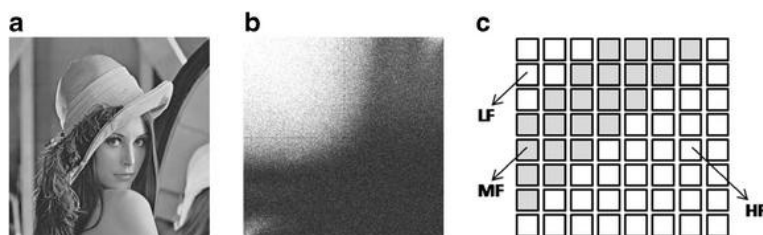
Ο Πλήρης σχολιασμένος κώδικας Ζητούμενης Επεξεργασίας 1 βρίσκεται στο τέλος της αναφοράς.

2. Συμπίεση Εικόνας με χρήση μετασχηματισμού DCT

Απεικονίστε το μέσο τετραγωνικό σφάλμα ανάμεσα στην αρχική και την ανακατασκευασμένη εικόνα για τις τιμές του x που ανήκουν στο σύνολο τιμών [5%, 50%].

Εδώ , επιδιώκουμε **συμπίεση** μέσω του μετασχηματισμού **DCT**, ο οποίος έχει την ιδιότητα να συγκεντρώνει την ενέργεια (πληροφορία) των περισσότερων εικόνων στους συντελεστές χαμηλών συχνοτήτων.

Καταρχάς, διαιρούμε την έγχρωμη (ή γκρι) εικόνα **board.png** σε **μη επικαλυπτόμενα blocks 32×32**. . Κάθε block θεωρείται ανεξάρτητη υποεικόνα.



Σε κάθε block εφαρμόζουμε τον **δισδιάστατο DCT** (2D-DCT), ο οποίος για κάθε block B 32×32 υπολογίζει:

$$C_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{31} \sum_{y=0}^{31} B_{x,y} \cos \left[\frac{\pi(2x+1)u}{2 \cdot 32} \right] \cos \left[\frac{\pi(2y+1)v}{2 \cdot 32} \right]$$

όπου $u, v = 0, \dots, 31$, και οι παράμετροι $\alpha(\cdot)$ είναι οι συντελεστές κλιμάκωσης του DCT.

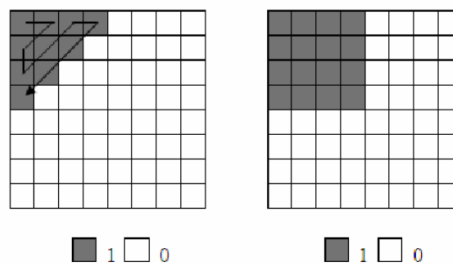
(στον κώδικα είναι embedded στην matlab ως command)

Μέθοδος Ζώνης

- Ο **σχεδιασμός της ζώνης**: μια απεικόνιση του block 32×32 με θέσεις μέσα στη ζώνη είναι επιλεγμένες (τιμή 1) και οι υπόλοιπες μηδενισμένες (τιμή 0).
- **Ανασύνθεση** κάθε block με τους περιορισμένους συντελεστές: εφαρμόζουμε το αντίστροφο DCT (IDCT) μόνο στις θέσεις που κρατήσαμε, για να δούμε την παραμορφωμένη εκδοχή του block.
- Την **πραγματική συμπίεση**: συγκρίνουμε το μέγεθος των δεδομένων πριν/μετά (αριθμός μη-μηδενικών συντελεστών), καθώς και το οπτικό αποτέλεσμα σε ολόκληρη την εικόνα όταν επανατοποθετούμε τα block.

Επιλογή ποσοστού r

Για κάθε block αποφασίζουμε να διατηρήσουμε ένα ποσοστό $r\%$ των συνολικών 32×32 συντελεστών. Με τη ζώνη επιλέγουμε τους πρώτους $\lceil r/100 \cdot 1024 \rceil$ συντελεστές που βρίσκονται εντός της καθορισμένης ζώνης, αφήνοντας εκτός (μηδενίζοντας) όλους τους υπόλοιπους.



Παράδειγμα διαφοράς Zonal Mask vs Zigzag mask method

Μέθοδο κατωφλίου:

Εδώ, αντί να κρατάμε πάντα ένα προκαθορισμένο σχήμα γύρω από τις χαμηλές συχνότητες, εντοπίζουμε **ποιοι DCT συντελεστές έχουν τη μεγαλύτερη απόλυτη τιμή** και διατηρούμε μόνο αυτούς μέχρι να καλύψουμε το επιθυμητό ποσοστό $r\%$ της συνολικής ενέργειας ή του πλήθους των συντελεστών.

Έτσι, αντί για στατικές ζώνες, χρησιμοποιούμε ένα ποιοτικό κριτήριο για να αποφασίσουμε ποιοι συντελεστές μετρούν πράγματι, επιτυγχάνοντας συμπίεση χωρίς άσκοπη απώλεια πληροφορίας.

Μελέτη μεθόδου Paper : DCT Image Denoising: a Simple and Effective Image Denoising Algorithm

2 Algorithm

2.1 Threshold Estimation and Sparse Signal Representation

A signal $\mathbf{f} \in \mathbb{R}^N$ is contaminated by a noise $\mathbf{w} \in \mathbb{R}^N$ that is often modeled as a zero-mean Gaussian process independent of \mathbf{f}

$$\mathbf{y} = \mathbf{f} + \mathbf{w},$$

where $\mathbf{y} \in \mathbb{R}^N$ is the observed noisy signal. Signal denoising aims at estimating \mathbf{f} from \mathbf{y} .

Let $B = \{\phi_n\}_{1 \leq n \leq N}$ be an orthonormal basis, whose vectors $\phi_n \in \mathbb{C}^N$ satisfy

$$\langle \phi_m, \phi_n \rangle = \begin{cases} 1, & \text{if } n = m \\ 0, & \text{otherwise.} \end{cases}$$

A thresholding estimator projects the noisy signal to the basis, and reconstructs the denoised signal with the transform coefficients larger than the threshold T

$$\tilde{\mathbf{f}} = \sum_{n=1}^N \rho_T(\langle \mathbf{y}, \phi_n \rangle) \phi_n,$$

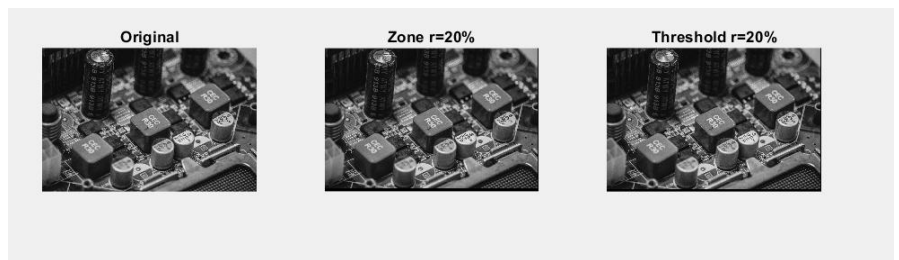
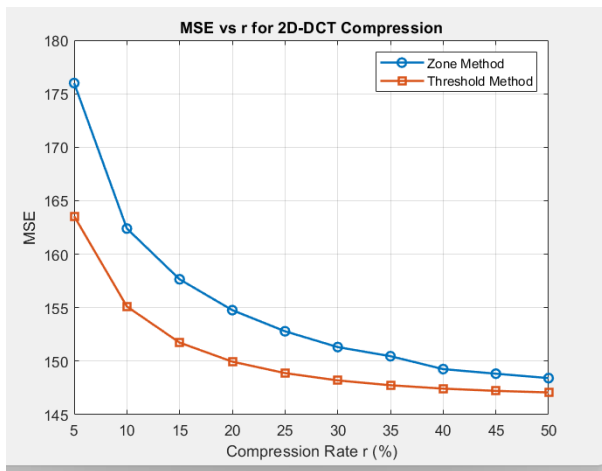
where

$$\rho_T(x) = \begin{cases} x, & \text{if } |x| > T \\ 0, & \text{otherwise,} \end{cases}$$

is a thresholding operator.

Κρατάμε μόνο τα "σημαντικά" coefficients (με μέγεθος μεγαλύτερο από T), και μηδενίζουμε τα υπόλοιπα — που πιθανόν είναι κυρίως θόρυβος.

Υλοποίηση: Zone $r=20\%$ και Threshold $r=20\%$.



2.1 Μέθοδος Ζώνης

Συνοπτική Περιγραφή Κώδικα

- Για κάθε block 32×32 διατηρήσαμε τους πρώτους K συντελεστές χαμηλών συχνοτήτων, σχηματίζοντας ένα ορθογώνιο τετράγωνο πλευράς $\lfloor \sqrt{K} \rfloor$ και, εάν χρειάστηκε, συμπληρώνοντας επιπλέον $K - \lfloor \sqrt{K} \rfloor^2$ συντελεστές στην επόμενη γραμμή.
- Όλοι οι υπόλοιποι συντελεστές μηδενίστηκαν.

–Υπάρχουν εκτενή σχόλια στον κώδικα για τη παραπάνω διαδικασία

Αποτελέσματα

1. MSE vs r

- Η καμπύλη Zone Method στο διάγραμμα MSE vs r (μπλε κύκλοι) ξεκινά από σχετικά υψηλό σφάλμα (περίπου 175) για $r=5\%$ και πέφτει σταδιακά (περίπου 150) για $r=50\%$.
- Η πορεία της καμπύλης είναι : στο χαμηλό r ($5 \rightarrow 15\%$) το MSE μειώνεται πιο απότομα, ενώ στα $r > 30\%$ η βελτίωση σβήνει, καθώς οι επιπλέον συντελεστές είναι όλο και πιο υψηλής συχνότητας, άρα μικρή συμβολή στην ανασύνθεση.

- Η ανακατασκευασμένη εικόνα εμφανίζει σημαντική απώλεια λεπτής υφής, ειδικά στις περιοχές με υψηλή λεπτομέρεια λόγω της γεωμετρικά περιορισμένης ζώνης που παραλείπει συντελεστές εκτός σχήματος.

2. Δείγμα ανασύνθεσης ($r=20\%$)

- Η εικόνα “Zone $r=20\%$ ” εμφανίζει ομαλές περιοχές με θόλωση στις λεπτές δομές (π.χ. οι ακμές των πυκνωτών και οι λεπτές γραμμές στο board), καθώς όλες οι συχνότητες εκτός της ζώνης έχουν κοπεί.

2.2 Μέθοδος Κατωφλίου

Συνοπτική Περιγραφή Κώδικα

- Σε κάθε block υπολογίζουμε το σύνολο των συντελεστών C .
- Ορίζουμε ένα κατώφλι τ ίσο με την K -οστή μεγαλύτερη απόλυτη τιμή $|C|$, έτσι ώστε ακριβώς K συντελεστές να υπερβαίνουν το τ .
- Κρατάμε αυτούς τους K συντελεστές, μηδενίζοντας όλους τους υπόλοιπους.
- Υπάρχουν εκτενή σχόλια στον κώδικα για τη παραπάνω διαδικασία

Αποτελέσματα

1. MSE vs r

- Η καμπύλη “Threshold Method” ξεκινά χαμηλότερα (~ 164 για $r=5\%$) και πέφτει πιο ομαλά, φτάνοντας ~ 147 για $r=50\%$.
- Η μείωση του MSE έχει πιο ήπια κλίση, αφού το threshold επιλέγει πάντα τους ισχυρότερους συντελεστές κάθε block.

2. Δείγμα ανασύνθεσης ($r=20\%$)

- Η εικόνα “Threshold $r=20\%$ ” εμφανίζει σημαντικά περισσότερες λεπτομέρειες σε σχέση με τη ζώνη: οι γραμμές και τα μικρά ηλεκτρονικά στοιχεία διατηρούνται καθαρότερα, καθώς το threshold διαλέγει τη δική του, δυναμικά ορισμένη, ζώνη συντελεστών.

Συγκριτική Ανάλυση

- **Ποιότητα ανασύνθεσης:** για κάθε r , η μέθοδος κατωφλίου δίνει χαμηλότερο MSE και πιο πιστό οπτικό αποτέλεσμα.
- **Όγκος μηδενισμένων συντελεστών:** και οι δύο μέθοδοι μηδενίζουν ακριβώς $(100-r)\%$ των συντελεστών, αλλά το πώς επιλέγονται αυτοί διαφέρει—γεωμετρικά (ζώνη) vs. ποσοτικά (threshold).

Τα παραπάνω επιβεβαιώνονται πλήρως από το διάγραμμα MSE vs r και τις τρεις εικόνες αρχικής, Zone r=20% και Threshold r=20%. Το threshold αποδεικνύεται καλύτερη επιλογή για την ίδια ποσότητα πληροφορίας.

