

ΜΗΧΑΝΙΚΟΙ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

Εργαστηριακές Ασκήσεις 2021

Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας

Κωνσταντοπούλου Ευαγγελία

1059560

## 1 Φιλτράρισμα στο πεδίο συχνοτήτων

Προεπεξεργασία:

Για την προεπεξεργασία της εικόνας χρησιμοποιούμε την εντολή `imadjust` για να αυξήσουμε την αντίθεση της grayscale εικόνας κάνοντας remapping τις τιμές ώστε να καλύπτουν ολόκληρη τη δυναμική περιοχή από [0 255].

Διακριτός Μετασχηματισμός Fourier 2D:

Επειδή:  $e^{2\pi i[\frac{ux}{M} + \frac{vy}{N}]} = e^{2\pi i[\frac{ux}{M}]} \cdot e^{2\pi i[\frac{vy}{N}]}$  Μπορούμε να διασπάσουμε τις εξισώσεις και να δουλέψουμε κατά γραμμές και στήλες αρχιζόντας από τις γραμμές:

$$F(u, :) = \sum_{x=0}^{M-1} [e^{-2\pi i[\frac{ux}{M}]}] \cdot f(x, :)$$

$$F(:, v) = \sum_{y=0}^{N-1} [e^{-2\pi i[\frac{vy}{N}]}] \cdot F(u, :)$$

Οπότε ο διακριτός F 2-D μπορεί να υπολογιστεί εκμεταλλευόμενοι την ιδιότητα αυτή της διαχωριστικότητας της εκθετικής συνάρτησης. Από πολυπλοκότητα  $N^4$  που θα προέκυπτε αν χρησιμοποιούσαμε τον 2D-DFT εξόρισμού, θα έχουμε πολυπλοκότητα  $N^2 \log N$

Οπότε αν έχουμε την εικόνα I (aerial.tiff) πρώτα παίρνουμε τον Διακριτό Μετασχηματισμό Fourier (ΔΜΦ) σε κάθε γραμμή και στην εικόνα F(u,:) που προκύπτει παίρνουμε τον ΔΜΦ σε κάθε στήλη για το τελικό αποτέλεσμα.

Ο συντελεστής DC του ΔΜΦ:

Είναι η τιμή  $F(0,0)$  του ΔΜΦ. Θέτοντας  $u=v=0$  ⇒

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [e^{-2\pi i[\frac{0 \cdot x}{M} + \frac{0 \cdot y}{N}]}] \cdot f(x, y) \Rightarrow$$

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [e^0] \cdot f(x, y) \Rightarrow$$

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

Δηλαδή το άθροισμα όλων των τιμών της εικόνας.

Η Μετατόπιση FFT μετασχηματίζει τον ΔΜΦ ώστε ο DC συντελεστής να είναι στο κέντρο της εικόνας. Καθώς ο ΔΜΦ είναι πίνακας μιγαδικών χρησιμοποιούμε την εντολή `abs` για να απεικονίσουμε το μέτρο του. Ο συντελεστής DC έχει τεράστια τιμή σε σχέση με τις υπόλοιπες τιμές του ΔΜΦ. Για να οπτικοποιήσουμε τον ΔΜΦ θα πάρουμε πρώτα το λογάριθμο του ΔΜΦ και στη συνέχεια θα κάνουμε οπτικοποίηση. Επειδή θα έχουμε πολύ υψηλές τιμές θα χρησιμοποιήσουμε την εντολή `mat2gray` για να τις φέρουμε από 0 έως 1.

Φιλτράρισμα της εικόνας στο πεδίο συχνοτήτων με χρήση χαμηλοπερατού φίλτρου:

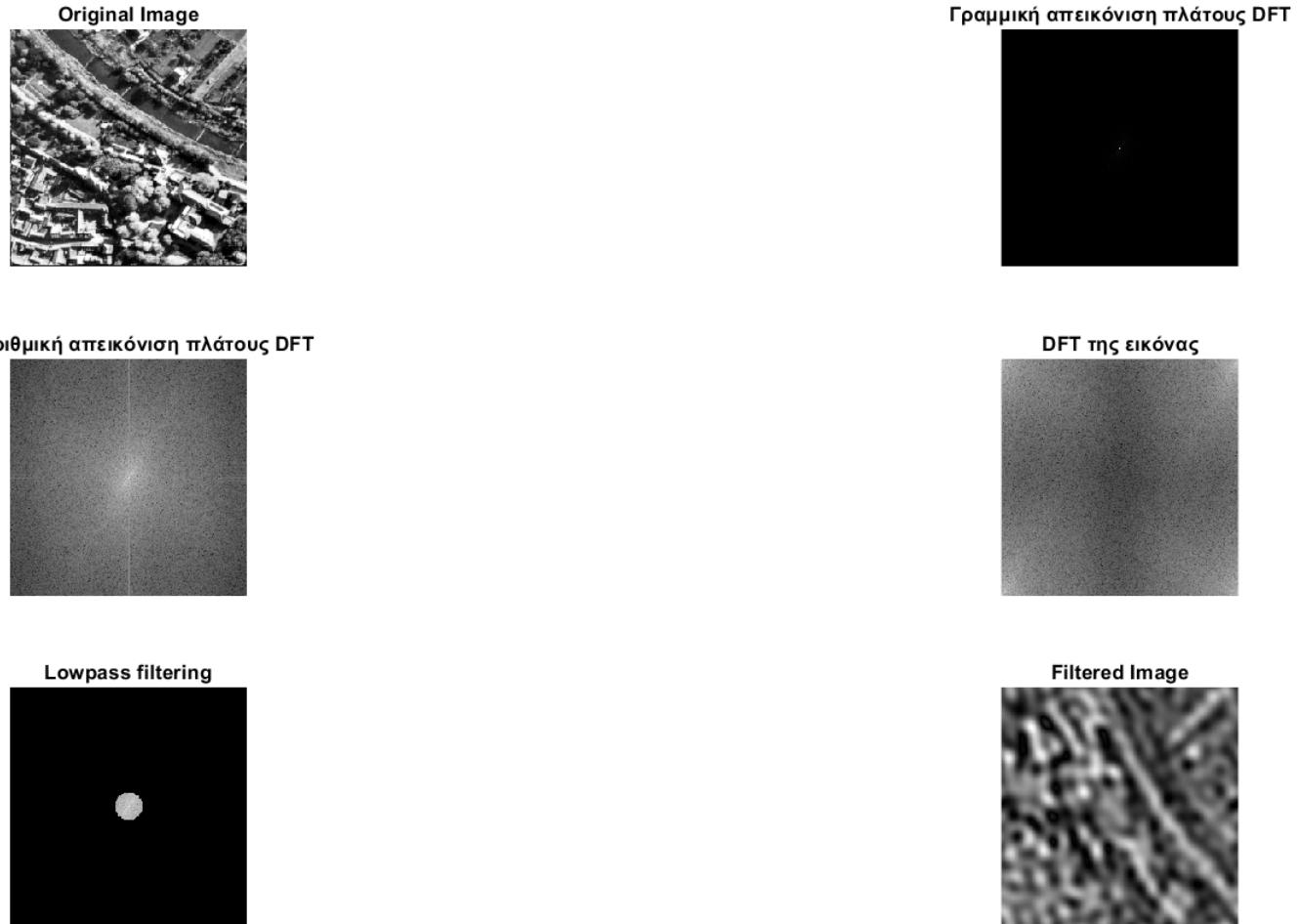
Για το φιλτράρισμα της εικόνας κάνω χρήση ιδεατού χαμηλοπερατού φίλτρου (IDLPF) με συνάρτηση μεταφοράς H της μορφής:

$$H(u, v) = \{ 1 , for D(u, v) \leq D_0 \\ 0 , for D(u, v) > D_0 \}$$

όπου  $D(u,v)$  είναι η απόσταση του σημείου με συχνότητες (u,v) από το σημείο (0,0), και  $D_0$  είναι ένας θετικός αριθμός.

Για την επιλογή του  $D_0$  υπολογίζω πόση από τη συνολική ισχύ της εικόνας θέλω να διατηρήσω. Δημιουργώ μια σφαίρα στο κέντρο της εικόνας, η οποία ανάλογα με την ακτίνα της ( $D_0$ ) μπορεί να κρατήσει συγκεκριμένες συχνότητες. Υστερα πολλαπλασιάζω την κυκλική μάσκα σ ακτίνας 15 pixel με τον πίνακα `cf` του shifted 2D-DFT που υπολόγισα προηγουμένως, ώστε να κρατήσουμε χαμηλές συχνότητες στο κέντρο του `cf`.

