

Εργασία 2 Ψηφιακή Επεξεργασία Εικόνας 2018

Μελεζιάδης Ιωάννης

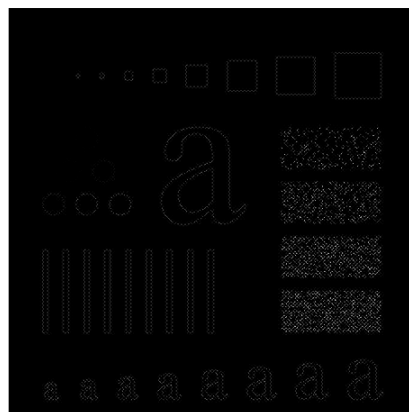
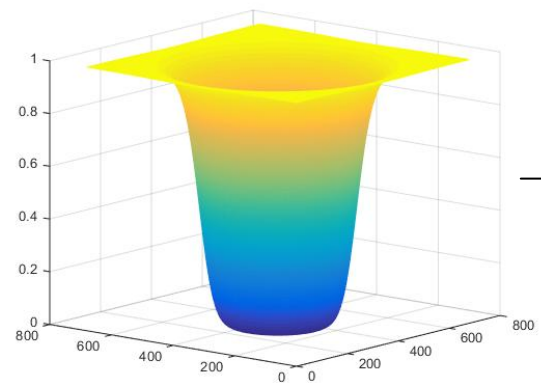
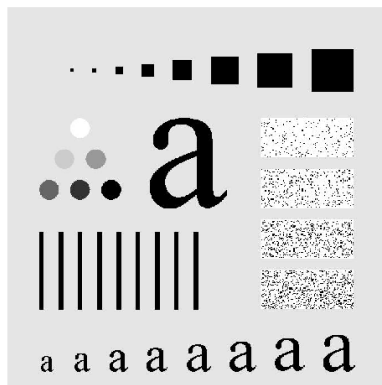
AEM 8760

Ενότητα 1

Χρησιμοποιώντας την εντολή

```
imOut = myFiltFreq(demo1Im, someFreqFilt)
```

Προέκυψε το εξής αποτέλεσμα :

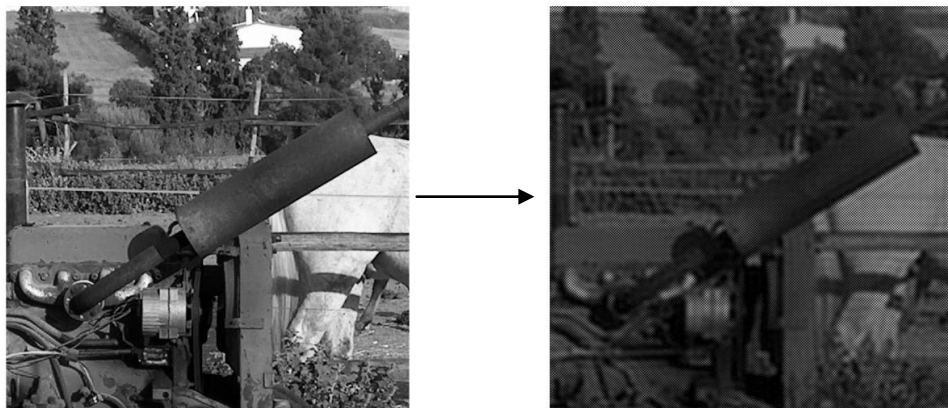
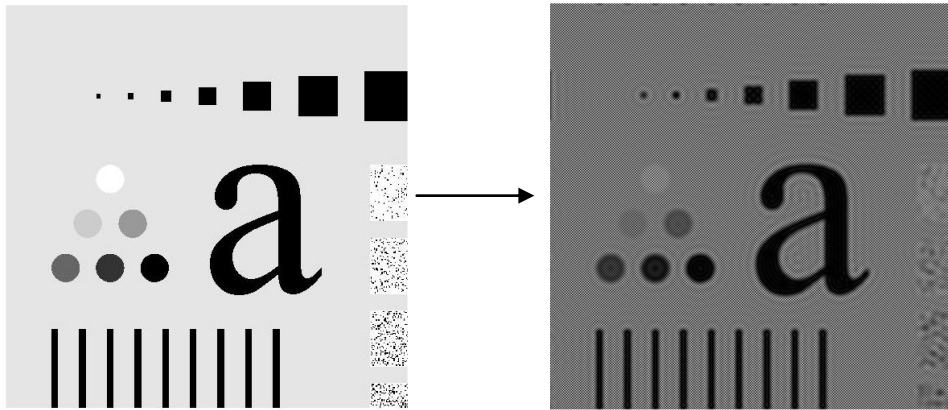


Όπου φαίνεται το αποτέλεσμα της δράσης του φίλτρου `someFreqFilt` στην εικόνα `demo1Im` και αναγνωρίζεται με βάση τη φιλτραρισμένη εικόνα σαν υπερπυκνό φίλτρο.