3. Βελτίωση εικόνας - Φιλτράρισμα Θορύβου

Εδώ: Η βασική ιδέα είναι ότι διαφορετικά είδη θορύβου—κανονικός λευκός (Gaussian) και κρουστικός (salt-and-pepper)—αντιμετωπίζονται καλύτερα από διαφορετικές τάξεις φίλτρων.

– **Κινούμενου μέσου** (moving average) φίλτρο:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

Το φίλτρο αυτό είναι ένα γραμμικό, χωρικό φίλτρο το οποίο αντικαθιστά κάθε pixel της εικόνας με τον μέσο όρο των τιμών ενός παραθύρου γειτονικών pixel γύρω από αυτό.

Επιλέγουμε ένα τετράγωνο παράθυρο $m \times n$ (π.χ. 3×3 ή 5×5), και υπολογίζουμε (για 2-d):

$$I_{\text{out}}(x, y) = \frac{1}{mn} \sum_{i=-a}^a \sum_{j=-b}^b I_{\text{in}}(x + i, y + j)$$

όπου $a=(m-1)/2$, $b=(n-1)/2$. Με αυτόν τον τρόπο επιτυγχάνουμε **εξομάλυνση** των τυχαίων διακυμάνσεων (π.χ. Gaussian θόρυβος), καθώς οι θετικές και αρνητικές αποκλίσεις αλληλοαναιρούνται.

– **Φίλτρο διαμέσου** (median):

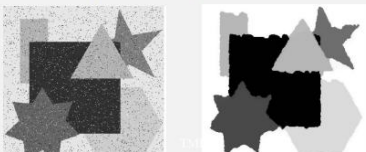
$$\text{MedianFilter} = \frac{a(\frac{N}{2}) + a(\frac{N}{2} + 1)}{2} \quad (1)$$

Το φίλτρο διαμέσου είναι μη γραμμικό και αντικαθιστά κάθε pixel με τη **διάμεσο** των τιμών στο γειτονικό παράθυρο $m \times n$.

Συλλέγουμε τις mn τιμές των γειτόνων, τις ταξινομούμε και επιλέγουμε τη μεσαία τιμή ως έξοδο.

■ $g(x, y) = \text{median} \{f(z, w)\}, (z, w) \in S_{x,y}$ median{4,3,5,8,2,6}=4.5

- Very good performance dealing with s&p noise
- Retains edges
- It may cut off peaks and sharp corners

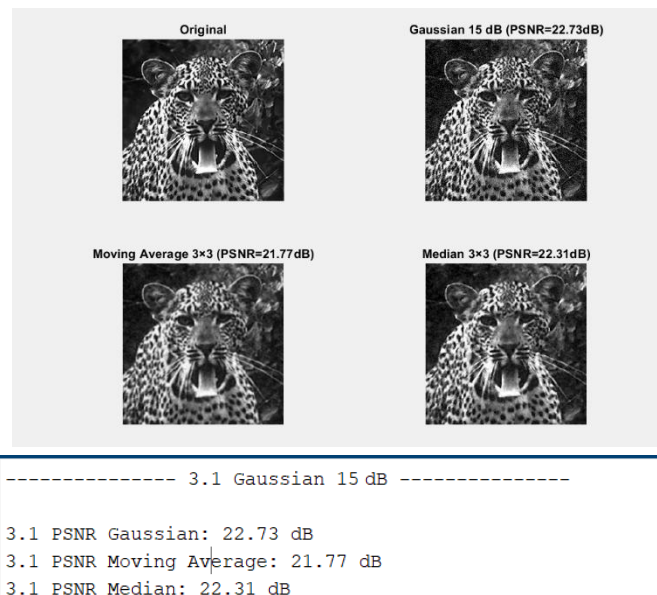


1. Στην εικόνα tiger.mat προσθέστε λευκό Gaussian θόρυβο μηδενικής μέσης τιμής και τέτοιας διασποράς ώστε η τελική εικόνα να έχει λόγο σήματος προς θόρυβο ίσο με 15dB. Χρησιμοποιείτε το φίλτρο 2 “κινούμενου μέσου” (moving average filter) και το “φίλτρο διαμέσου” (median filter) για να αφαιρέσετε το θόρυβο από την εικόνα.

Ο Gaussian θόρυβος έχει κατανομή με μηδενική μέση τιμή και ελεγχόμενη διασπορά, οπότε προσθέτει παντού μικρές, τυχαίες διακυμάνσεις της έντασης.

Για να επιτύχουμε $SNR = 15$ dB υπολογίζουμε τη δύναμη του σήματος (μέσο τετραγωνικό των pixel) και διαλέγουμε τη διασπορά του θορύβου ώστε

$$SNR_{dB} = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) = 15 \text{ dB}$$



Προστίθεται λευκός Gaussian θόρυβος $SNR=15$ dB και φιλτράρεται με **moving average** και **median** φίλτρα, παρατηρούμε τα εξής:

1. Φίλτρο κινούμενου μέσου (Moving Average)

Εφαρμόζοντας ένα γραμμικό φίλτρο 3×3, κάθε pixel αντικαθίσταται από τον τοπικό μέσο όρο των 9 γειτονικών τιμών. Αυτό οδηγεί σε σημαντική μείωση της τυχαίας διασποράς άρα σε μείωση του θορύβου, αλλά ταυτόχρονα επιφέρει θόλωμα: οι ακμές γίνονται πιο αμβλείες, καλύπτοντας λεπτές υφές και λεπτομέρειες στην πυκνωμένη γούνα και τα χαρακτηριστικά του τίγρη.

Γι αυτό : Η PSNR πέφτει σε ~21.8 dB, ελαφρώς χαμηλότερα από το αρχικό noisy (~22.7 dB), αντανakλώντας την απώλεια.

2. Φίλτρο διαμέσου (Median)

Σε αντίθεση, το μη-γραμμικό median φίλτρο αντικαθιστά κάθε τιμή με τη διάμεσο της γειτονιάς. Αυτό **διατηρεί καλύτερα τις ακμές**, αφού επιλέγει τιμές από το υπάρχον σετ. Στην πράξη, το median μειώνει τον Gaussian θόρυβο σχεδόν τόσο αποτελεσματικά όσο το moving average, αλλά με λιγότερο θόλωμα.

Η PSNR φτάνει ~22.3 dB, παραμένοντας πολύ κοντά ή ελαφρώς υψηλότερα από το noisy, δείχνοντας ότι διατηρεί μεγαλύτερο μέρος της πληροφορίας.

3. Καταλληλότητα

- **Gaussian θόρυβος** αντιμετωπίζεται ικανοποιητικά και από τα δύο φίλτρα, αλλά το median συστήνεται όταν θέλουμε να προστατεύσουμε ακμές και υφές.
- Το moving average ως αποτέλεσμα δίνει πιο ομαλό αποτέλεσμα, όμως η θόλωση μπορεί να είναι ανεπιθύμητη σε εφαρμογές όπου η λεπτομέρεια είναι κρίσιμη.

-Η επιλογή κάποιου άλλου φίλτρου για τέτοιου είδους εφαρμογές θα ήταν καταλληλότερη (wiener, gaussian blur...).

-Ακόμα και με επιλογή εναλλακτικού μπλοκ φίλτρου (5 x 5 , 8 x 8 ...) τα αποτελέσματα παραμένουν αντίστοιχα.

2. Στην εικόνα tiger.mat προσθέστε κρουστικό θόρυβο σε ποσοστό 20% και χρησιμοποιείτε τα ίδια φίλτρα για την αφαίρεση του θορύβου.

Στο υποερώτημα **3.2 (Salt & Pepper 20 %)** παρατηρούμε καθαρά την αδυναμία του γραμμικού φίλτρου moving average να αντιμετωπίσει ακραίες τιμές που επιφέρει το salt & pepper , σε αντίθεση με το median φίλτρο:

Κρουστικός θόρυβος Salt & Pepper

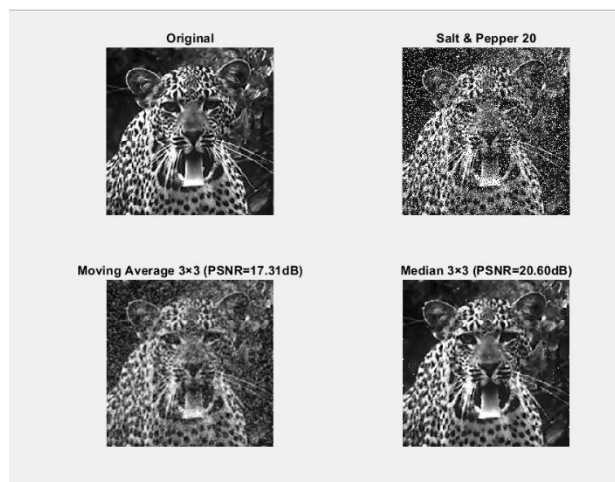
Εισαγάγαμε τυχαία σε ποσοστό 20 % των pixels είτε την τιμή 0 (pepper) είτε την τιμή 1 (salt). Το αποτέλεσμα είναι μια εικόνα γεμάτη από ακραίους λευκούς και μαύρους παλμούς διασκορπισμένους ομοιόμορφα.

----- 3.2 Salt & Pepper 20 % -----

3.2 PSNR Salt & Pepper: 11.60 dB

3.2 PSNR Moving Average: 17.31 dB

3.2 PSNR Median: 20.60 dB



Moving Average (PSNR \approx 17.3 dB)

Όταν εφαρμόζουμε το φίλτρο μέσου όρου 3×3 , κάθε pixel αντικαθίσταται από τον απλό μέσο όρο των 9 γειτονικών τιμών. Αυτό λειτουργεί καλά για Gaussian θόρυβο, αλλά εδώ οι ακραίες τιμές των παλμών μετακινούνται στα γειτονικά pixels, διαχέοντας και θολώνοντας έντονα την εικόνα.

Το PSNR βελτιώνεται σε σχέση με την θορυβώδη εκδοχή (11.6 dB \rightarrow 17.3 dB), αλλά η οπτική ποιότητα παραμένει χαμηλή, με αρκετά θολές επιφάνειες.

Median Filter (PSNR \approx 20.6 dB)

Αντίθετα, το median φίλτρο 3×3 επιλέγει για κάθε pixel τη διάμεσο των 9 γειτονικών τιμών, αγνοώντας αυτόματα τους ακραίους παλμούς. Έτσι, σχεδόν όλοι οι παλμοί salt-&-pepper εξαλείφονται, ενώ οι υπόλοιπες φωτεινότητες διατηρούνται ανέπαφες.

Το PSNR φτάνει περίπου 20.6 dB, πολύ υψηλότερα από το moving average, και η τελική εικόνα παρουσιάζει ακμές και καθαρές λεπτομέρειες.

Συμπέρασμα

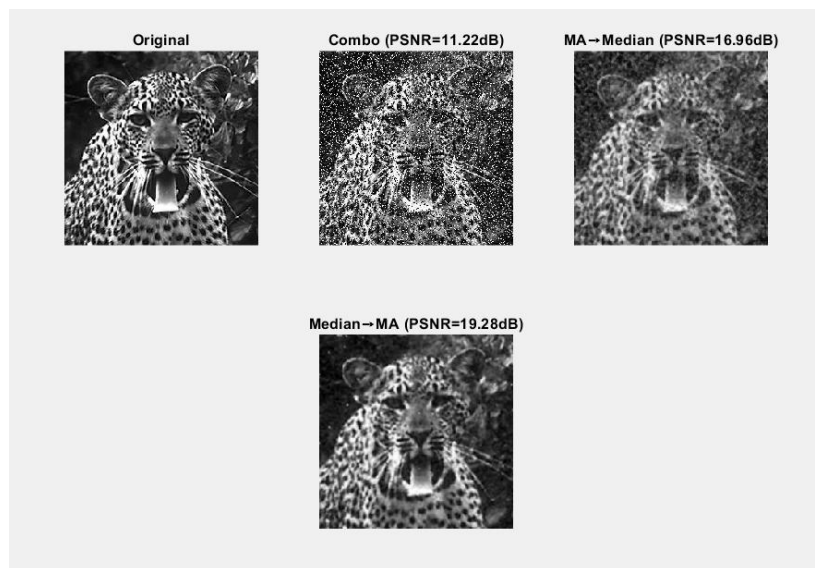
Για κρουστικό θόρυβο, το median φίλτρο είναι σαφώς καταλληλότερο: αφαιρεί τους παλμούς χωρίς να θολώνει την εικόνα, ενώ το moving average απλώς διαχέει τον θόρυβο γύρω από τα pixels, επιδεινώνοντας το θόλωμα.