Αντίστροφος δισδιάστατος διακριτός μετασχηματισμός IDFT



Σχήμα 1: Αποτελέσματα μετασχηματισμών και φιλτροφίσματος της πρώτης άσκησης

Για τον υπολογισμό του IDFT κάνω χρήση της εντολής `(fft(fft(cF).').')` ώστε να χρησιμοποιήσω την εντολή της Matlab για τον 1D-DFT. Για την οπικοποίηση του IDFT ακολουθώ την ίδια διαδικασία με προηγουμένως. Παίρνω δηλαδή τις απόλυτες τιμές της `IcF` και μετά χρησιμοποιώ το `mat2gray` της Matlab για να μετασχηματιστούν όλες οι τιμές από 0 εως 1.

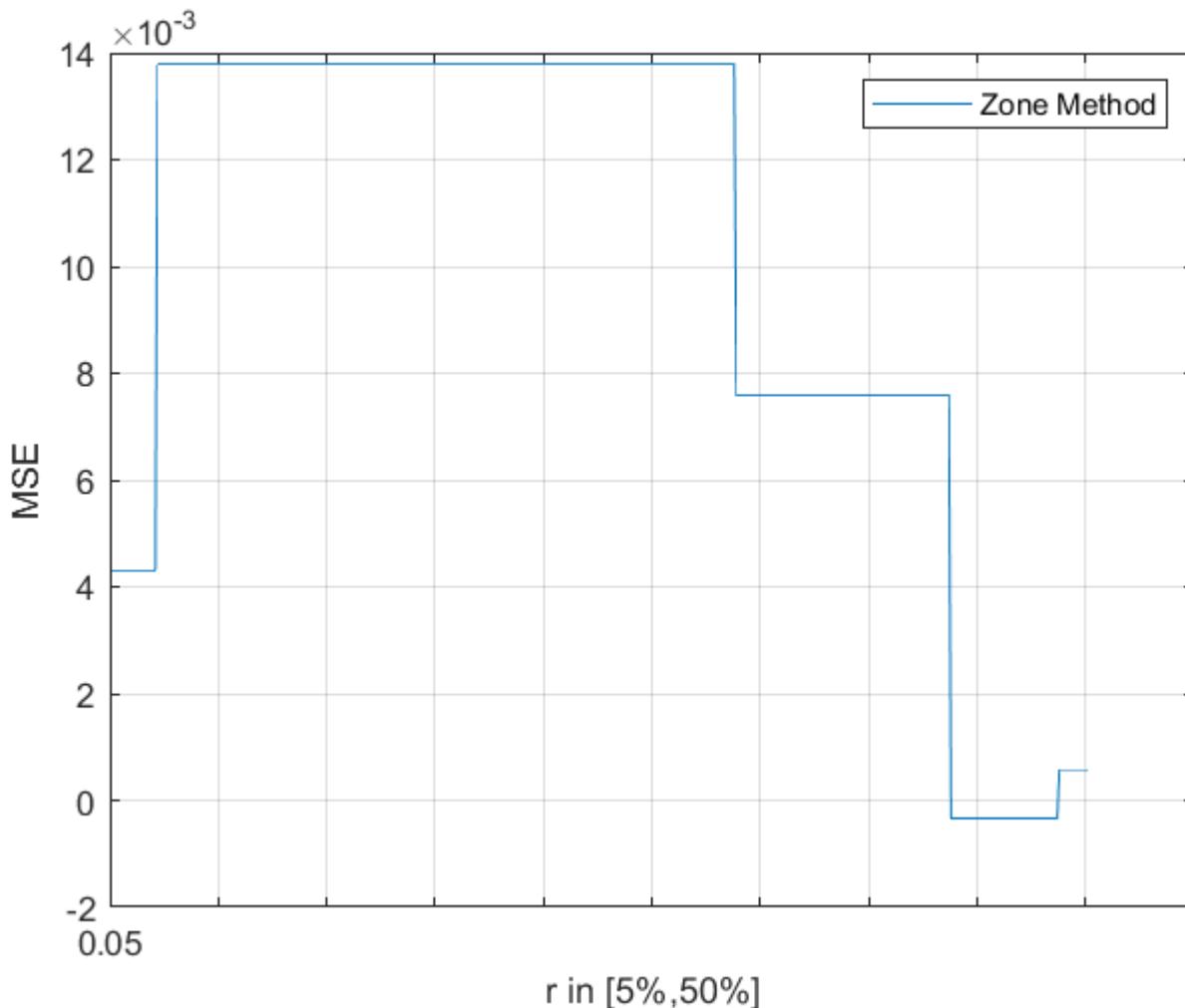
## 2 Συμπίεση Εικόνας με χρήση μετασχηματισμού **DCT**

### Μέθοδος ζώνης

Αυτή η μέθοδος κωδικοποίησης βασίζεται στην παραδοχή ότι οι συντελεστές με τη μεγαλύτερη διασπορά σε κάθε μπλοκ μεταφέρουν το μεγαλύτερο ποσοστό οπτικής πληροφορίας και επομένως είναι αυτοί που θα πρέπει να διατηρηθούν για κωδικοποίηση, ενώ οι υπόλοιποι πρέπει να μηδενιστούν. Χωρίζω την εικόνα σε μη επικαλυπτόμενες περιοχές διαστάσεων  $32 \times 32$ , και εφαρμόζω το μετασχηματισμό 2D-DCT σε κάθε μία περιοχή για να βρω τους συντελεστές του μετασχηματισμού.

Τυπολογίζω τη διασπορά των συντελεστών και ύστερα κρατάω μόνο αυτούς που είναι μεγαλύτεροι από ένα ποσοστό πληροφορίας  $r$ . Οι τιμές που μπορεί να πάρει στην άσκηση μας είναι από 0.05 έως 0.5. Υστερα εφαρμόζω αντίστροφο μετασχηματισμό 2D-DCT για να ανακτήσω τη συμπιεσμένη εικόνα.