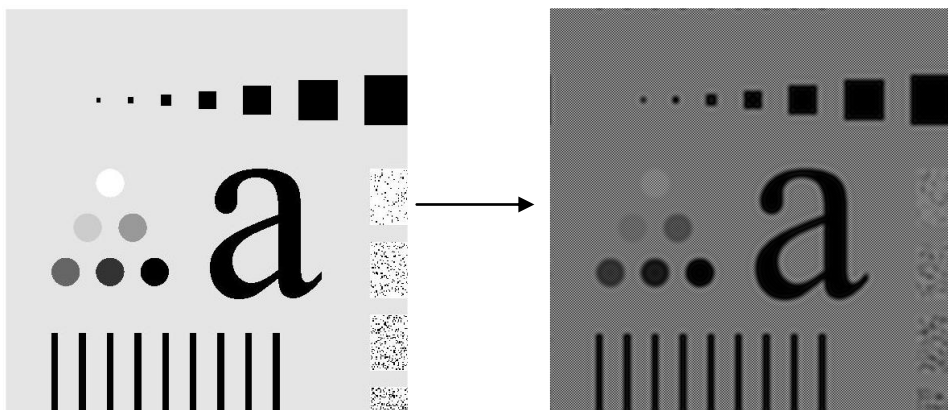
Ενότητα 2

Lowpass filters

Lowpass ideal filter:

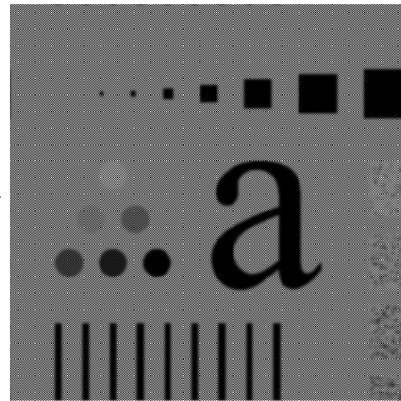
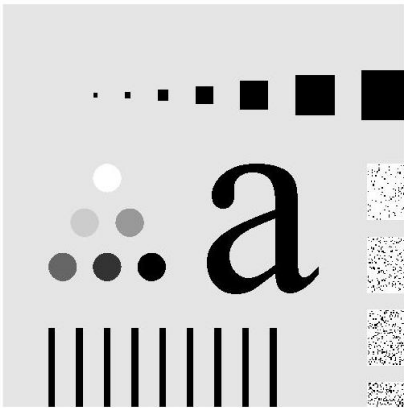


Lowpass Butterworth filter :