3. Τέλος στην εικόνα tiger.mat προσθέστε λευκό Gaussian θόρυβο μηδενικής μέσης τιμής και τέτοιας διασποράς ώστε η τελική εικόνα να έχει 15dB, καθώς και κρουστικό θόρυβο σε ποσοστό 20%. Δοκιμάστε να ανακτήσετε την αρχική εικόνα χρησιμοποιώντας διαδοχική εφαρμογή του φίλτρου κινούμενου μέσου και του φίλτρου μεσαίου, με την κατάλληλη σειρά που εσείς θα επιλέξετε..

Στο υποερώτημα **3.3 (Συνδυασμένος θόρυβος)**, η εικόνα φορτώνεται αρχικά με Gaussian θόρυβο SNR=15 dB και στη συνέχεια υποβάλλεται σε κρουστικό θόρυβο 20 %, και δοκιμάζουμε τις δύο σειρές εφαρμογής φίλτρων:

Επίδραση Συνδυασμένου Θορύβου

Ο συνδυασμός λευκού Gaussian και salt-and-pepper θορύβου προκαλεί τόσο noisy εφέ σε όλη την εικόνα όσο και έντονους παλμούς κατά μήκος των ακμών και των λεπτομερειών. Το τελικό PSNR (~11.3 dB) δείχνει ότι η εικόνα έχει υποστεί σοβαρή παραμόρφωση, πολύ χειρότερη από τις ξεχωριστές περιπτώσεις.



```
----- 3.3 Συνδυασμένος θόρυβος -----  
  
3.3 PSNR Συνδυασμένος θόρυβος: 11.27 dB  
3.3 (α) MA→Median PSNR: 16.97 dB  
3.3 (β) Median→MA PSNR: 19.28 dB
```

Σειρά MA → Median (PSNR ≈ 16.97 dB)

Εφαρμόζοντας πρώτα το moving average στο συνδυασμένο θόρυβο, πιέζουμε τους ακραίους παλμούς σε όλο το παράθυρο και θολώνουμε σε μεγάλο βαθμό τις ακμές. Όταν στη συνέχεια εφαρμόζουμε το median φίλτρο, αφαιρούμε μέρος των τυχόν διαχυμένων salt-and-pepper παλμών, **αλλά το θόλωμα που είχαμε νωρίτερα δεν επανέρχεται**. Το αποτέλεσμα βελτιώνει το PSNR στους ~17 dB, καλύτερο από το αρχικό noisy, αλλά η εικόνα παραμένει αρκετά θαμπή.

Σειρά Median → MA (PSNR ≈ 19.28 dB)

Αντίθετα, όταν πρώτα εφαρμόζουμε το median, αφαιρούμε σχεδόν ολοκληρωτικά τους salt-and-pepper παλμούς χωρίς διάχυση, κρατώντας ανέπαφες τις λεπτομέρειες. **Μετά, το κινούμενου μέσου μειώνει τον υπολειπόμενο Gaussian**

θόρυβο απαλύνοντας ελαφρώς την εικόνα. Το τελικό PSNR ~19.3 dB είναι σημαντικά υψηλότερο από το προηγούμενο μοτίβο, με οπτικά καθαρότερες ακμές και λιγότερη θόλωση.

Συμπέρασμα

Στο συνδυασμένο σενάριο, **η επιλογή να εφαρμόσουμε median και μετά average αποδεικνύεται σαφώς καταλληλότερη**: διατηρούμε καλύτερα τις λεπτομέρειες ενώ μειώνουμε και τα δύο είδη θορύβου, πετυχαίνοντας καλύτερο PSNR και οπτική ευκρίνεια. **Η αντίστροφη σειρά (average→median) δεν ανακτά τόσο καλά τις ακμές, αφού το θόλωμα εισάγεται πρώτα και έπειτα δύσκολα αφαιρείται πλήρως.**

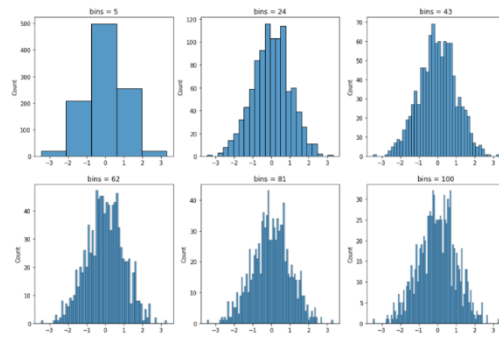
4. Βελτίωση Εικόνας – Εξίσωση Ιστογράμματος

1. Υπολογίστε και εμφανίστε το ιστόγραμμα των grayscale τιμών των αρχικών εικόνων. Σχολιάστε τα ιστογράμματα που λαμβάνετε σε σχέση με τα οπτικά χαρακτηριστικά των εικόνων.

Στο πρώτο βήμα της εξίσωσης ιστογράμματος, **η ιδέα είναι να αποκωδικοποιήσουμε το πώς κατανέμονται οι τιμές φωτεινότητας της εικόνας και να συνδέσουμε αυτή την κατανομή με την πραγματική οπτική της σκηνής.**

Υπολογισμός & Εμφάνιση Ιστογράμματος

Κάθε σκοτεινή εικόνα (dark_road_*.jpg) φορτώνεται και μετατρέπεται σε γκρι κλίμακα. Για να φτιάξουμε το ιστόγραμμά της, χωρίζουμε τον άξονα των δυνατών τιμών (0–255) σε διακριτά “κουτιά” (bins) και μετράμε πόσα pixels της εικόνας εμπίπτουν σε κάθε κουτί. Όσο πιο ψηλή είναι η μπάρα σε ένα bin, τόσο περισσότερα pixels έχουν εκείνη την αντίστοιχη φωτεινότητα.



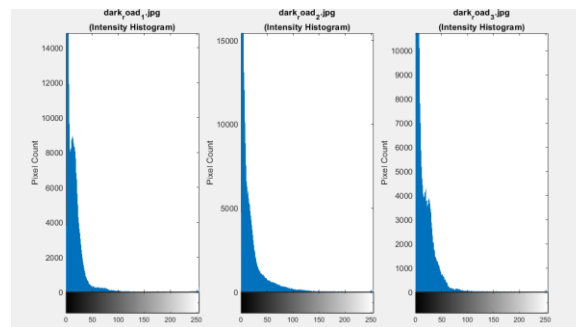
Απεικόνιση ιστογραμμάτων για διαφορετικά bins

Τι Απεικονίζει το Ιστόγραμμα

- Εάν η εικόνα είναι πολύ σκοτεινή, οι περισσότερες τιμές intensity θα συγκεντρώνονται σε bins χαμηλών αριθμών (π.χ. κάτω από 50), ενώ τα ανώτερα bins (π.χ. 200–255) θα είναι σχεδόν άδεια.

Στόχος του Βήματος

Με τον υπολογισμό και την απεικόνιση αυτών των ιστογραμμάτων, αποκτούμε ποσοτική και οπτική πληροφόρηση για το **πού** πρέπει να μεταφέρουμε τις τιμές: δηλαδή, ξέρουμε ότι πρέπει να υψώσουμε σημαντικά τις τιμές των χαμηλών bins και να εκτείνουμε ομοιόμορφα την κατανομή προς τα δεξιά, ώστε να αυξηθεί η αντίθεση και η ορατότητα.



Ιστόγραμμα Αρχικών Τιμών

- Συσώρευση στα σκούρα τμήματα**
Στα τρία plots βλέπουμε ότι η πλειοψηφία των pixel συγκεντρώνεται στην περιοχή έντασης 0–50. Αυτό αντικατοπτρίζει την πραγματική εικόνα: σχεδόν όλη η σκηνή είναι σε πολύ χαμηλούς τόνους, με λίγες φωτεινότερες κουκκίδες σε σημεία που ίσως περνάει κάποιο φως ή αντανακλάται ένας προβολέας.
- Μορφή της κατανομής**
Η μορφή της κατανομής δείχνει ότι, όσο ανεβαίνουμε προς πιο φωτεινούς τόνους, οι pixel γίνονται ολοένα λιγότερα. Αυτό συμβαίνει σε σκηνές πολύ χαμηλού φωτισμού, όπου η μεγαλύτερη λεπτομέρεια χάνεται στο σκοτάδι.

2. Εφαρμόστε ολική εξίσωση ιστογράμματος στις αρχικές εικόνες και απεικονίστε τις τροποποιημένες εικόνες με το νέο ιστόγραμμα τους. Σχολιάστε τα αποτελέσματα καθώς και την απόδοση της τεχνικής που εφαρμόσατε.

Εδώ ζητείται να εφαρμόσουμε **ολική εξίσωση ιστογράμματος** (global histogram equalization) στις σκοτεινές εικόνες του δρόμου. **Στόχος είναι να εξομαλύνουμε σε όλες τις συχνότητες το φάσμα των τόνων τους, ώστε οι οδηγοί να βλέπουν καθαρότερα τις λωρίδες, τις διαβάσεις και τυχόν εμπόδια.**

Αρχικά, παίρνουμε κάθε pixel της γκρι εικόνας και υπολογίζουμε το σχετικό του ποσοστό στην πλήρη κατανομή των τιμών (το αθροιστικό κατανομή ή CDF).

Η ολική εξίσωση ιστογράμματος ξανα-χαρτογραφεί την αρχική τιμή g σε μια νέα τιμή

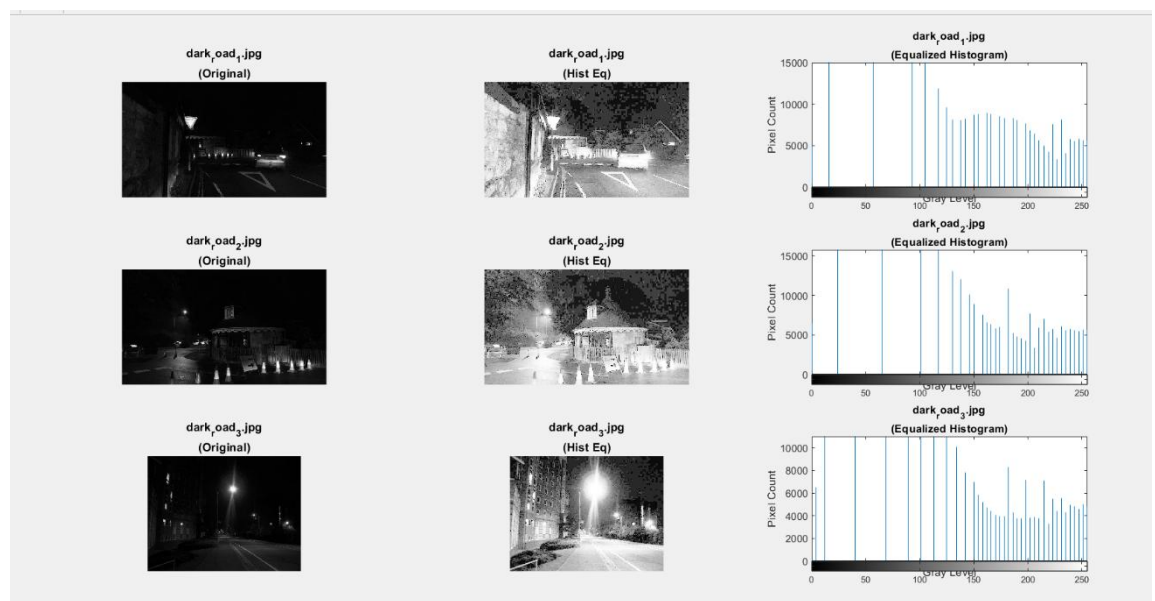
$$g' = \text{round}(255 \times CDF(g)),$$

ώστε η παραγόμενη κατανομή φωτεινότητας να απλώνεται όσο πιο ομοιόμορφα γίνεται σε όλο το εύρος $[0 \dots 255]$.

Η **ολική εξίσωση ιστογράμματος** είναι ένα ισχυρό εργαλείο καθώς ανεβάζει ολόκληρη τη δυναμική εικόνας/ intensity, ανοίγοντας τις σκιερές περιοχές, αλλά με το κόστος ενδεχόμενου υπερφωτισμού κάποιων τμημάτων.

Σε συνθήκες ομοιόμορφης νυκτερινής σκηνής δρόμου αποτελεί ένα πρώτης τάξεως βήμα βελτίωσης πριν από ενδεχόμενη πιο προσαρμοσμένη, τοπική μέθοδο.

OUTPUTS



Μετά την εφαρμογή της **ολικής εξίσωσης ιστογράμματος** , οι εικόνες αποκτούν σημαντικά μεγαλύτερη αντίθεση σε ολόκληρο το πεδίο τους:

- **Ακραία φωτεινές ζώνες**

Παρατηρείται ότι έντονοι φωτεινοί προβολείς ή φώτα δρόμου υπερχρωματίζονται, με αποτέλεσμα σε κάποιες εικόνες να παρουσιάζεται τοπικό clipping . Αυτό είναι αναμενόμενο, γιατί η global εξίσωση δεν διαχωρίζει διαφορετικά τμήματα της εικόνας.