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

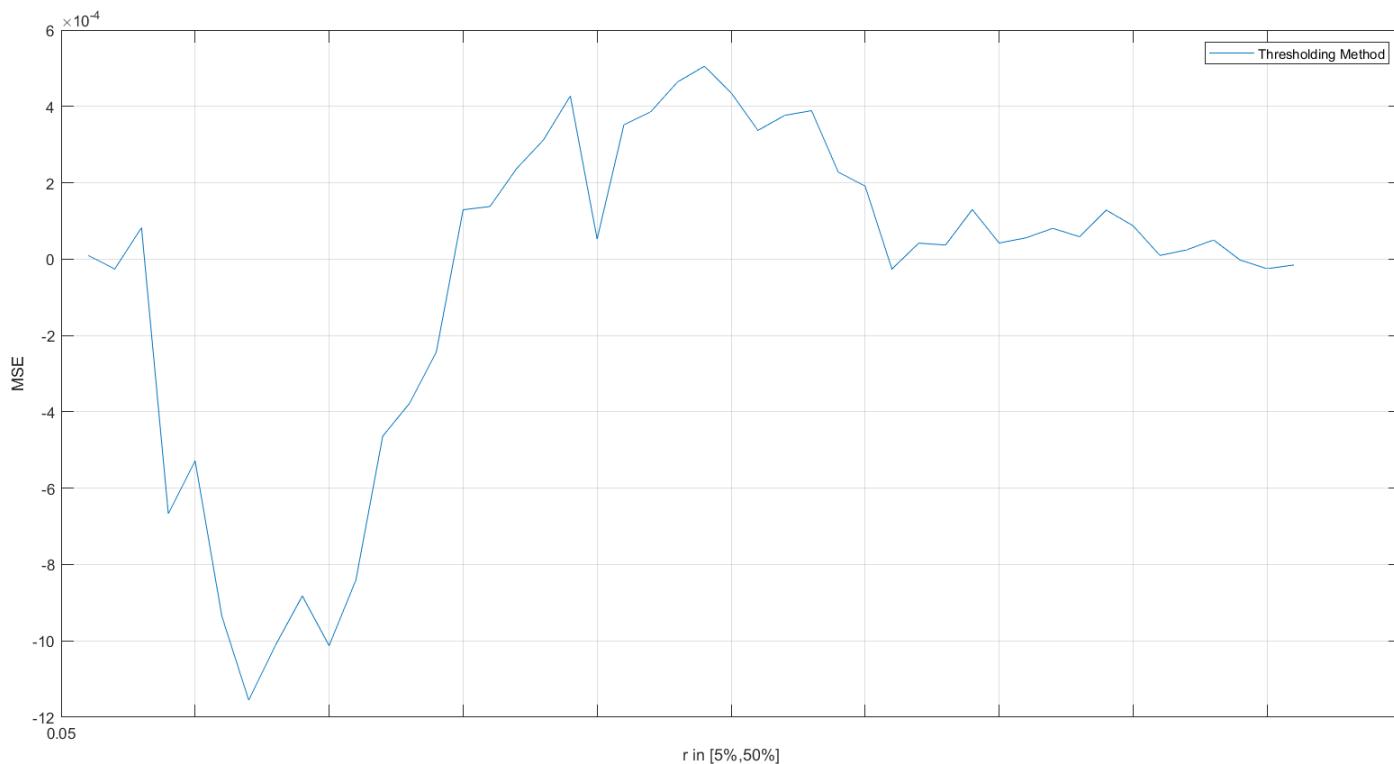


## Μέθοδος κατωφλίου

Χωρίζω την εικόνα σε μη επικαλυπτόμενες περιοχές διαστάσεων  $32 \times 32$ , και εφαρμόζω το μετασχηματισμό 2D-DCT σε κάθε μία περιοχή για να βρω τους συντελεστές του μετασχηματισμού.

Με αυτή τη μέθοδο, σε κάθε μπλοκ εφαρμόζεται ένα επίπεδο κατωφλίου το οποίο διατηρεί τους περισσότερο σημαντικούς συντελεστές. Εδώ μας ενδιαφέρει να χρατήσουμε τους συντελεστές με το μεγαλύτερο magnitude και όχι variance. Έπειτα εφαρμόζω αντίστροφο μετασχηματισμό 2D-DCT για να ανακτήσω τη συμπιεσμένη εικόνα.

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



### 3 Φιλτράρισμα στο πεδίο συχνοτήτων

#### **Moving Average Filter:**

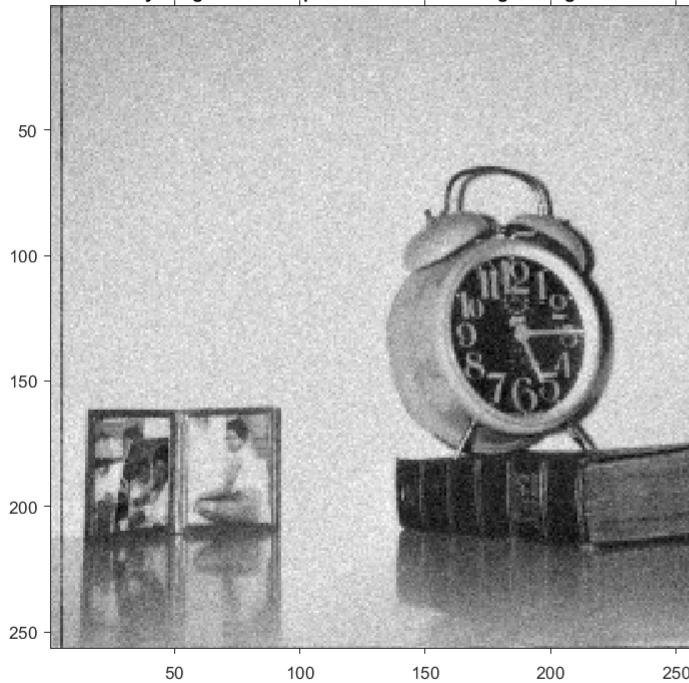
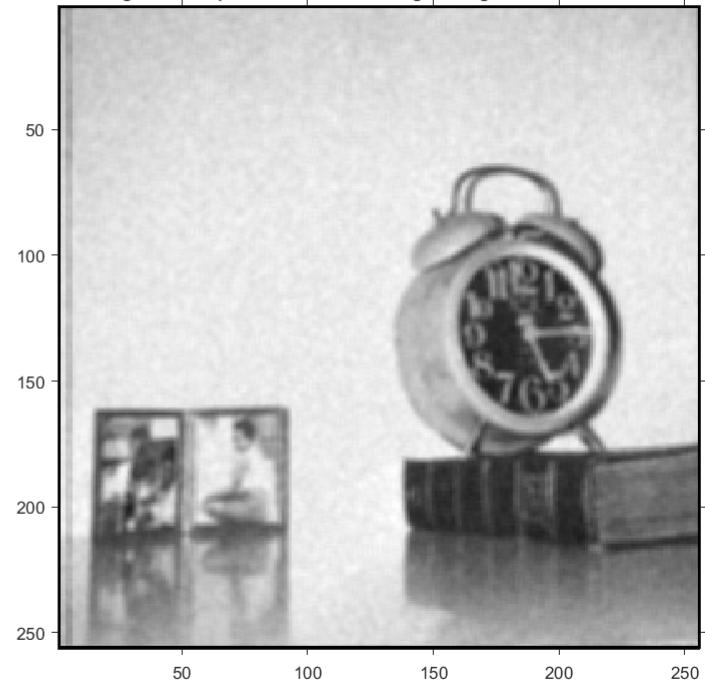
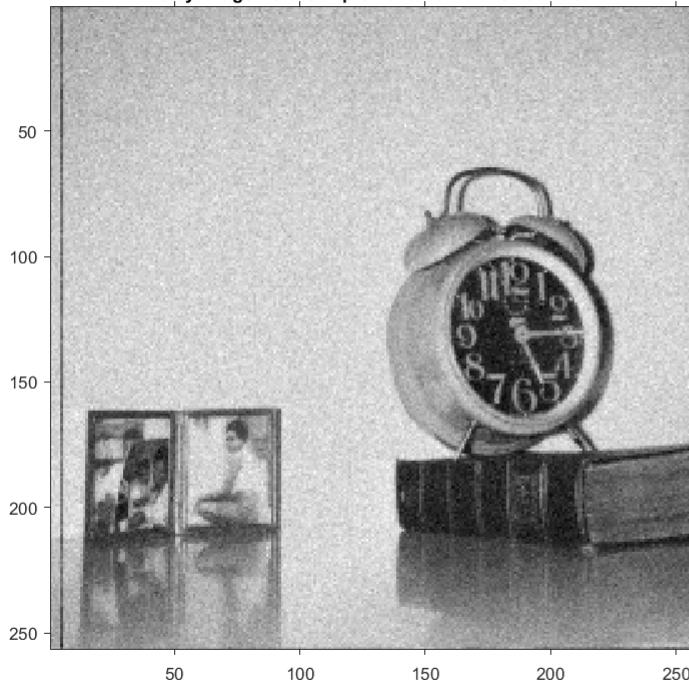
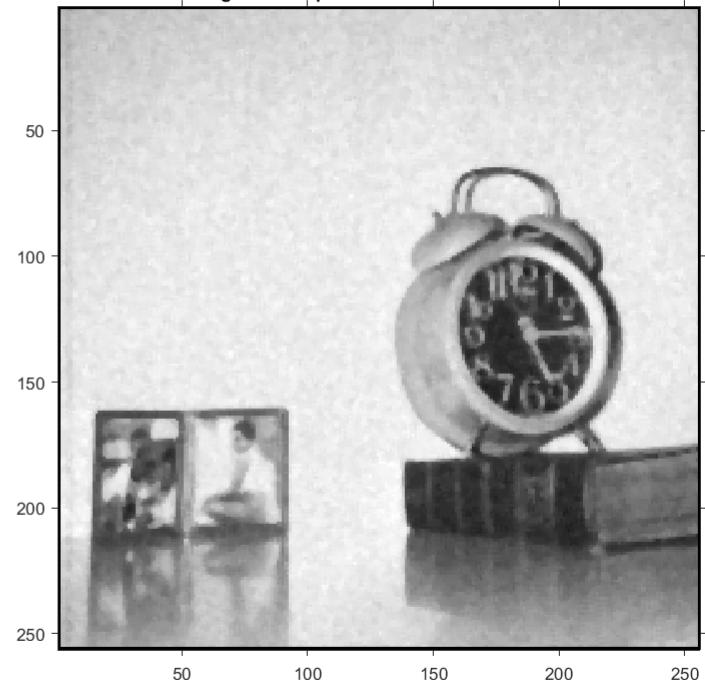
Για την υλοποίηση του Moving Average Filter, θα κάνω συνέλιξη με τη χρήση ενός απλού kernel που εξαρτάται από το μέγεθος του παραθύρου που έχω επιλέξει. Αυτό γίνεται διότι το Moving Average Filter είναι η συνέλιξη του σήματος εισόδου με έναν μοναδιαίο τετραγωνικό παλμό. Επιλέγω την ομάδα σημείων από το σήμα εισόδου συμμετρικά γύρω από το σημείο εξόδου, κάτι το οποίο προϋποθέτει το μέγεθος του παραθύρου να είναι περιττός αριθμός. Μετά από δοκιμές κατέληξα στη διαπέραση της εικόνας πρώτα μέσω γραφμών και ύστερα στηλών καθώς αυτή είναι η πιο γρήγορη προσέγγιση.