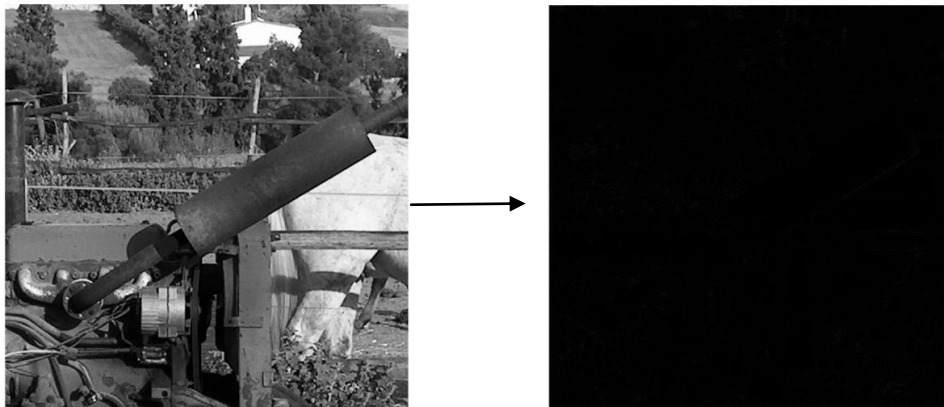
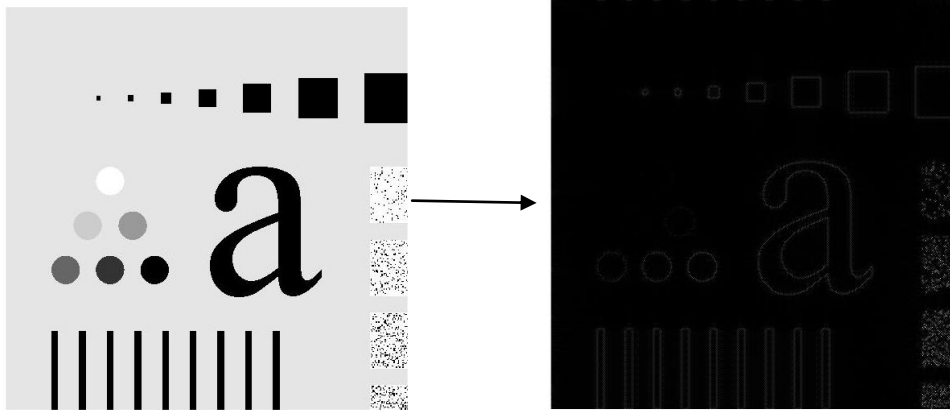


Lowpass Gauss filter :

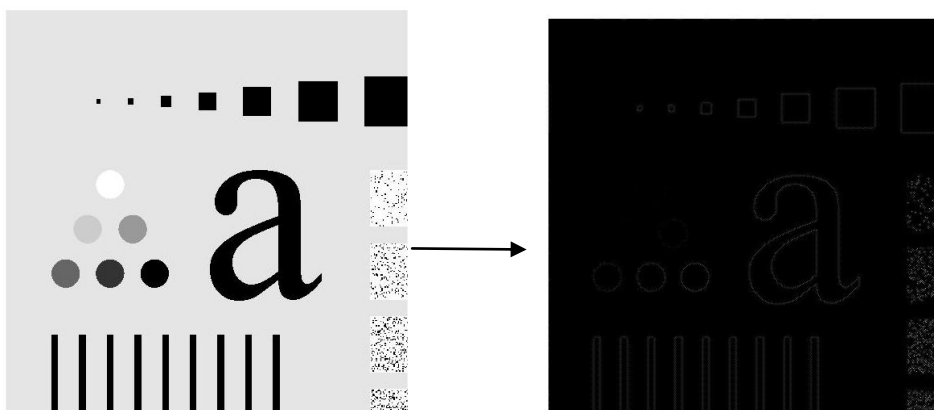


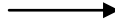
Highpass filters

Highpass ideal filter:

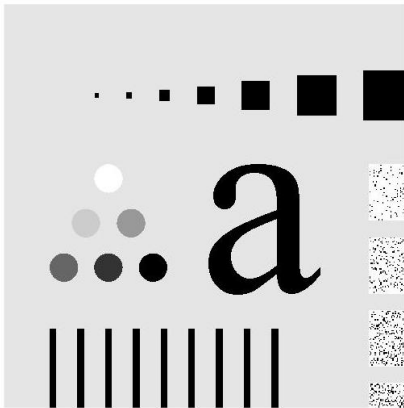


Highpass Butterworth filter:



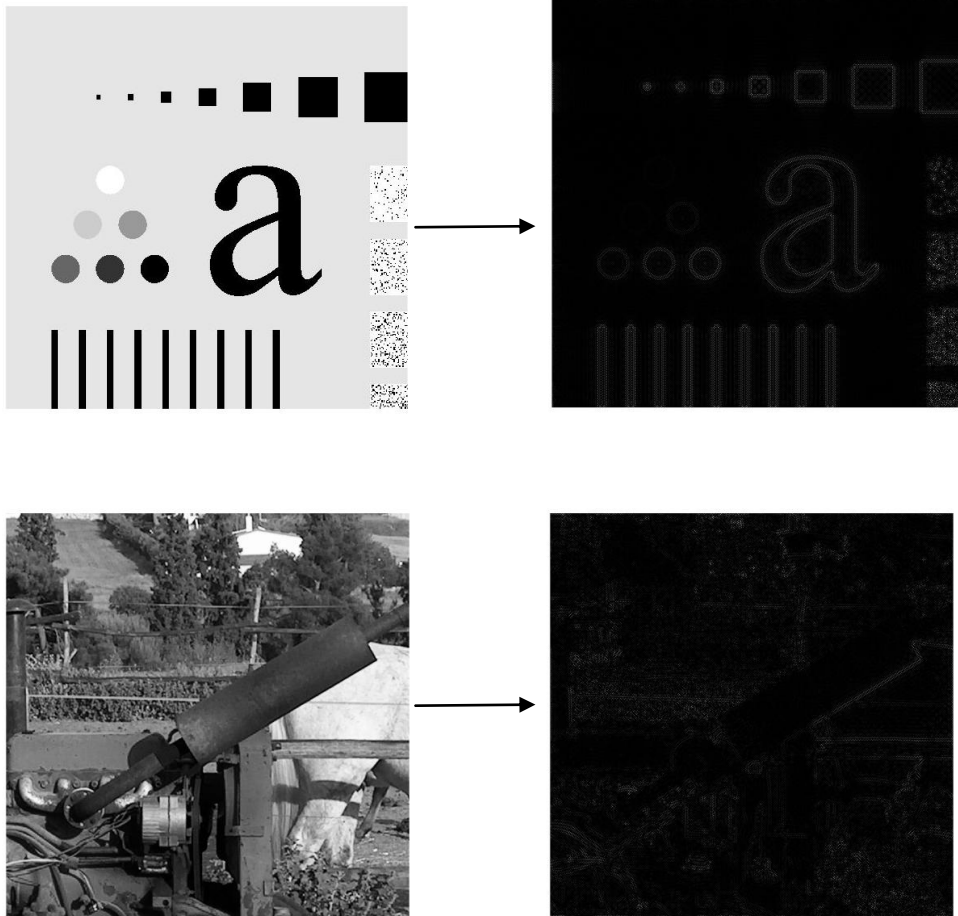


Highpass Gauss filter :

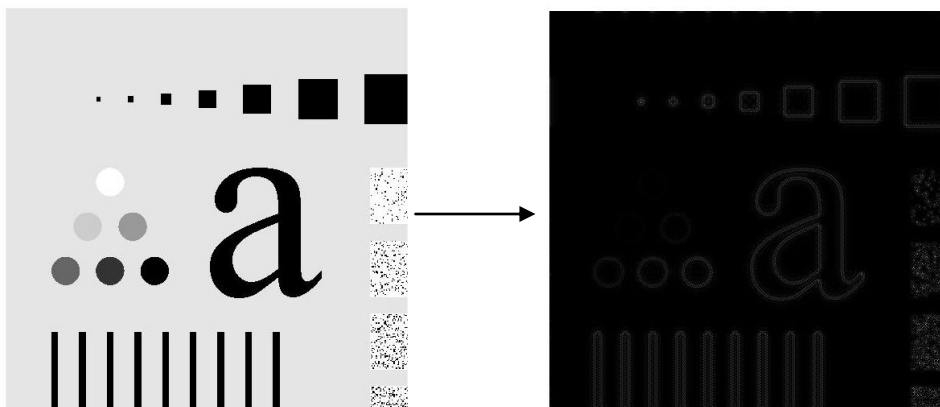


Bandpass filters

Bandpass ideal filter :

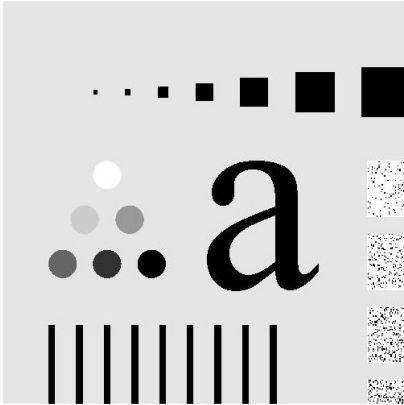


Bandpass Butterworth filter :