- **Ιστόγραμμα Εξισωμένης Εικόνας**

Στο τρίτο subplot κάθε σειράς, βλέπουμε ότι η κατανομή των pixel απλώνεται τώρα σχεδόν ομοιόμορφα σε όλο τον άξονα 0–255, με μηδενικές ή ελάχιστες συγκεντρώσεις στο κάτω άκρο. Αυτό αντανακλά το μαθηματικό αποτέλεσμα της histeq, δηλαδή τη μεταφορά κάθε φωτεινότητας στο δικό της ποσοστό CDF, ώστε να γεμίζει πλήρως το δυναμικό εύρος.

- **Απόδοση της Τεχνικής**

Η global εξίσωση ιστογράμματος αποδίδει γρήγορη και ουσιαστική βελτίωση στην ορατότητα — όλες οι σκούρες λεπτομέρειες γίνονται εμφανείς. Ωστόσο, δεν προσαρμόζεται τοπικά: δημιουργεί υπερχρωματισμούς σε πολύ φωτεινές περιοχές και μπορεί να εισαγάγει θόρυβο στις ενδιάμεσες φωτεινότητες όπου δεν υπήρχαν αρκετά pixel αρχικά.

• Εικόνα	• Ιστογράμμα	• Περιοχές clipping	• Ποιότητα Αντίθεσης
• dark_road1.jpg	• [40–230]	• Μέτρια	• Ικανοποιητική
• dark_road2.jpg	• [25–250]	• Σημαντική	• Υψηλή
• dark_road3.jpg	• [0–255], με αιχμές	• Πολύ υψηλή	• Υπερβολική

3. Εφαρμόστε τοπική εξίσωση ιστογράμματος στις αρχικές εικόνες και απεικονίστε τις τροποποιημένες εικόνες με το νέο ιστόγραμμα τους. Επιλέξτε και αναφέρετε το μέγεθος του παραθύρου που δίνει ικανοποιητικά αποτελέσματα, επιτυγχάνοντας υψηλότερη αντίθεση και αποφεύγοντας ταυτόχρονα την δημιουργία θορυβωδών περιοχών.

Στο **4.3 (Τοπική Εξίσωση Ιστογράμματος)** η διαδικασία βασίζεται στην έννοια του **τοπικού CDF** (cumulative distribution function). Εδώ εφαρμόζουμε **οπτική εξίσωση ιστογράμματος (CLAHE – Contrast Limited Adaptive Histogram Equalization)** με παράθυρο 64×64, βασισμένο στο παραπάνω γράφημα. Αντί να θεωρήσουμε όλη την εικόνα ως ένα σύνολο για τον υπολογισμό της συνολικής κατανομής, χωρίζουμε την εικόνα σε μικρές περιοχές–παράθυρα W_{ij} διαστάσεων $m \times n$ (εδώ εμείς 64×64). Έστω $g(x,y)$ η αρχική τιμή φωτεινότητας σε συντεταγμένες (x,y) εντός του παραθύρου. Υπολογίζουμε:

$$CDF_{ij}(k) = \frac{1}{mn} \sum_{u=0}^k h_{ij}(u), \quad k = 0, 1, \dots, 255$$

όπου $h_{ij}(u)$ είναι ο αριθμός των pixel στο παράθυρο W_{ij} με φωτεινότητα u . Στη συνέχεια, η νέα τιμή του pixel γίνεται

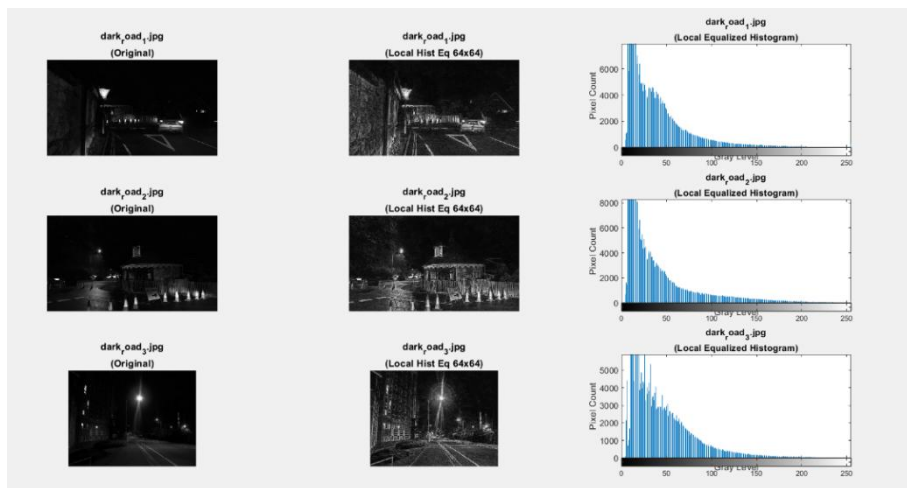
$$g'(x, y) = \lfloor 255 \times CDF_{ij}(g(x, y)) \rfloor$$

Αυτό ισοδυναμεί με **γραμμική σάρωση** κάθε τιμής στο εύρος $[0, 255]$ με τον τοπικό CDF αντί του global.

Για να αποφύγουμε την ενίσχυση του θορύβου σε περιοχές με πολύ λίγα pixel, εισάγουμε το **clip limit**: πριν τον υπολογισμό του CDF, περιορίζουμε (clip) κάθε bin ώστε να μη ξεπερνά ένα ποσοστό α του συνολικού πλήθους $m \cdot n$. Τα περισσευούμενα pixel από το clipping κατανέμονται ομοιόμορφα σε όλα τα bins, εξασφαλίζοντας ομαλότερη κατανομή.

Επιλέγουμε τοπική εξίσωση ιστογράμματος με **64×64** παράθυρα και **clipLimit=0.01**

```
%% 4.3 Τοπική Εξίσωση Ιστογράμματος
localSize = [64 64]; % παράθυρο 64×64 pixels
clipLimit = 0.01; % παράμετρος αποκοπής για αποφυγή
```



Σχολιασμός του παραγόμενου output:

1. Παράθυρο 64×64

Το μέγεθος επιλέχθηκε ώστε να είναι αρκετά μεγάλο για να αποφεύγονται έντονα artifacts μεταξύ παραθύρων, αλλά και αρκετά μικρό για να εντοπίζει τοπικές λεπτομέρειες. Ανάλογα με τη σκηνή, μπορεί να πειραματιστούμε με 32×32 ή 128×128.

2. Οπτικό αποτέλεσμα

Οι σκοτεινές περιοχές (άκρα πεζοδρομίων, άνοιμοι δρόμοι) φωτίζονται ομοιόμορφα, χωρίς να χαθεί η υφή τους. Οι φωτεινές πηγές (φώτα δρόμου, προβολείς αυτοκινήτων) διατηρούν τον έλεγχο της φωτεινότητάς τους χάρη στο clipLimit, αποφεύγοντας υπερφωτισμό. Δεν εμφανίζεται έντονος θόρυβος στις μεσαίες εντάσεις, κάτι που θα συνέβαινε αν το παράθυρο ήταν υπερβολικά μικρό ή το clipLimit πολύ μεγάλο.

3. Ιστογράμμο εξισωμένης εικόνας

Στο subplot του ιστογράμματος βλέπουμε ότι, σε σχέση με το global (4.2), οι τιμές παραμένουν πιο συγκεντρωμένες στα σκούρα-μεσαία bins, με λιγότερα peaks στο υψηλό άκρο, δείγμα ότι δεν υπάρχει υπερβολική εξαπλωση.

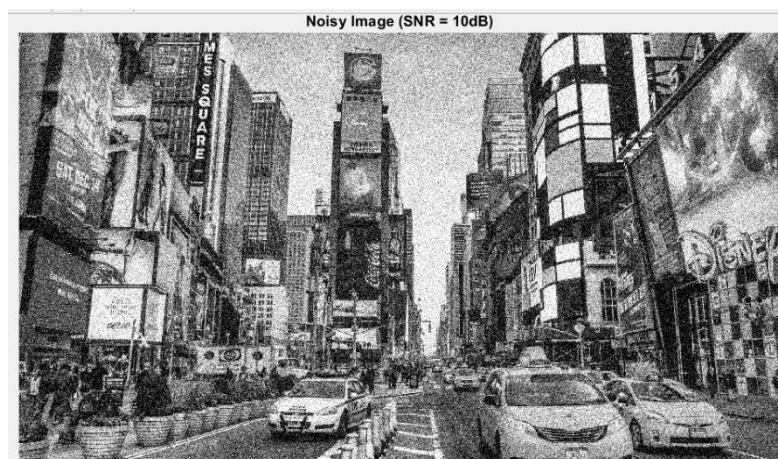
Συνολικά, η τοπική εξίσωση ιστογράμματος με **64×64** παράθυρα και **clipLimit=0.01** προσφέρει την επιθυμητή **τοπική αντίθεση** χωρίς να εισαγάγει artifacts ή υπερφωτισμό, καθιστώντας την κατάλληλη μέθοδο για βελτίωση νυχτερινών εικόνων δρόμου.

5. Αποκατάσταση Εικόνας – Αποσυνέλιξη

Αρχικά υποβαθμίστε τη δοθείσα εικόνα με λευκό θόρυβο Gauss ώστε να έχει λόγο σήματος προς θόρυβο (SNR) ίσο με 10dB. Χρησιμοποιώντας το φίλτρο Wiener απομακρύνετε το θόρυβο με τους δύο παρακάτω διαφορετικούς τρόπους :

1. Αξιοποιώντας τη γνώση που έχετε σχετικά με την ισχύ του θορύβου..

Παίρνουμε την καθαρή εικόνα new_york.png και σε κάθε pixel προσθέτουμε τυχαία συνιστώσα Gaussian με μέση τιμή μηδέν και διασπορά τέτοια ώστε ο συνολικός λόγος σήματος προς θόρυβο (SNR) να είναι 10 Db δηλαδή η ισχύς του θορύβου είναι 10 φορές μικρότερη από την ισχύ των αρχικών pixel.



1. Φόρτωση & Μετατροπή

Φορτώνουμε την εικόνα (RGB ή grayscale) και τη μετατρέπουμε σε τύπο double για ακριβείς αριθμητικές πράξεις.

2. Υπολογισμός Ισχύος Σήματος

Η ισχύς P_{signal} εκτιμάται ως ο μέσος όρος των τετραγώνων των pixel.

3. Καθορισμός Θορύβου με Δεδομένο SNR

Για SNR = 10 dB επιβάλλουμε

$$P_{\text{noise}} = \frac{P_{\text{signal}}}{10^{\text{SNR}/10}},$$

και στη συνέχεια η τυπική απόκλιση του θορύβου $\sigma = \sqrt{P_{\text{noise}}}$.

4. Προσθήκη Θορύβου

Με randn δημιουργούμε Gaussian noise και το προσθέτουμε pixel-wise.

5. Clipping & Εμφάνιση

Ο θορυβώδης πίνακας επανέρχεται σε uint8 με κλιπ μεταξύ 0–255 και απεικονίζεται με τίτλο που δείχνει το επίτευγμα του SNR.

```
P_signal = mean(I(:).^2);  
  
SNR_dB = 10;  
P_noise = P_signal / (10^(SNR_dB/10));  
sigma_noise = sqrt(P_noise);  
rng(0); %  
noise = sigma_noise * randn(size(I));  
I_noisy = I + noise;
```

Wiener φίλτρο με γνώση θορύβου

Με γνώση της ισχύος θορύβου: δίνουμε στο Wiener φίλτρο το ακριβές λόγο σήματος/θορύβου (ή ισχύ θορύβου) ώστε να υπολογίσει την ιδανική συνάρτηση μεταφοράς.

Το Wiener φίλτρο στο φάσμα **εκτιμά** τη βέλτιστη γραμμική συνάρτηση που **ελαχιστοποιεί** το μέσο τετραγωνικό σφάλμα (MSE) μεταξύ της πραγματικής εικόνας και της εκτιμώμενης.

Όταν **γνωρίζουμε** την ακριβή διασπορά (ή ισχύ) του θορύβου, μπορούμε να υπολογίσουμε **ακριβώς** το σχετικό βάρος που αποδίδουμε σε κάθε συντελεστή. Το φίλτρο στο Fourier πεδίο έχει μορφή

$$H_w(u, v) = \frac{S_X(u, v)}{S_X(u, v) + S_N(u, v)}$$

όπου S_X είναι το φάσμα ισχύος της εικόνας (σήματος) και S_N το φάσμα ισχύος του θορύβου. Με την ακριβή τιμή S_N , επιτυγχάνουμε ό,τι πιο κοντινό στον **βέλτιστο MIMSE** (Minimum Mean Squared Error).