#### **Median Filter:**

Για την υλοποίηση του μεσαίου φίλτρου διατρέχω το σήμα εισόδου "entry by entry", αντικαθιστώντας κάθε πίζελ με τη μέση τιμή των γειτονικών. Το μοτίβο των γειτόνων ονομάζεται "παράθυρο", το οποίο ολισθαίνει μέχρις ότου να διατρέξουμε όλο το σήμα.

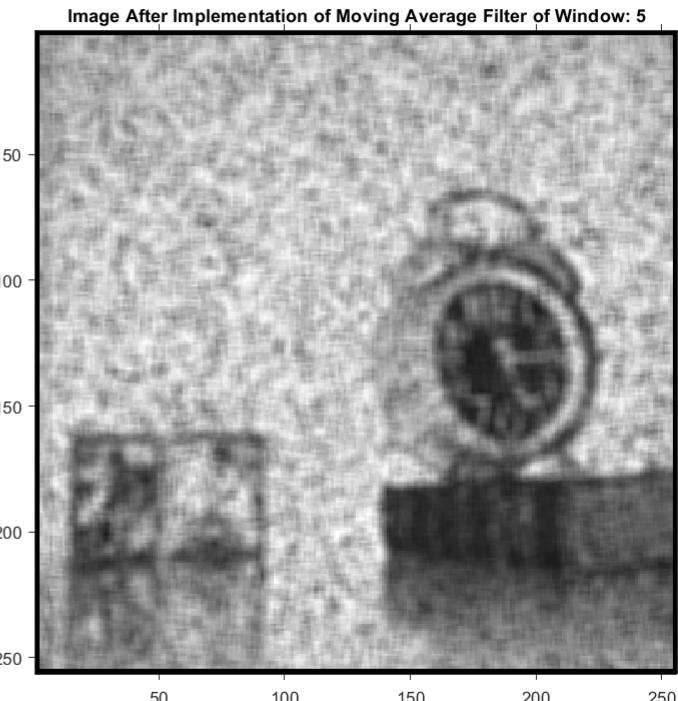
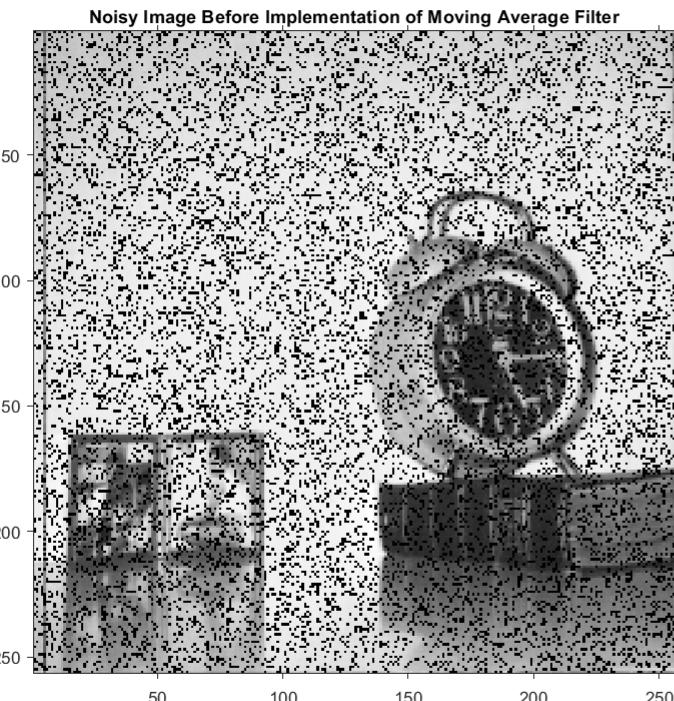
#### **3.1 Zero-mean Gaussian Additive Noise**

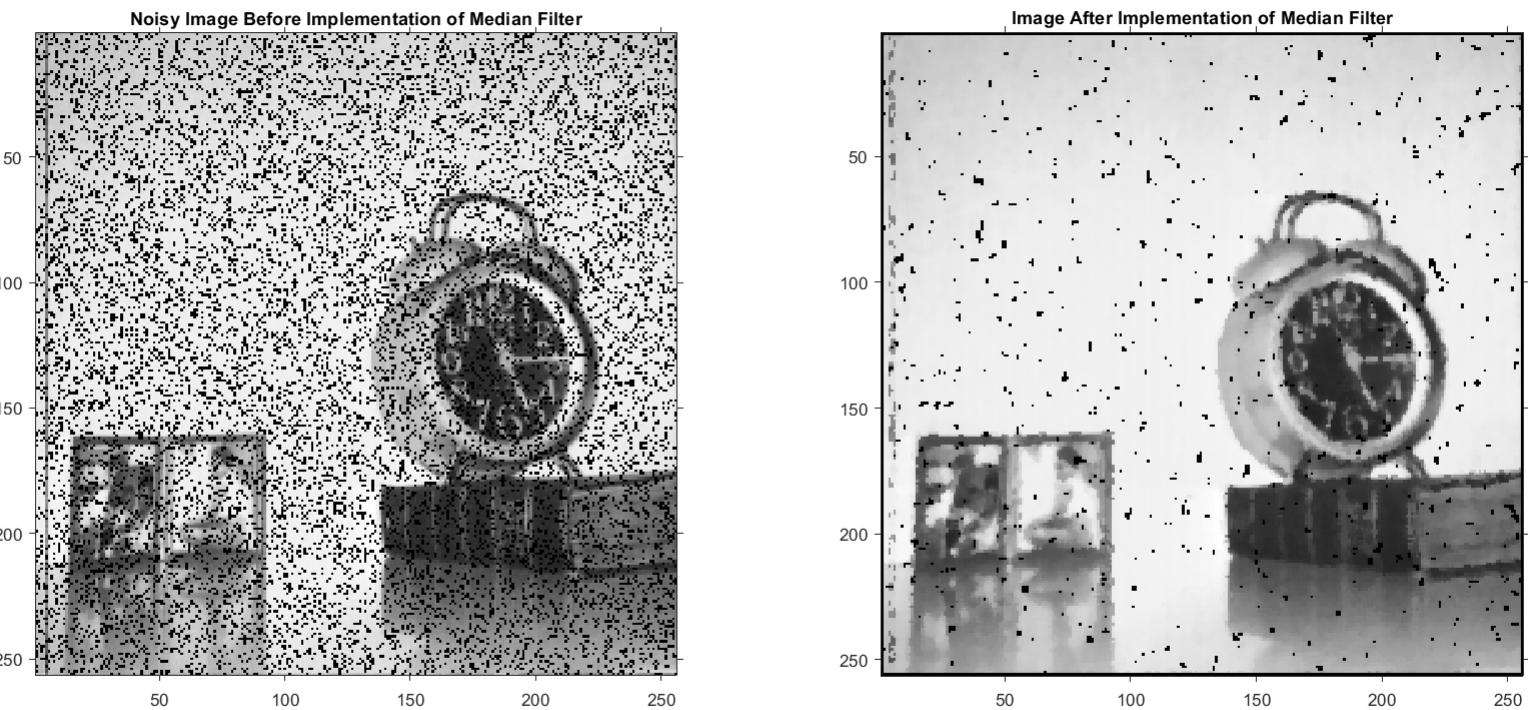
Τα αποτελέσματα της εφαρμογής των φίλτρων στην εικόνα με λευκό Gaussian θόρυβο φαίνονται παρακάτω. Και τα δύο φίλτρα έχουν ως αποτέλεσμα μια θιλωμένη εικόνα, βέβαια το moving average φίλτρο έχει το θετικό του ότι μπορούμε να ορίσουμε το παράθυρο έτσι ώστε να έχουμε το αποτέλεσμα που θέλουμε.