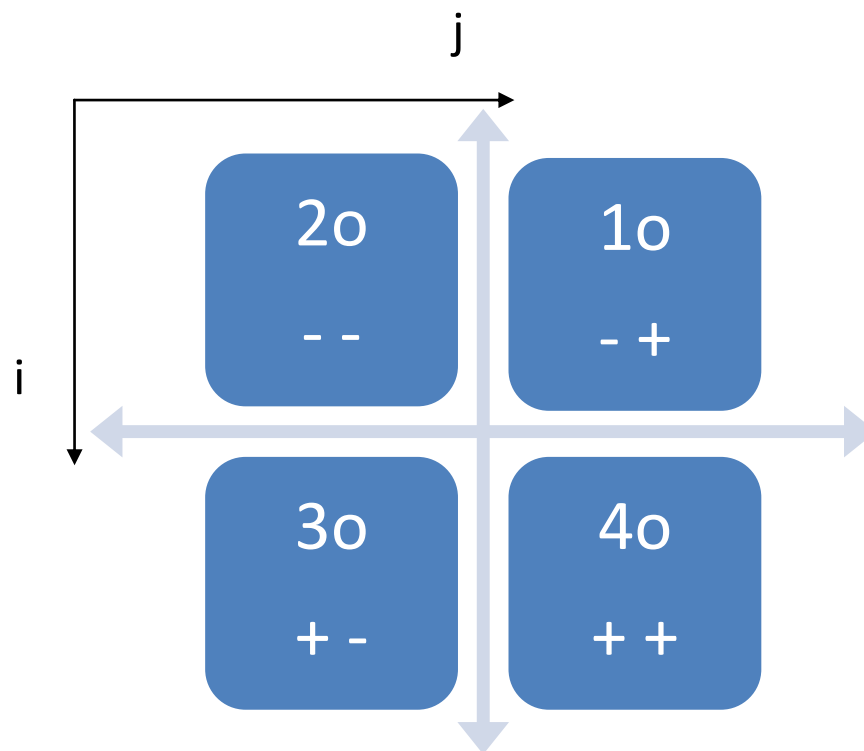
Bandpass Gauss filter :



Ενότητα 3

Η δημιουργία των κατευθυντικών φίλτρων υλοποιείται από την συνάρτηση `myDirectional()` της οποίας τη σχεδίαση θα εξηγήσω παρακάτω :

- ✧ Αρχικά υπολογίζει το άνοιγμα της γωνίας (upper-lower)
- ✧ Στη συνέχεια υπολογίζει το κέντρο της εικόνας που χρησιμεύει ώστε να βρίσκω σε ποιο τεταρτημόριο βρίσκεται το κάθε πίξελ.
- ✧ Ακολουθεί ένα διπλό loop στο οποίο :
 1. Γίνεται προσπέλαση κάθε στοιχείου του πίνακα χρησιμοποιώντας και τις συντ/νες του κέντρου της εικόνας
 2. Υπολογίζεται η γωνία για κάθε πίξελ ανάλογα σε ποιο τεταρτημόριο βρίσκεται. Παρατηρώ ότι τα πρόσημα των $i\text{-centroid}(1)$, $j\text{-centroid}(2)$ αλλάζουν ανάλογα με το τεταρτημόριο σύμφωνα με την παρακάτω εικόνα :