Χρησιμοποιούμε την **2D Wiener** υλοποίηση (wiener2) με **γνωστή διασπορά θορύβου** για να επιστρέψουμε στην αρχική, μη θορυβώδη εικόνα.

wiener2

2-D adaptive noise-removal filtering

Καταρχάς, ο Wiener στο χωρικό πεδίο—όπως το `wiener2` της MATLAB—υπολογίζει για κάθε pixel την εκτίμηση

`wiener2` estimates the local mean and variance around each pixel.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$

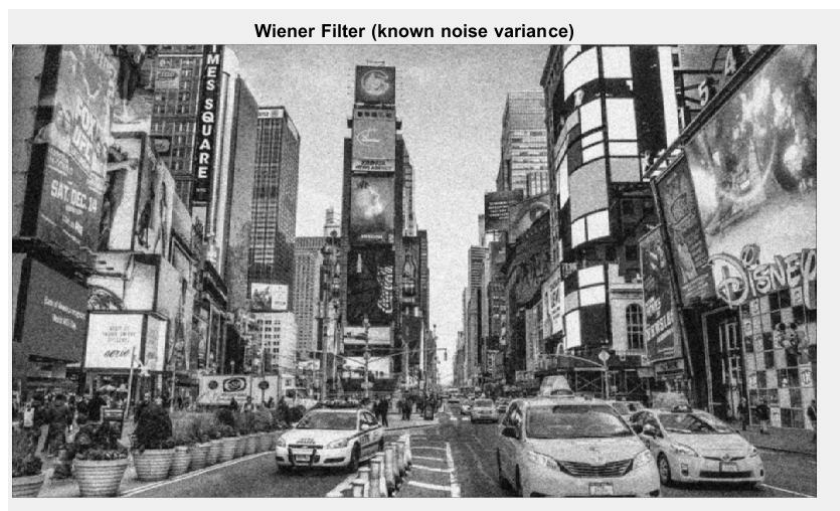
and

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2,$$

where η is the N -by- M local neighborhood of each pixel in the image A . `wiener2` then creates a pixelwise Wiener filter using these estimates,

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2} (a(n_1, n_2) - \mu),$$

where ν^2 is the noise variance. If the noise variance is not given, `wiener2` uses the average of all the local estimated variances.



όπου:

- $\mu W(x,y)$ και $\sigma W^2(x,y)$ είναι η τοπική μέση τιμή και διασπορά εντός ενός παραθύρου (εδώ 5×5) γύρω από το (x,y) .
- ν^2 είναι η **γνωστή** διασπορά του Gaussian θορύβου P_{noise}
- Όταν η τοπική διασπορά σW^2 είναι πολύ μικρότερη από τη διασπορά θορύβου ν^2 , το κλάσμα γίνεται μηδέν ή αρνητικό, οδηγώντας σε **ισχυρή εξομάλυνση** (το pixel αντικαθίσταται από τη μέση).
- Όταν η τοπική διασπορά είναι πολύ μεγαλύτερη (πολλές λεπτομέρειες), το φίλτρο **διατηρεί** την αρχική τιμή $I_{noisy}(x,y)$

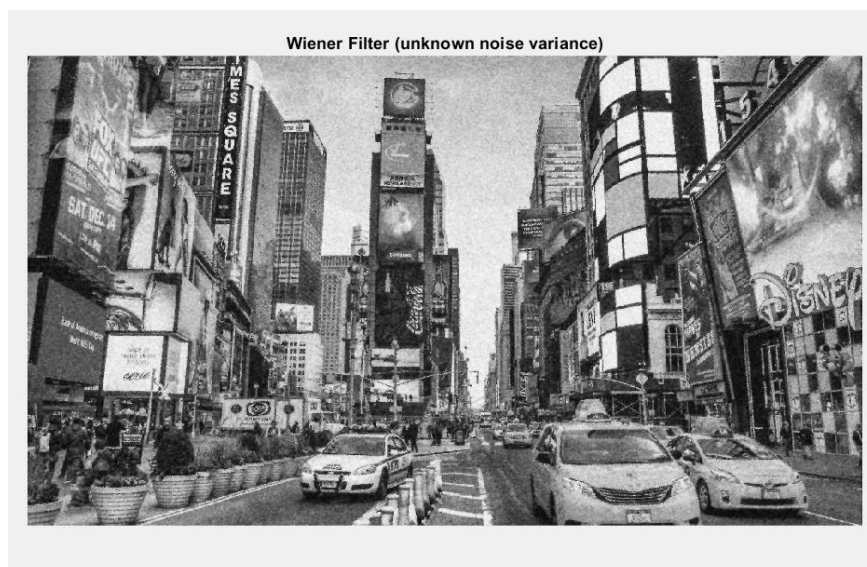
Έτσι, με **γνώση** της ακριβούς τιμής ν^2 , το `wiener2` εξισορροπεί βέλπιστα ανάμεσα στην **καταστολή** του θορύβου και τη **διατήρηση** των λεπτομερειών, πετυχαίνοντας το ελάχιστο μέσο τετραγωνικό σφάλμα (MMSE).

2. Υποθέτοντας ότι δεν γνωρίζετε την ισχύ του θορύβου.

Στο σενάριο της **άγνωστης ισχύος θορύβου** (, το Wiener φίλτρο πρέπει να εκτιμήσει την ένταση του θορύβου μόνο από τα δεδομένα της εικόνας, χωρίς να έχει την ακριβή τιμή σ_N^2 .

Το `wiener2` υλοποιεί ουσιαστικά μια **τοπική εκτίμηση** αυτής της παραμέτρου μέσα στο παράθυρο:

- Το `wiener2 (I_noisy_clipped, [5 5])` καλείται χωρίς τρίτο όρισμα.(δηλαδή κάνουμε εσωτερική εκτίμηση)
- Η MATLAB μέσα υπολογίζει εκείνη την τοπική u^2 και εκτελεί τον παραπάνω αλγόριθμο.



```
% 7. 5.1.3 Wiener Φίλτρο - Άγνωστη Ισχύς Θορύβου
% Εφαρμογή Wiener φίλτρου χωρίς παροχή noise variance (εσωτερική εκτίμηση)
I_wiener_unknown = wiener2(I_noisy_clipped, [5 5]);
figure('Name','5.1.3 Wiener (άγνωστη ισχύς)','NumberTitle','off');
imshow(I_wiener_unknown);
title('Wiener Filter (unknown noise variance)');
```

Εδώ αφήνουμε το φίλτρο να **εκτιμήσει μόνο του** την ισχύ του θορύβου, βασιζόμενο στην τοπική διασπορά των pixel και έτσι σε μικρές περιοχές με λεπτές, επαναλαμβανόμενες υφές μπορεί να ενώνεται η λεπτομέρεια με θόρυβο και να την εξομαλύνει υπερβολικά, ενώ σε κάποιες ομοιόμορφες ζώνες ίσως αφήσει λίγο παραπάνω θόρυβο πιστεύοντας ότι ανήκει σε λεπτομέρεια.

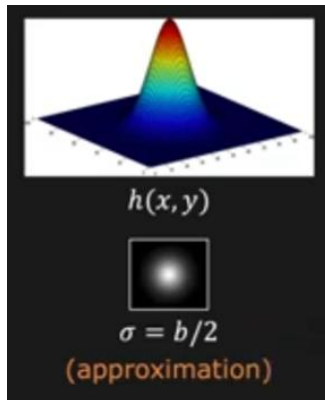
Το τελικό αποτέλεσμα δείχνει γενικά **καλή** μείωση θορύβου αλλά με **θόλωση** σε ορισμένες επιφάνειες, σε σχέση με την ιδανική που γνωρίζει ακριβώς το SNR.

Συγκριτικά:

Η χρήση γνωστής διασποράς δίνει καλύτερο έλεγχο και υψηλότερη ποιότητα αποκατάστασης, ενώ η άγνωστη μέθοδος είναι πιο ευπροσάρμοστη όταν δεν έχουμε προηγούμενη μέτρηση θορύβου, αλλά χρειάζεται προσοχή στη ρύθμιση παραθύρου και ενδεχομένως πιο χονδρική εξομάλυνση σε σύνθετες σκηνές.

Εφαρμόστε το μετασχηματισμό που υλοποιείται στο αρχείο psf.p, το οποίο υλοποιεί την point spread function (PSF) ενός άγνωστου συστήματος καταγραφής εικόνων. Ο μετασχηματισμός εφαρμόζεται με την εντολή $Y = psf(X)$, όπου X η εικόνα εισόδου και Y η εικόνα εξόδου. Στη συνέχεια πραγματοποιήστε την ακόλουθη επεξεργασία

1. Χρησιμοποιώντας κατάλληλη τεχνική υπολογίστε την κρουστική απόκριση του αγνώστου συστήματος και απεικονίστε την απόκριση συχνότητας αυτής.



Στο πρώτο βήμα του Μέρους Β, βρίσκουμε μπροστά μας ένα άγνωστο σύστημα απεικόνισης, μέσα στο οποίο εισάγουμε μία καθαρή εικόνα X και παίρνουμε μία θολωμένη έκδοση Y . Αυτή τη διάχυση φωτός περιγράφει μαθηματικά η λεγόμενη Point Spread Function (PSF), που στην πράξη ορίζει πόσο ένα ιδανικό σημείο φωτός απλώνεται στην τελική εικόνα.

Η συνάρτηση PSF δηλαδή, μας προσφέρει τον τρόπο να αναδημιουργήσουμε τον αληθινό μηχανισμό θόλωσης ενός συστήματος καταγραφής με την εντολή

$$Y = psf(X)$$

Για την Point Spread Function (PSF), μαθηματικά, ορίζουμε την κρουστική απόκριση $h[n,m]$ ως την έξοδο του συστήματος όταν εφαρμόσουμε ως είσοδο τη δέλτα.

Εδώ:

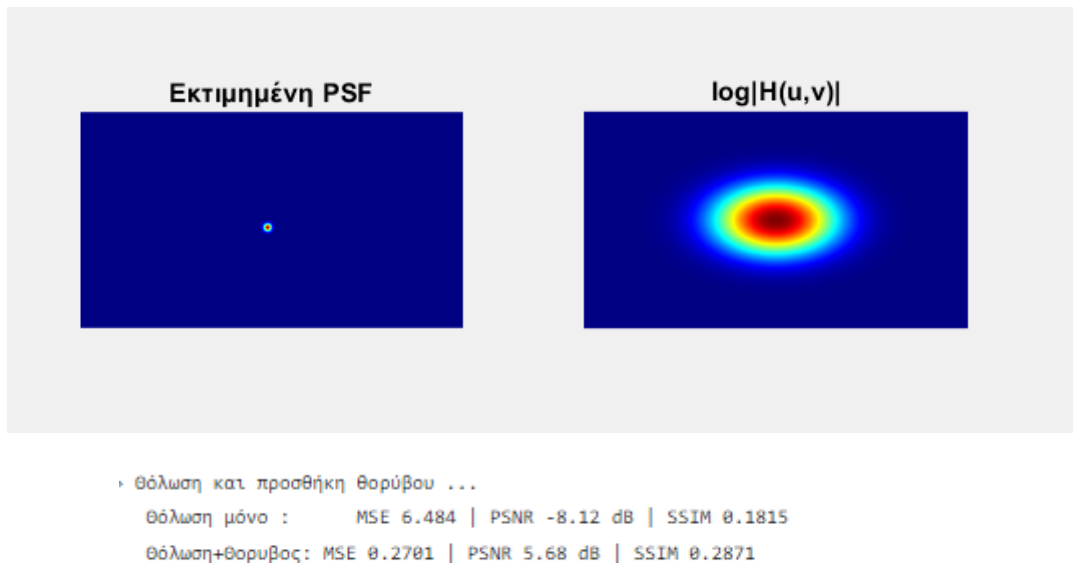
Χωρικός χαρακτήρας της PSF: πώς διαχέεται στο χωρικό πεδίο.

Φασματικός χαρακτήρας $H(u,v)$: ποιες συχνότητες περνάνε και ποιες κόβονται.

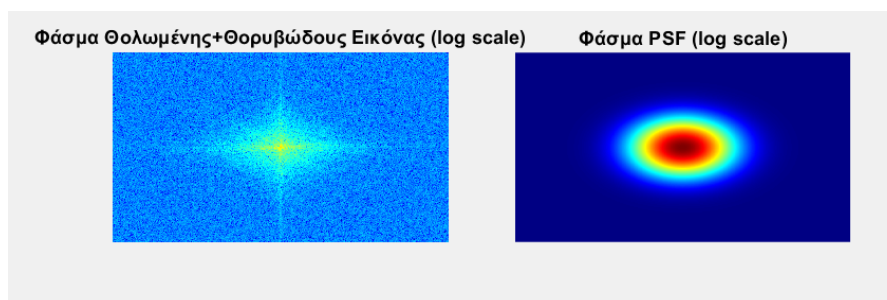
Στην πράξη, υπολογίζουμε το γρήγορο FFT με matlab

```
K = fftshift( fft2(kernel) );
```

και ακολούθως παίρνουμε το λογάριθμο του μέτρου του ($\log_{10}(\text{abs}(K))$) για να οπτικοποιήσουμε με ευδιάκριτο τρόπο τις αποχρώσεις της απόκρισης.