**Noisy Image Before Implementation of Moving Average Filter****Image After Implementation of Moving Average Filter of Window: 3****Noisy Image Before Implementation of Median Filter****Image After Implementation of Median Filter**

### 3.2 Εισαγωγή χρουστικού θορύβου

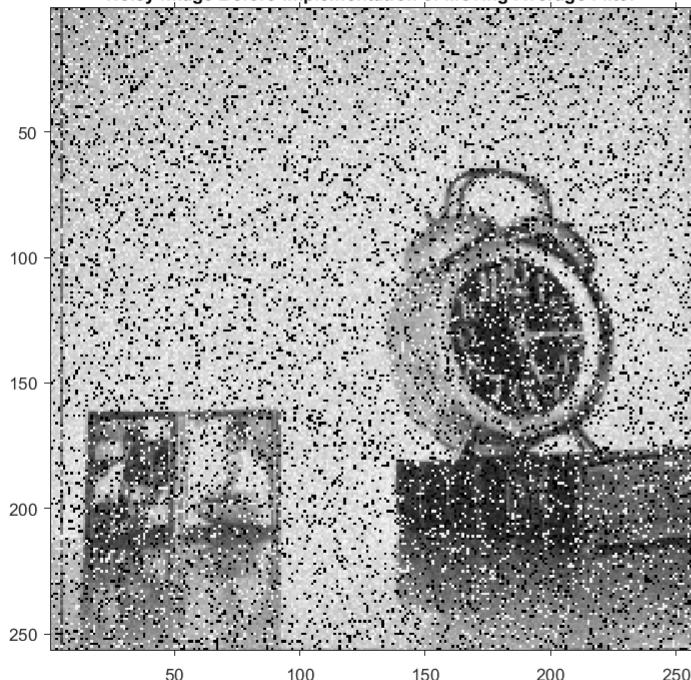
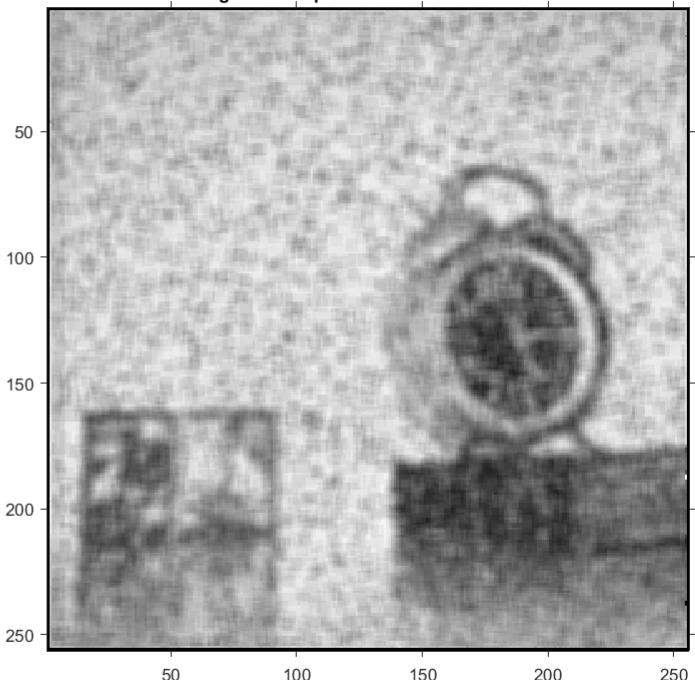
Τα αποτελέσματα της εφαρμογής των φίλτρων στην εικόνα με χρουστικό θόρυβο φαίνονται παρακάτω. Παρατηρούμε πως το φίλτρο μεσαίου είναι πολύ αποτελεσματικό στην απομάκρυνση τέτοιου ειδους θορύβου. Αυτό είναι λογικό αν σκεφτούμε ότι το φίλτρο κινούμενου μέσου πάίρνει υπ' όψιν του όλες τις τιμές της εικόνας ενώ το φίλτρο μέσου μπορεί από τη γειτονιά του να εξαλείψει τα προβληματικά πίξελ.





### 3.3 Εισαγωγή **gaussian** και χρουστικού θορύβου

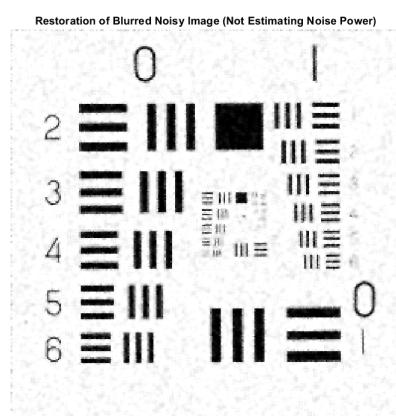
Για την απομάκρυνση και των δύο θορύβων, πρώτα εφαρμόζω το φίλτρο μέσου για την αποδοτική απομάκρυνση του χρουστικού θόρυβου και ύστερα το φίλτρο κινούμενου μέσου για το 'denoising' της εικόνας.

**Noisy Image Before Implementation of Moving Average Filter****Image After Implementation of both filters**

## 4 Αποκατάσταση Εικόνας – Αποσυνέλιξη

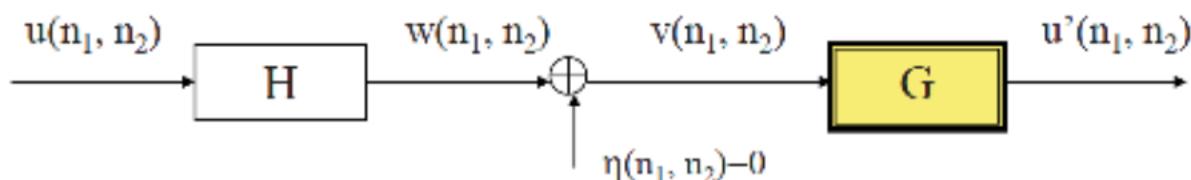
### Μέρος Α

Η αποσυνέλιξη Wiener είναι αποτελεσματική όταν τα χαρακτηριστικά του θορύβου (Ισχύς) είναι γνωστά. Στη δεύτερη προσπάθεια επιχειρώ να επαναφέρω τη θολή θορυβώδη εικόνα χρησιμοποιώντας το φίλτρο Wiener χωρίς να παρέχω εκτίμηση θορύβου. Επειδή η εικόνα εισόδου έχει υποβαθμιστεί από σταθερό θόρυβο πρόσθετου ισχύος, το wiener2 χρησιμοποιεί μια προσαρμοσμένη μέθοδο Wiener pixelwise βάσει στατιστικών που υπολογίζονται από μια τοπική γειτονιά του κάθε pixel. Από προεπιλογή, το φίλτρο αποκατάστασης Wiener υπονήντει ότι το NSR είναι ίσο με 0. Στην περίπτωση αυτή, το φίλτρο αποκατάστασης Wiener είναι ισοδύναμο με ένα ιδανικό αντίστροφο φίλτρο, το οποίο μπορεί να είναι εξαιρετικά ευαίσθητο στον θόρυβο στην εικόνα εισόδου.

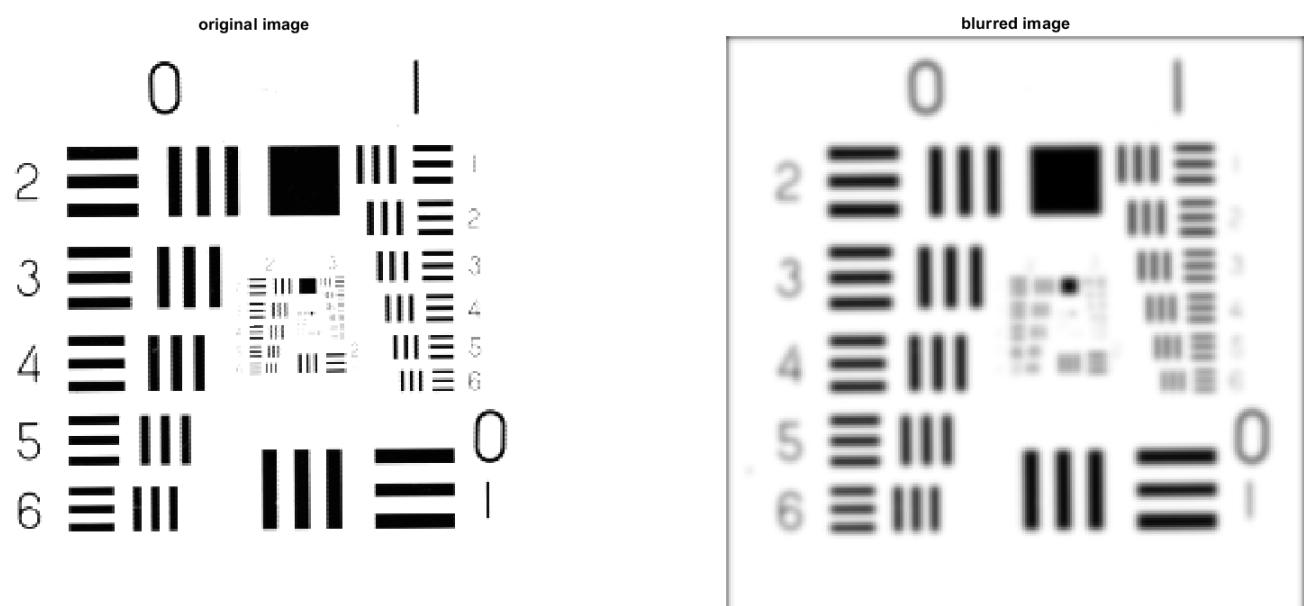


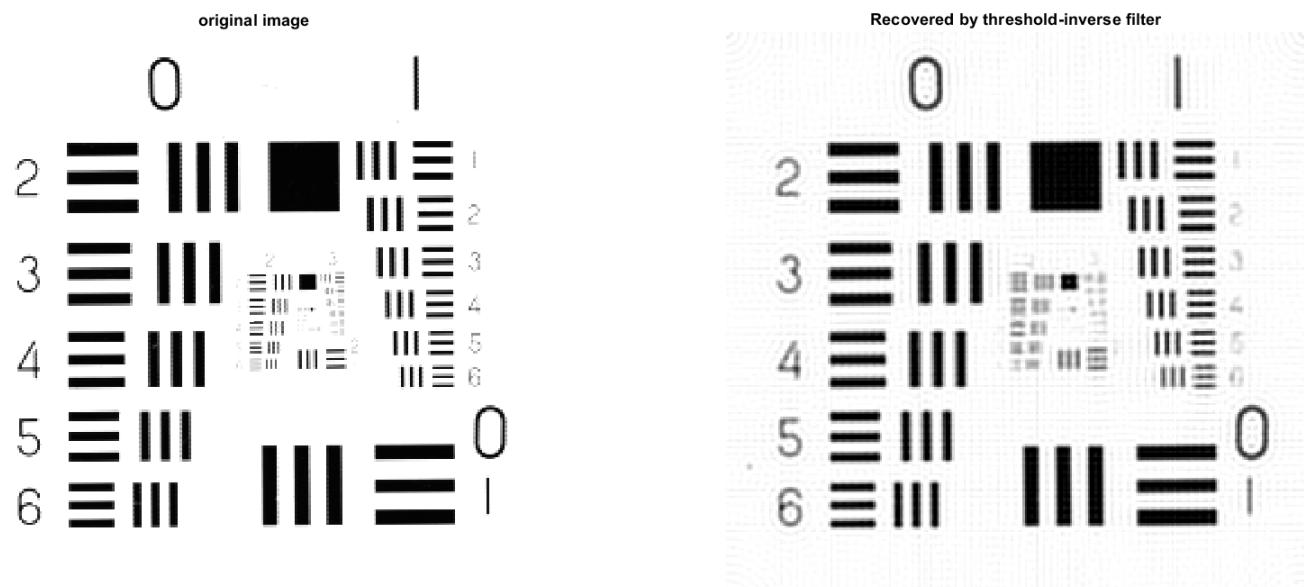
## Μέρος Β

Η αποκατάσταση της εικόνας μπορεί να γίνει χρησιμοποιώντας ένα άλλο χωρικό αμετάβλητο φίλτρο που επιβάλλεται στην εικόνα blurred, έτσι ώστε η συνάρτηση διασποράς να είναι  $h(m, n) = \delta(m, n)$ . Το αντίστροφο φιλτράρισμα χρησιμοποιεί την ιδέα ότι η κρουστική απόχριση του φίλτρου θολώματος και ξεθολώματος θα πρέπει να είναι  $\delta(m, n)$ , και έτσι έχουμε  $G(w_1, w_2) = 1/H(w_1, w_2)$ .

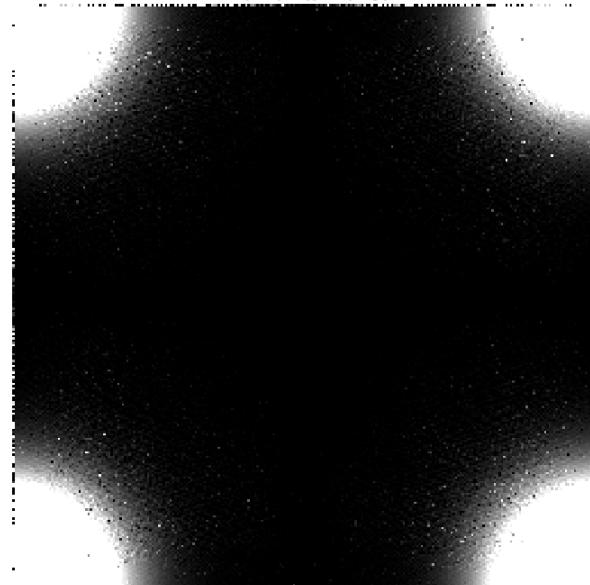


Σχήμα 2: Μοντέλο ξεθολώματος εικόνας



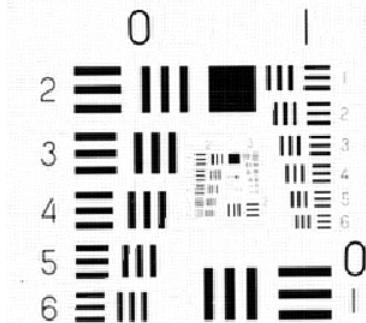


Απόκριση συχνότητας psf

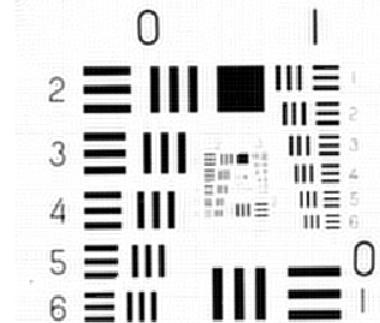


Παρατηρούμε ότι, όσο μεγαλύτερο είναι το κατώφλι, τόσο χειρότερο είναι το φαινόμενο ξεθολώματος, κάτι το οποίο φαίνεται και στο μέσο τετραγωνικό σφάλμα. Μεγαλύτερο κατώφλι σημαίνει μεγαλύτερη απόκλιση από το πραγματικό αντίστροφο.

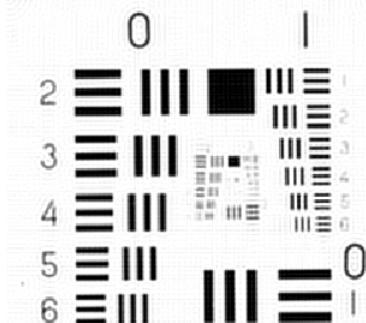
Threshold of Pseudo-inverse filtering = , mse =0.01



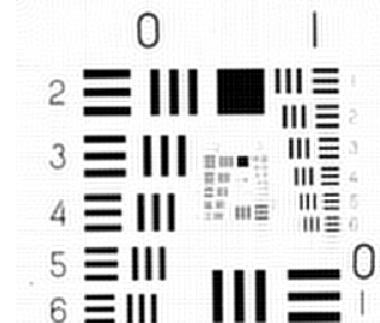
Threshold of Pseudo-inverse filtering = , mse =0.06



Threshold of Pseudo-inverse filtering = , mse =0.11



Threshold of Pseudo-inverse filtering = , mse =0.16



Αν και θεωρητικά το αντίστροφο φιλτράρισμα χωρίς κατώφλι θα μπορούσε να χειριστεί τέλεια το πρόβλημα της αποκατάστασης της εικόνας, θα δημιουργηθεί πρόβλημα όταν υπάρχει κάποιος θόρυβος στην εικόνα παραμόρφωσης. Από το  $G(w_1, w_2) = 1/H(w_1, w_2)$ , εάν το  $H$  παίρνει πολύ μικρές τιμές σε κάποια συχνότητα, για να αντισταθμίσει αυτό το φαινόμενο καταστολής το  $G$  θα πάρει μεγάλες τιμές. Αυτό θα μπορούσε να είναι ένα μεγάλο πρόβλημα όταν υπάρχει κάποιος θόρυβος, καθώς θα τον μεγεθύνει.