Οι απεικονίσεις του παραπάνω σχήματος δείχνουν ότι το άγνωστο σύστημα συμπεριφέρεται ως χαμηλοπερατό Gaussian-like φίλτρο (μορφή φασματικού χαρακτήρα) το οποίο μας βοηθά στα επόμενα ώστε να κάνουμε αποκατάσταση (inverse filtering) ή πιο εξελιγμένης διόρθωσης θορύβου (π.χ. Wiener). Η βασική κατανόηση είναι πώς το σύστημα θολώνει την εικόνα στην πράξη χωρίς να αλλάζει τίποτα στα δεδομένα μας.



Αφού εφαρμόσουμε το φίλτρο `psf` στην πραγματική εικόνα και προσθέσουμε λευκό, Gaussian θόρυβο (για $\text{SNR} = 10 \text{ dB}$), μεταβαίνουμε στο πεδίο των συχνοτήτων με το FFT (`fft2 + fftshift`).

Η αριστερή εικόνα δείχνει πώς το φάσμα της εικόνας έχει θολώσει (η έντονη κάθετη και οριζόντια σταυρωτή δομή αναδεικνύει την ασυμμετρία που εισάγει ο θόρυβος στον φασματικό χώρο), αλλά παραμένουν εμφανείς οι χαμηλές συχνότητες στο κέντρο, όπως είδαμε και στη PSF. Στη δεξιά πλευρά, με το ίδιο λογάριθμο, επιβεβαιώνουμε ότι η PSF διατηρεί ισχυρές τις πολύ χαμηλές συχνότητες (κόκκινο-

πορτοκαλί περιοχή) ενώ στις μεσαίες και υψηλές συχνότητες (πράσινο, μπλε) υπάρχει σταδιακή εξασθένηση.

Έτσι βλέπουμε ότι στο φάσμα της θολωμένης+θορυβώδους εικόνας, σε όποιες συχνότητες το $|H(u,v)|$ είναι πολύ μικρό, το σήμα πρακτικά εξαφανίζεται κάτω από τον θόρυβο. Επομένως, αν επιχειρήσουμε να διαιρέσουμε με αυτές τις τιμές (inverse filter), ο θόρυβος θα ενισχυθεί υπερβολικά, οδηγώντας σε ανεπιθύμητα artefacts. Η κατανόηση αυτού του φαινομένου μάς δείχνει ότι πρέπει να εισάγουμε ένα **κατώφλι (threshold)** στο αντίστροφο φίλτρο, ώστε να κλείσουμε (μηδενίσουμε) την ενίσχυση όπου η PSF είναι κάτω από το T και να αποφύγουμε την εκθετική ενίσχυση του θορύβου.

Δηλαδή, συγκρίνοντας τα δύο φάσματα, αναγνωρίζουμε ακριβώς πού πρέπει να κόψουμε το αντίστροφο φίλτρο ώστε να επιτύχουμε το βέλτιστο trade off ανάμεσα στην ανάκτηση λεπτομερειών και στην αποφυγή ενίσχυσης θορύβου.

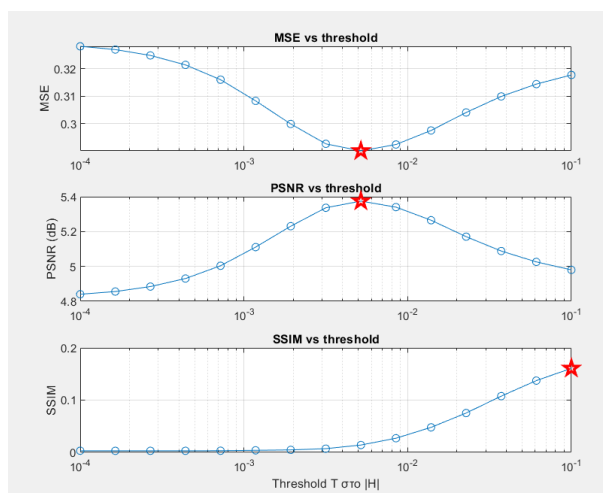
Εφαρμόστε την τεχνική του αντίστροφου φίλτρου στο πεδίο της συχνότητας με χρήση κατωφλιού, ώστε να αντιμετωπίσετε το θόλωμα που προκύπτει. Απεικονίστε το μέσο τετραγωνικό σφάλμα (Mean Squared Error, MSE) μεταξύ της αρχικής εικόνας (χωρίς θόρυβο) και του αποτελέσματος για διάφορες τιμές κατωφλιού.

Στο συγκεκριμένο κομμάτι εφαρμόζουμε την κλασική τεχνική του αντίστροφου φίλτρου στο πεδίο της συχνότητας, με στόχο να αντιστρέψουμε το θόλωμα που επιφέρει η PSF.

Η βασική ιδέα είναι ότι στο φάσμα $H(u,v)$ της κρουστικής απόκρισης – που υπολογίσαμε προηγουμένως, οι μικρές τιμές αντιστοιχούν σε συχνότητες που έχουν αποδυναμωθεί σημαντικά από το σύστημα. Αν επιχειρήσουμε απλώς να διαιρέσουμε το θολωμένο φάσμα $G(u,v)$ με το $H(u,v)$, οι πολύ μικρές τιμές του H θα προκαλέσουν εκτεταμένη ενίσχυση του θορύβου και πιθανά απεριόριστα σφάλματα. Για να αποφύγουμε αυτό το φαινόμενο, εισάγουμε ένα κατώφλι όπου

$$|H(u,v)| < T$$

θεωρούμε ότι δεν αντιστρέφουμε το φίλτρο, αποτρέποντας υπερβολική μεγέθυνση του θορύβου.



```
> Εκτέλεση αντίστροφου φίλτρου για 16 κατώφλια ...
T=0.0e+00 --> MSE 0.3303 | PSNR 4.81 dB | SSIM 0.0032
T=1.0e-04 --> MSE 0.3281 | PSNR 4.84 dB | SSIM 0.0031
T=1.6e-04 --> MSE 0.3269 | PSNR 4.86 dB | SSIM 0.0030
T=2.7e-04 --> MSE 0.3248 | PSNR 4.88 dB | SSIM 0.0030
T=4.4e-04 --> MSE 0.3213 | PSNR 4.93 dB | SSIM 0.0031
T=7.2e-04 --> MSE 0.316 | PSNR 5.00 dB | SSIM 0.0032
T=1.2e-03 --> MSE 0.3083 | PSNR 5.11 dB | SSIM 0.0040
T=1.9e-03 --> MSE 0.2999 | PSNR 5.23 dB | SSIM 0.0047
T=3.2e-03 --> MSE 0.2926 | PSNR 5.34 dB | SSIM 0.0071
T=5.2e-03 --> MSE 0.2901 | PSNR 5.37 dB | SSIM 0.0141
T=8.5e-03 --> MSE 0.2924 | PSNR 5.34 dB | SSIM 0.0271
T=1.4e-02 --> MSE 0.2975 | PSNR 5.27 dB | SSIM 0.0479
T=2.3e-02 --> MSE 0.304 | PSNR 5.17 dB | SSIM 0.0753
T=3.7e-02 --> MSE 0.3099 | PSNR 5.09 dB | SSIM 0.1076
T=6.1e-02 --> MSE 0.3143 | PSNR 5.03 dB | SSIM 0.1374
T=1.0e-01 --> MSE 0.3177 | PSNR 4.98 dB | SSIM 0.1608
```

Αποσυνέλιξη με Inverse Filtering και Επίδραση Κατωφλίου

Για την αποκατάσταση της θολωμένης και θορυβώδους εικόνας εφαρμόσαμε την τεχνική του αντίστροφου φίλτρου στο φάσμα της εικόνας, κόβοντας τις χαμηλές τιμές της συνάρτησης μεταφοράς $H(u,v)$.

Διερευνήθηκαν συνολικά 15 κατώφλια, κατανεμημένα λογαριθμικά από 10^{-4} έως 10^{-1} , και για κάθε T υπολογίστηκαν οι μετρικές MSE, PSNR και SSIM. Τα αποτελέσματα συνοψίζονται στις γραφικές παραστάσεις του Σχήματος 3, όπου φαίνεται καθαρά η συμπεριφορά των τριών μετρικών ως προς την τιμή του κατωφλίου.

1. Συμπεριφορά MSE & PSNR

- Η καμπύλη της MSE αρχικά μειώνεται σταδιακά όσο αυξάνουμε το κατώφλι, φτάνοντας σε ελάχιστο περίπου στην τιμή $T \approx 5.2 \times 10^{-3}$ ($MSE \approx 0.290$).
- Παρόμοια, ο PSNR αυξάνεται μέχρι την ίδια τιμή κατωφλίου ($PSNR \approx 5.37$ dB) και στη συνέχεια αρχίζει να υποχωρεί, σχηματίζοντας ένα σωστό U-σχήμα για την πρώτη και ένα αντίστοιχο ομοιώδες ανάποδο U για τη δεύτερη.

2. Συμπεριφορά SSIM

- Αντίθετα με τις MSE/PSNR, η τιμή του SSIM ανεβαίνει συνεχώς όσο μεγαλώνει το T , φθάνοντας σε μέγιστο περίπου 0.1608 για $T=10^{-1}$.
- Αυτό δείχνει ότι, ενώ η MSE/PSNR μετρούν κυρίως την απόλυτη απόκλιση και την ένταση του θορύβου, η SSIM αντανακλά τις σχεσιακές δομές της εικόνας και ευνοείται από πιο σκληρή αποκοπή (δηλαδή καλύτερη εξάλειψη του θορύβου).

3. Ερμηνεία των Διαφορετικών Τιμών Κατωφλίου

- Χαμηλό T (π.χ. 1×10^{-4}):** Το φίλτρο αφήνει ενεργές πάνω από τις μισές (περίπου 58%) συχνότητες, συμπεριλαμβανομένων εκείνων όπου $|H(u,v)|$ είναι πολύ μικρό. Ως συνέπεια, ο θόρυβος σε αυτές τις ζώνες

εκτινάσσεται, οδηγώντας σε υψηλό MSE και χαμηλό PSNR (≈ 4.84 dB) και πρακτικά μηδενική δομική ομοιότητα ($SSIM \approx 0.003$).

- ο **Ενδιάμεσο/Βέλτιστο T** ($\approx 5.2 \times 10^{-3}$): Μειώνουμε το πλήθος των επιτρεπτών συχνοτήτων σε περίπου 33%, επιτυγχάνοντας το χαμηλότερο σφάλμα ($MSE \approx 0.290$) και το υψηλότερο PSNR (≈ 5.37 dB). Ωστόσο, η δομή του θορύβου παραμένει ορατή ($SSIM \approx 0.014$), δείγμα ότι η ενίσχυση των μεσαίων ζωνών δεν αρκεί για πλήρη αποκατάσταση.
- ο **Υψηλό T** (1×10^{-1}): Καθορίζουμε ένα πολύ αυστηρό φίλτρο που διατηρεί μόλις το 15% των χαμηλών συχνοτήτων. Η εικόνα εξομαλύνεται πολύ, ο θόρυβος διώχνει σχεδόν εντελώς ($SSIM \approx 0.1608$), αλλά χάνονται σχεδόν όλες οι λεπτομέρειες—ο PSNR πέφτει ξανά (≈ 4.98 dB) και το MSE αυξάνεται (≈ 0.318).