## 5 Βελτίωση Εικόνας – Εξίσωση Ιστογράμματος

Για τη βελτίωση της έγχρωμης εικόνας κάνουμε ολική και τοπική εξίσωση ιστογράμματος με τις ανάλογες συναρτήσεις της Matlab. Η εξίσωση ιστογράμματος μπορεί να θεωρηθεί ως η αναδιανομή της έντασης της εικόνας. Για να την εφαρμόσουμε στο σύστημα HSI μετατρέπουμε την εικόνα σε Hue, Saturation, και Intensity μορφή και κάνουμε εξίσωση ιστογράμματος μόνο στο μητρώο της έντασης. Κάνοντας πειραματισμούς ανακαλύπτουμε ότι το παράθυρο 8x8 δίνει ικανοποιητικά αποτελέσματα.

Image Before RGB Histogram Equalization



Image After RGB Histogram Equalization



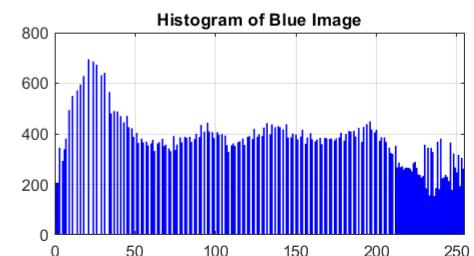
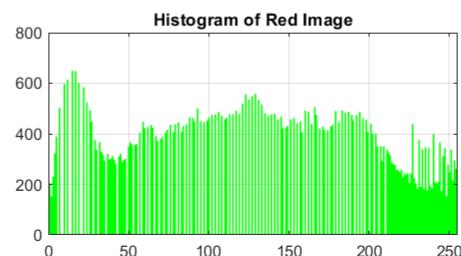
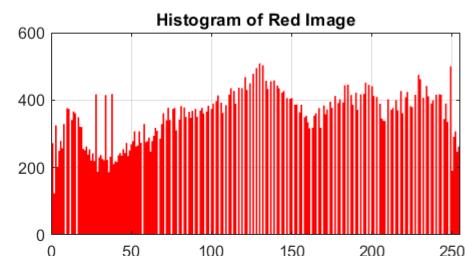
Histogram equalization of Red Image



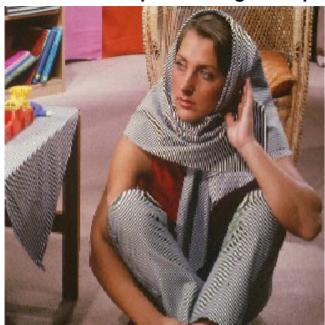
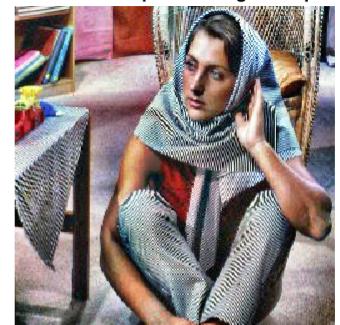
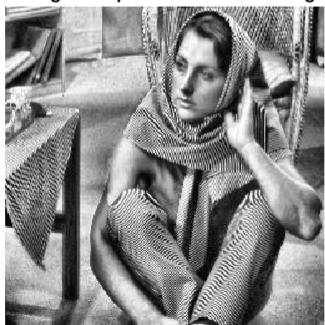
Histogram equalization of Green Image



Histogram equalization of Blue Image



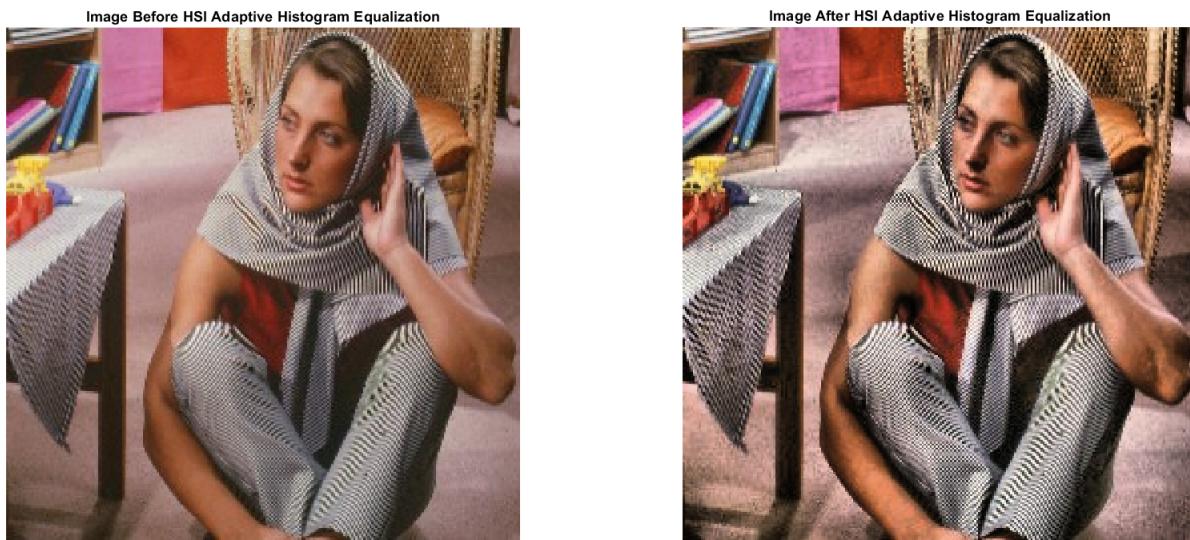
Σχήμα 3: Ολική εξίσωση ιστογράμματος στο σύστημα RGB

**Image Before RGB Adaptive Histogram Equalization****Image After RGB Adaptive Histogram Equalization****Histogram equalization of Red Image****Histogram equalization of Green Image****Histogram equalization of Blue Image**

Σχήμα 4: Τοπική εξίσωση ιστογράμματος στο σύστημα RGB

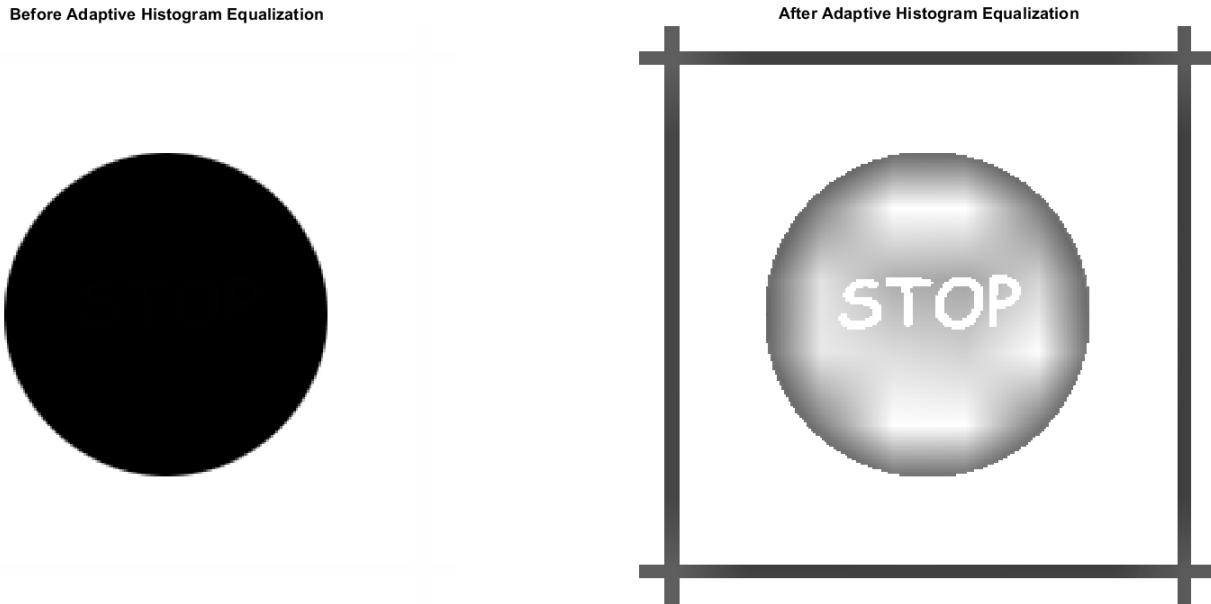


Σχήμα 5: Ολική εξίσωση ιστογράμματος στο σύστημα HSI



Σχήμα 6: Τοπική εξίσωση ιστογράμματος στο σύστημα HSI

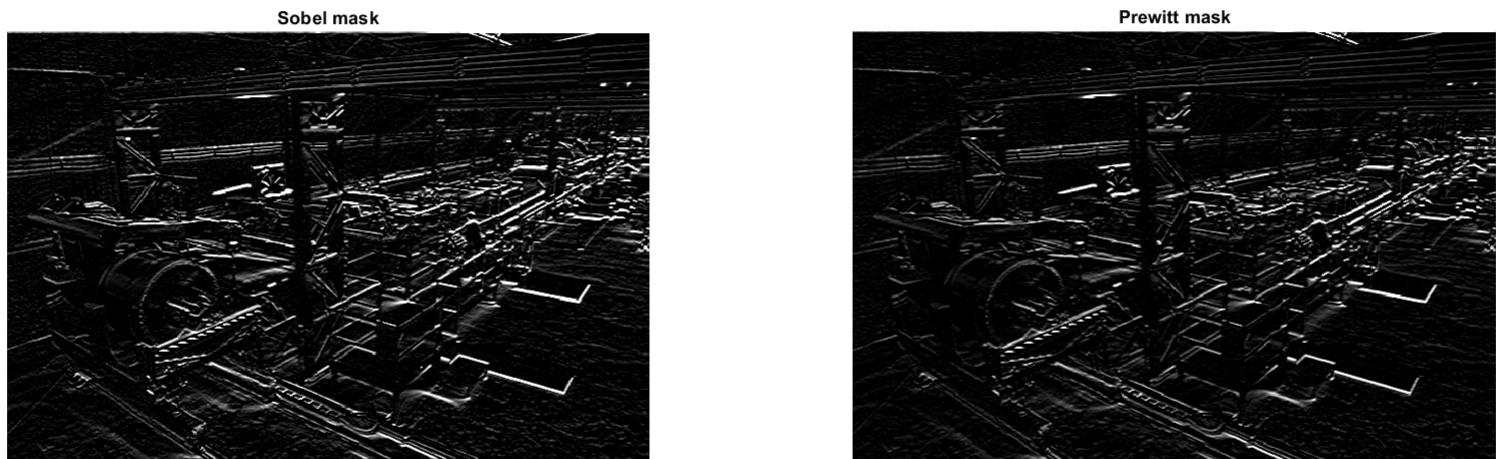
Η circle.tif είναι μια λευκή εικόνα με ένα φαινομενικά μαύρο κύκλο στη μέση. Τα grayscales είναι σε πολύ υψηλά επίπεδα επομένως δε μπορούμε να διαχρίνουμε τις λεπτομέρειες της εικόνας. Αν στην εικόνα αυτή εφαρμόζαμε ολική εξίσωση ιστογράμματος δε θα επιτυγχάναμε καλά αποτελέσματα διότι η εικόνα αυτή 'κυριαρχείται' από τα λευκά. Εάν όμως εφαρμόσω μια τοπική εξίσωση ιστογράμματος αποκαλύπτω χρυμμένες λεπτομέρειες της εικόνας όπως φαίνεται στο παρακάτω σχήμα:



Σχήμα 7: Τοπική εξίσωση ιστογράμματος

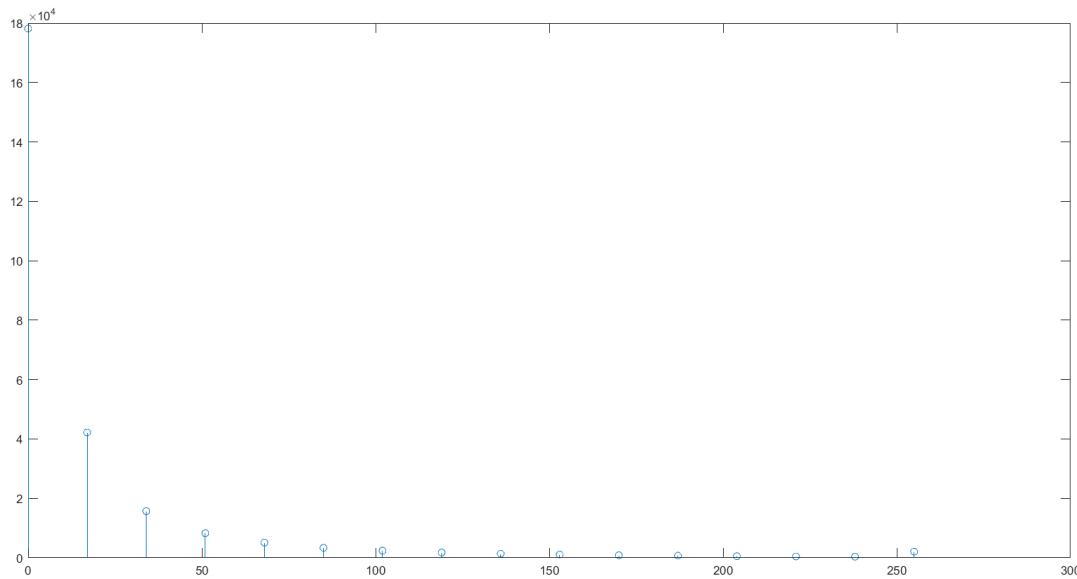
## 6 Ανίχνευση Ακμών - Κατάτμηση

Για την ανίχνευση ακμών χρησιμοποιώ δύο μάσκες, τη Sobel και τη Prewitt. Και στις δύο μεθόδους γίνεται χρήση πρώτων παραγώγων.

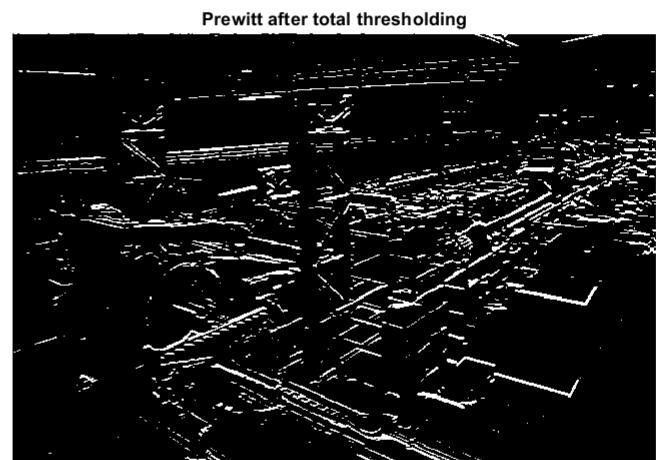


Σχήμα 8: Αποτέλεσμα ανίχνευσης ακμών

Για τον υπολογισμό του κατωφλίου στο δεύτερο ερώτημα χρησιμοποιώ τη μέθοδο Otsu. Πρώτα υπολογίζω το ιστόγραμμα της εικόνας για να το εισάγω ως παράμετρο στη συνάρτηση otsu\_thresh. Η μέθοδος του Otsu επιλέγει μέσα από δοκιμές το κατάλληλο κατώφλι που ελαχιστοποιεί την intra-class variance των κατωφλιών μένων ασπρόμαυρων πιζελ. Χρησιμοποιώ το καθολικό κατώφλι T με την εντολή imbinarize για να μετατρέψω μια grayscale εικόνα σε δυαδική εικόνα.



Σχήμα 9: Το ιστόγραμμα της εικόνας μετά τη μάσκα Sobel



Σχήμα 10: Αποτέλεσμα ολικής κατωφλίωσης