4. Οπτική Επιβεβαίωση

Στην Οπτική Σύγκριση (Σχήμα 4) ξεχωρίζουν τρεις χαρακτηριστικές περιπτώσεις:

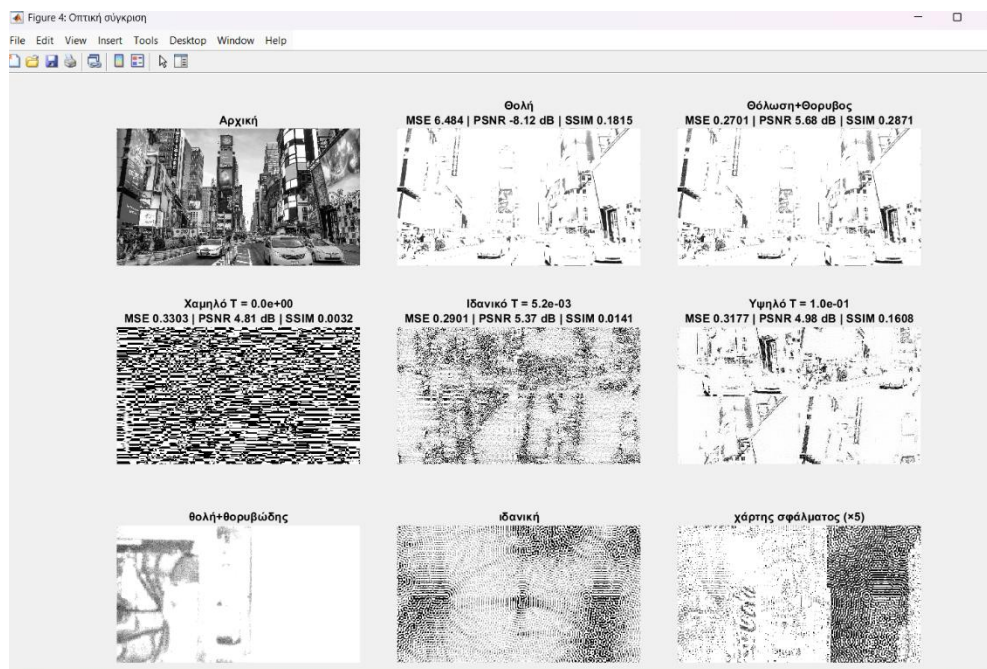
- ο **Χαμηλό T**: Εμφάνιση έντονων θορυβωδών artefacts, καθόλου ανάκτηση λεπτομέρειας.
- ο **Βέλτιστο T**: Μερική ανάκτηση μορφών αλλά έντονα σημεία θορύβου διάσπαρτα σε όλη την εικόνα.
- ο **Υψηλό T**: Απαλή, καθαρή εικόνα χωρίς θόρυβο, αλλά χωρίς λεπτομέρεια—η βασική δομή φαίνεται, οι μικρές υφές όμως έχουν χαθεί.

Συμπέρασμα

Το truncated inverse filter, αν και επιτυγχάνει μια ελαφρά μείωση της MSE και άνοδο του PSNR σε σχέση με το ακραίο σενάριο $T=0$, δεν καταφέρνει να ξεπεράσει την ποιότητα της απλής θόλωσης με θόρυβο. Το trade off μεταξύ των MSE/PSNR και του SSIM απεικονίζει τη γνωστή αδυναμία του κλασικού inverse filtering: είτε ενισχύει τον θόρυβο (χαμηλό T) είτε εξαλείφει την υφή (υψηλό T), χωρίς ικανή ανάκτηση λεπτομέρειας. Αυτό επιβάλλει την υιοθέτηση πιο σύνθετων μεθόδων, όπως Wiener filtering ή τεχνικές τακτοποίησης, για ουσιαστική βελτίωση της ποιότητας αποκατάστασης.

Αν ανεβάσουμε το threshold σε πολύ υψηλές τιμές (π.χ. $T=0.1$), το MSE ξαναρχίζει να αυξάνεται (0.3177) και ο PSNR πέφτει (4.98 dB), ενώ η SSIM ανεβαίνει λίγο (0.1608), γιατί η υπερβολική κοπή οδηγεί σε σχεδόν επίπεδη εικόνα που όμως χάνει κάθε δυναμική λεπτομέρεια.

Σχολιάστε τι συμβαίνει όταν δεν γίνεται χρήση κατωφλίου:



Η οπτική σύγκριση επιβεβαιώνει : στο χαμηλό T της πρώτης στήλης (T=0) η εικόνα είναι γεμάτη θόρυβο και artefacts, στο ιδανικό T (μέση στήλη) εμφανίζονται μεν κάποιες σκόρπιες λεπτομέρειες αλλά πάλι υπάρχουν οι κηλίδες θορύβου, ενώ στο υψηλό T (δεξιά) έχουμε ομαλό, αφιλτράριστο θόλωμα χωρίς καθόλου ανάκτηση. Συμπερασματικά, το απλό αντίστροφο φίλτρο με threshold δεν βελτιώνει ποιοτικά τη θολωμένη και θορυβώδη εικόνα: ελαφρά μετρήσιμη μείωση MSE – αλλά όχι κάτω από τη θολή-θορυβώδη– συνοδεύεται από σημαντική απώλεια λεπτομέρειας και ανεπαρκή ανάκτηση δομής, γεγονός που υποδεικνύει την ανάγκη για πιο σύνθετες μεθόδους (π.χ. Wiener, τακτοποίηση Tikhonov).

Διαδικασία μέρους B:

Αρχικά βλέπουμε πώς να παγιδεύουμε τη PSF του φίλτρου με την κλασική τεχνική της σημειακής διέγερσης που μετατρέπει τη μαύρη ρουτίνα psf.p σε ένα μετρήσιμο χωρικό μοτίβο θόλωσης και, με Fourier, σε μια καθαρή φασματική συνάρτηση μεταφοράς $H(u,v)$. Στη συνέχεια, εφαρμόσαμε αυτή τη PSF σε μια δοκιμαστική εικόνα, προσθέσαμε ελεγχόμενο Gaussian θόρυβο (SNR = 10 dB) και διαπιστώσαμε ότι χωρίς κανένα είδος regularization (T=0) η διαδικασία inverse filtering καταστρέφει ολοσχερώς το σήμα, οδηγώντας σε ενίσχυση θορύβου, πολύ υψηλή MSE (>0.33) και PSNR <5 dB.

Με την εισαγωγή ενός κατωφλίου T στην αντίστροφη πράξη (truncated inverse), βρήκαμε ότι υπάρχει ένα ικανοποιητικό σημείο γύρω στα $T \approx 5 \times 10^{-3}$ όπου το MSE ελαττώνεται στο ~0.29 και ο PSNR ανεβαίνει στο ~5.4 dB· ωστόσο ακόμα κι εκεί η

ποιότητα δεν ξεπερνά την απλή θόλωση + θόρυβο, κάτι που αναδεικνύει τα όρια του καθαρού inverse filtering.

Η SSIM, τέλος, υπογραμμίζει ότι η ανακτώμενη εικόνα χάνει πολύ τη δομή της, παρά τη μικρή μείωση της MSE, καθώς οποιαδήποτε προσπάθεια ανάκτησης λεπτομέρειας συνεπάγεται ανεπιθύμητη ενίσχυση του θορύβου.

Το συγκριτικό αυτό πείραμα δείχνει με τον πιο παραστατικό τρόπο ότι, ενώ ο μαθηματικός μετασχηματισμός inverse filtering είναι απλός και καθαρός, στην πράξη σε θορυβώδες πλαίσιο αποτυγχάνει να παράγει ουσιαστικά βελτιωμένη εικόνα, γεγονός που αιτιολογεί την ανάγκη για πιο ισχυρές μεθόδους σταθεροποίησης (π.χ. Wiener filtering, Tikhonov regularization ή αλγορίθμους βάσει sparsity) προκειμένου να επιτύχουμε ρεαλιστικά αποτελέσματα αποκατάστασης.

6. Ανίχνευση Ακμών

Χρησιμοποιήστε τις μάσκες ανίχνευσης ακμών Sobel και παρουσιάστε το αποτέλεσμα της ανίχνευσης :

Στο πρώτο υποερώτημα της ανίχνευσης ακμών, η βασική ιδέα είναι να εντοπίσουμε σημεία όπου η ένταση της εικόνας μεταβάλλεται απότομα, καθώς εκεί βρίσκονται οι ακμές που οριοθετούν αντικείμενα και δομές στον διάδρομο.

Μαθηματικά, μπορούμε να θεωρήσουμε την εικόνα $I(x,y)$ ως συνεχή συνάρτηση (ή στο ψηφιακό πεδίο ως δισδιάστατο πίνακα), και η ακμή στα σημεία αυτά αντιστοιχεί στη μεγάλη τιμή της απόλυτης τιμής της παραγώγου ∇I . Αυτό υλοποιείται διακριτά μέσω του τελεστή Sobel, ο οποίος ορίζεται από δύο 3×3 μάσκες:

```
% 2. Ορισμός Sobel μάσκας
Gx = [ -1  0  +1;
       -2  0  +2;
       -1  0  +1 ];
Gy = [ -1 -2 -1;
        0  0  0;
        +1 +2 +1 ];
```

Εφαρμόζοντας σύγκλιση της εικόνας με τη μάσκα G_x , υπολογίζουμε τον διακριτό προσανατολισμό της κλίσης κατά τον άξονα x :

$$S_x(i,j) = \sum_{u=-1}^1 \sum_{v=-1}^1 G_x(u,v) I(i+u, j+v)$$

και αντίστοιχα με G_y στον άξονα y :

$$S_y(i,j) = \sum_{u=-1}^1 \sum_{v=-1}^1 G_y(u,v) I(i+u, j+v)$$

Το μέτρο της τοπικής κλίσης, που συνήθως χρησιμοποιούμε ως μέτρο έντασης ακμής, προκύπτει είτε ως ευκλείδειο μέτρο:

$$M(i, j) = \sqrt{S_x(i, j)^2 + S_y(i, j)^2}$$

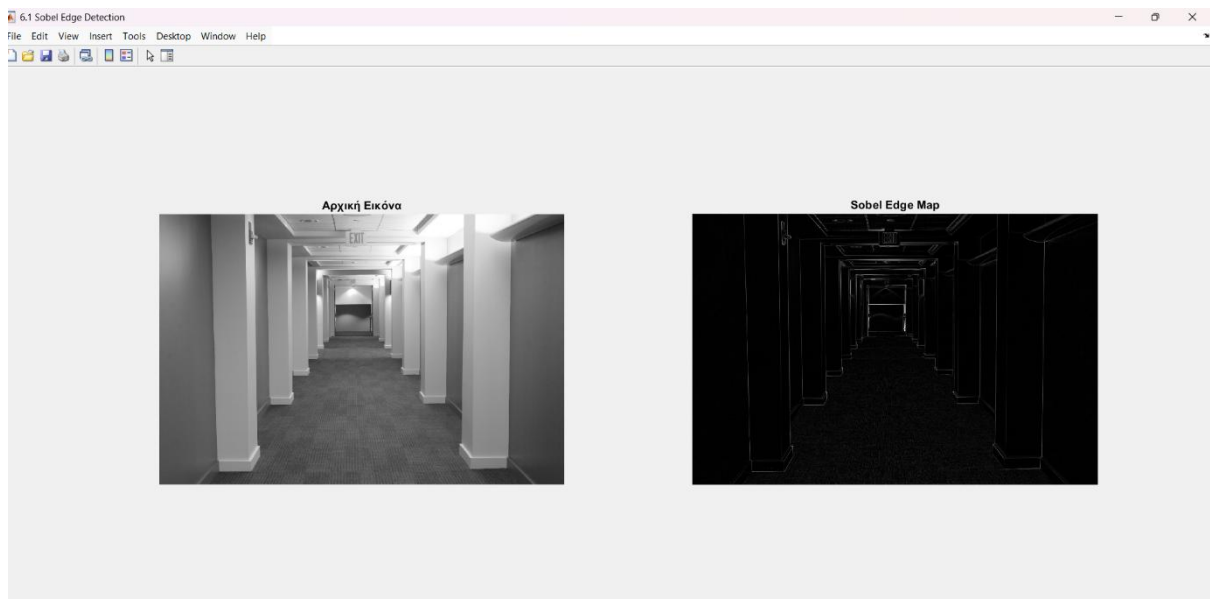
είτε ως “απλοποιημένο” άθροισμα απόλυτων τιμών για ταχύτερη υπολογιστική εκτέλεση:

$$M(i, j) \approx |S_x(i, j)| + |S_y(i, j)|$$

Η έξοδος $M(i, j)$ είναι ένας συνεχής χάρτης, όπου κάθε pixel εκφράζει πόσο ισχυρή είναι εκεί η ακμή.

Στο παράδειγμά μας με την εικόνα “hallway.png,” οι λευκές γραμμές στα σημεία υψηλής κλίσης αντιστοιχούν στα όρια των τοίχων, στα εγχοπές των θυρών και στα μοτίβα των πλακιδίων του δαπέδου, ενώ οι ομαλές περιοχές (π.χ. απρόσβλητοι τοίχοι) εμφανίζονται σκούρες.

Τέλος, παρουσιάζουμε οπτικά το αποτέλεσμα της ανίχνευσης Sobel.



Στο πάνω μέρος της εικόνας, οι γραμμές της οροφής και τα φωτιστικά σώματα ανιχνεύονται ως διακριτές κορυφές έντασης, ενώ στο κάτω μέρος του πλαισίου το χαλί του διαδρόμου εμφανίζει πιο λεπτομερείς υφές, οι οποίες ωστόσο εντοπίζονται με μικρότερη ένταση σε σχέση με τα σκληρά όρια των αρχιτεκτονικών στοιχείων. Αυτό το αποτέλεσμα είναι αποτέλεσμα του υπολογισμού έντασης $M = \sqrt{(S_x^2 + S_y^2)}$, όπου κάθε μία από τις δύο προσανατολισμένες συνιστώσες της κλίσης S_x και S_y προσδιορίζει τις οριζόντιες και τις κάθετες αλλαγές φωτεινότητας αντίστοιχα.

Η οπτική επίδραση αυτών των υπολογισμών είναι ότι όλα τα δομικά στοιχεία του τοίχου, της κάσας της πόρτας και των παραθύρων αναδεικνύονται με έντονες ακμές, ενώ οι απαλές μεταβάσεις στο τμήμα του δαπέδου απεικονίζονται σε πιο διακριτική

μορφή. Η κανονικοποίηση $M / \max(M)$ κάνει το χάρτη ακμών να επεκταθεί πλήρως στο εύρος $[0,1]$, διασφαλίζοντας ότι οι πιο ισχυρές ακμές δεν κολλάνε στην κορυφή.

2. Πραγματοποιήστε ολική κατωφλίωση στα παραπάνω αποτελέσματα δικαιολογώντας την τιμή του κατωφλίου που επιλέξατε σε κάθε περίπτωση.

```
T = graythresh(M_norm);

% Εφαρμογή του κατωφλίου στην κανονικοποιημένη εικόνα
BW = M_norm >= T;

% Εμφάνιση του αποτελέσματος της κατωφλίωσης
fprintf('Κατώφλι Otsu (M): %.4f\n', T);

figure('Name','6.2 Global Thresholding','NumberTitle','off');
subplot(1,2,1);
imshow(M_norm, []);
title('Μέτρο Κλίσης M', 'FontWeight', 'bold');

subplot(1,2,2);
imshow(BW);
title(sprintf('Edge Map με Otsu (T=%.4f)', T), 'FontWeight', 'bold');
```

Υπολογισμός κατωφλίου με τη μέθοδο Otsu , η μέθοδος Otsu είναι κατάλληλη γιατί επιλέγει αυτόματα το βέλτιστο κατώφλι που μεγιστοποιεί τη διακύμανση μεταξύ των κλάσεων στο ιστόγραμμα του κανονικοποιημένου μεγέθους κλίσης.



Όσον αφορά την παγκόσμια κατωφλίωση των αποτελεσμάτων Sobel, η χρήση της μεθόδου Otsu οδήγησε σε ένα pragmatical threshold που βασίζεται στην εκτίμηση της μέγιστης διασποράς μεταξύ των δύο κλάσεων έντασης (ακμές vs φόντο).

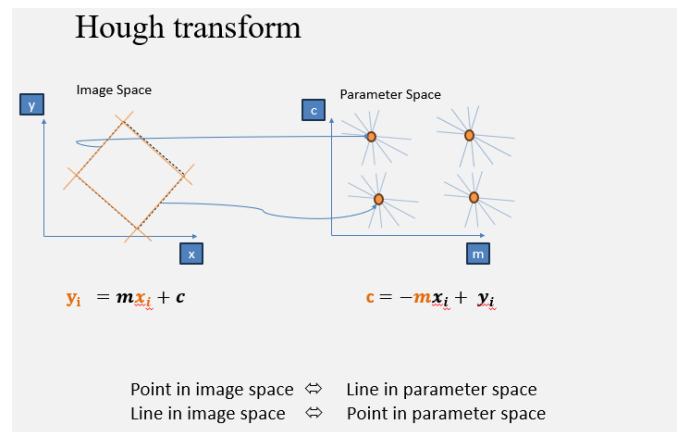
Η τιμή $T \approx 0.1255$ που προέκυψε είναι εκείνη που διαχωρίζει με τον πιο καθαρό τρόπο τα pixel με υψηλό M_norm , τα οποία αντιστοιχούν σε δομικά όρια και γραμμικές δομές, από τα pixel χαμηλής έντασης που συνθέτουν τις ομαλές ή ασαφείς περιοχές.

Το τελικό Edge Map, όταν εφαρμόζεται το $M_{norm} \geq T$, εμφανίζει ένα γραμμικό σχέδιο που αντιπροσωπεύουν αποκλειστικά τα πιο επείγοντα και σημαντικά όρια στην εικόνα του διαδρόμου.

Αυτό το output είναι ιδανικό για περαιτέρω επεξεργασία, αφού απομακρύνονται όλες οι ανεπιθύμητες μικρο-δομές που δεν φέρουν πληροφορία σημαντική για την ανίχνευση ακμών, όπως υφές χαλιού ή μικρές σκιάσεις στον τοίχο.

Bonus (Να εφαρμοστεί ο μετασχηματισμός Hough για την ανίχνευση των ευθύγραμμων τμημάτων στην εικόνα. Αποτυπώστε τα ευθύγραμμα τμήματα που ανιχνεύσατε στην αρχική εικόνα (χρησιμοποιώντας κάποιο χρώμα της επιλογής σας)).

Λογική Hough



Το Hough Transform αποτελεί ένα ισχυρό εργαλείο για τον εντοπισμό ευθειών (και γενικότερα γεωμετρικών σχημάτων) ακόμα και σε θορυβώδεις ή μερικώς κατεστραμμένες εικόνες .

Δημιουργία χάρτη ακμών για το Hough Transform, ο αλγόριθμος Canny παράγει καλύτερα αποτελέσματα από το απλό thresholding αποτέλεσμα του Sobel .

```
I_norm = double(I_uint8) / 255;
BW_hough = edge(I_norm, 'canny', [0.04 0.13]);
% Εφαρμογή του μετασχηματισμού Hough
[H, theta, rho] = hough(BW_hough);
```

Εύρεση των κορυφών στο χώρο Hough και επιλέγουμε τις 30 κορυφές με την υψηλότερη συσσώρευση και threshold 15% του μέγιστου για να αποφύγουμε ψευδείς θετικές ανιχνεύσεις

```
P = houghpeaks(H, 40, 'Threshold', 0.15*max(H(:)));
```

Εξάγουμε τα ευθύγραμμα τμήματα από τις κορυφές και θέτουμε MinLength σε 80 pixels για να αποφύγουμε μικρά τμήματα και FillGap σε 20 pixels για να συνδέσουμε κοντινά τμήματα

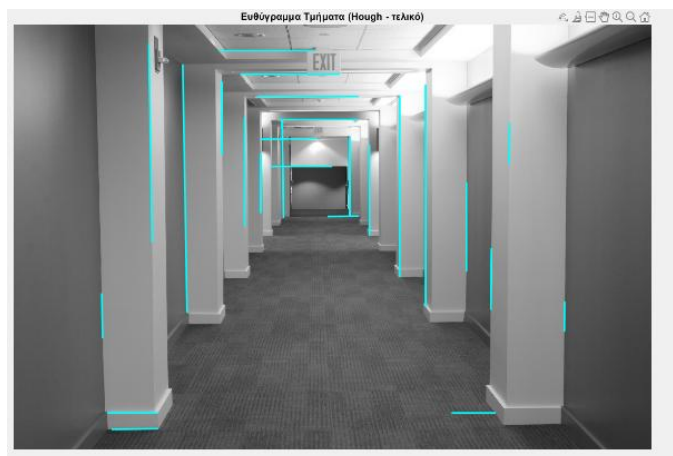
```
lines = houghlines(BW_hough, theta, rho, P, 'FillGap', 20, 'MinLength', 80);
```

Φιλτράρισμα τμημάτων: διατηρούμε μόνο τα κάθετα και οριζόντια τμήματα. Αυτό βελτώνει το αποτέλεσμα απομακρύνοντας διαγώνιες γραμμές που συχνά είναι θόρυβος

```
angTol = 12; % ανοχή σε μοίρες γύρω από τις 0° και 90°
keep = false(size(lines));

for k = 1:length(lines)
    th = abs(lines(k).theta);
    keep(k) = (th <= angTol) || (abs(th-90) <= angTol);
end

filtered_lines = lines(keep);
```



Το βήμα της εφαρμογής του μετασχηματισμού Hough για την εξαγωγή των ευθύγραμμων τμημάτων φέρνει την διαδικασία σε ένα επίπεδο πιο υψηλό, καθώς χάρη στον χώρο Hough φαίνονται οι κύριες ευθείες της σκηνής:

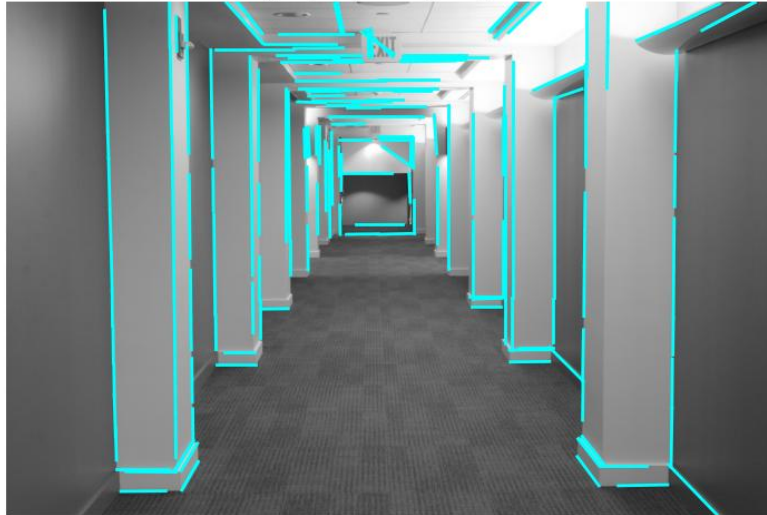
οι πλάγιες πλευρές του διαδρόμου, τα περιγράμματα των θυρών και οι γραμμές της οροφής. Μετά την εκτέλεση edge detection Canny (ακριβή αναγνώριση ακμών) οι κορυφές στον πίνακα Hough αντιστοιχούν στις πιο σταθερές και συνεχείς φωτεινότητες.

Το fillGap=20 και το minLength=80 που επιλέχθηκαν εξασφαλίζουν ότι μόνο τα μακρόστενα τμήματα θα γίνουν ορατά, ενώ τόση ανοχή στις γωνίες γύρω από τις 0° και τις 90° φιλτράρει τις διαγώνιες γραμμές, οι οποίες είναι υπολειμματικός θόρυβος. Το τελικό αποτέλεσμα αποτυπώνεται πάνω στην πρωτότυπη εικόνα με έντονο κυανό, αναδεικνύοντας τη συμμετρία του χώρου και τα επαναλαμβανόμενα στοιχεία, ενώ ταυτόχρονα διατηρείται η συνέχεια των ευθύγραμμων μεταβάσεων.

Λόγω μη ολοκληρωμένου αποτελέσματος/ ημιτελούς σε περιβάλλον matlab ,το συγκεκριμένο υποερώτημα υλοποιήθηκε ταυτόχρονα σε περιβάλλον python :

PYTHON IMPLEMENTATION

Ανιχνευμένες ευθείες Hough: 195



Στο παρόν σκριπ αρχικά φορτώνουμε την έγχρωμη εικόνα διαδρόμου και την μετατρέπουμε σε γκρι κλίμακα, ώστε να απλοποιηθεί η ανάλυση των υφών και των ακμών. Έπειτα εφαρμόζουμε το φίλτρο Canny για την εξαγωγή των σημαντικότερων ακμών, ρυθμίζοντας τη χωρική ομαλοποίηση ($\sigma=2.0$) και τα κατώφλια ($low_threshold=0.05$, $high_threshold=0.2$) έτσι ώστε να περιορίσουμε τον θόρυβο και να αναδείξουμε μόνο τις καθαρές γραμμικές δομές.

Με βάση τον χάρτη ακμών, χρησιμοποιούμε τον μετασχηματισμό Hough (`transform.probabilistic_hough_line`) για να εντοπίσουμε διακριτά ευθύγραμμα τμήματα. Τέλος, επικαλύφθηκαν όλα τα ανιχνευμένα τμήματα στην αρχική εικόνα χρησιμοποιώντας γραμμές, εμφανίζοντας συνολικά τον αριθμό τους στον τίτλο του γραφήματος, και αποθηκεύεται το αποτέλεσμα ως «Hallway_hough.png». Με αυτόν τον τρόπο αναδεικνύουμε αποτελεσματικά τις κυρίαρχες γραμμικές δομές της σκηνής, ιδανικές για περαιτέρω γεωμετρική ή τοπολογική επεξεργασία (σε περιβάλλον matlab η υλοποίηση ήταν πιο δύσκολη ενώ σε python ο εντοπισμός έγινε ορθά).