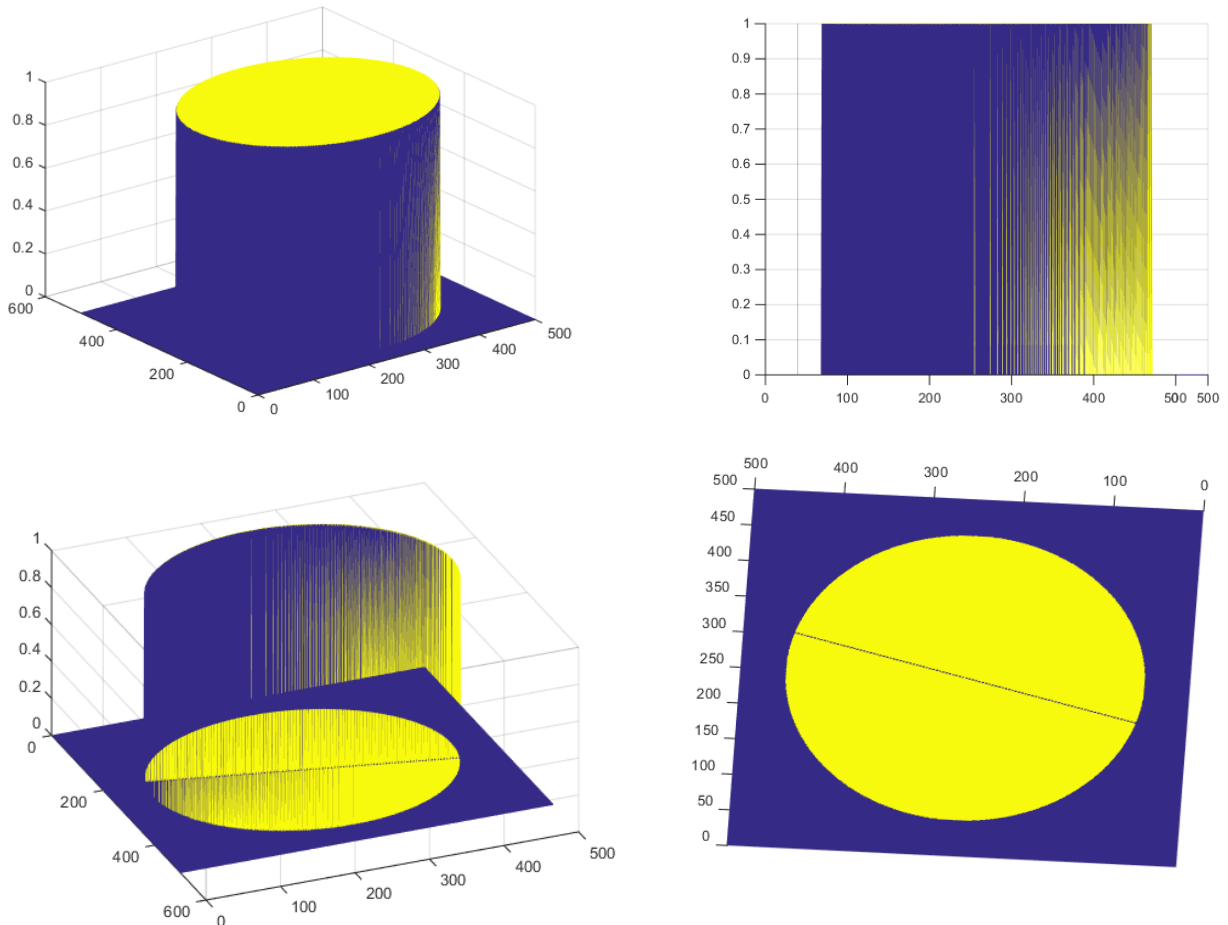


3. Τέλος αν η γωνία που βρέθηκε ανήκει στο στο διάστημα $(lower, upper]$ τότε

βάζω την τιμή 1 στο ανάλογο πίξελ.

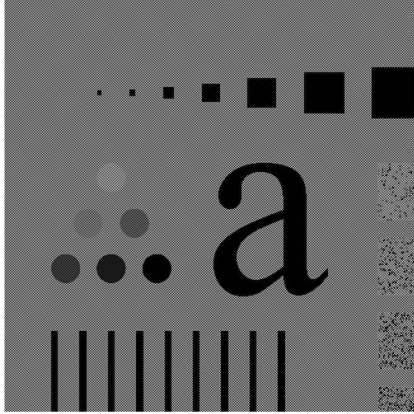
- ✧ Ακολουθεί ακόμα ένα διπλό loop στο οποίο υλοποιείται ο καθρεπτισμός της γωνίας που δημιουργήθηκε ως προς το κέντρο του πίνακα.
- ✧ Σημείωση : η συνάρτηση λειτουργεί ορθά για $\theta \rightarrow [0, 180]$ $\phi \rightarrow [0, 180]$

Ακολουθεί το demo3.m . Για να κατασκευάσω 6 μη επικαλυπτόμενα φίλτρα καθώς και για να επιβεβαιώσω ότι δεν επικαλύπτονται καλώ τη συνάρτηση completeCircle.m η οποία με τη σειρά της καλεί πολλές φορές την myDirectional() με τα κατάλληλα ορίσματα . Η συνάρτηση αυτή δημιουργεί κατευθυντικά φίλτρα με $\phi=30$ μοίρες για τις γωνίες $\theta=0,30,60,90,120,150$ καθώς μόνο αυτές χρειάζονται για να δημιουργηθεί ο πλήρης κύκλος. Στη συνέχεια προσθέτω τα διαφορετικά κατευθυντικά φίλτρα που δημιουργούνται και προκύπτει το παρακάτω αποτέλεσμα(εμφανίζεται από πολλές όψεις) :

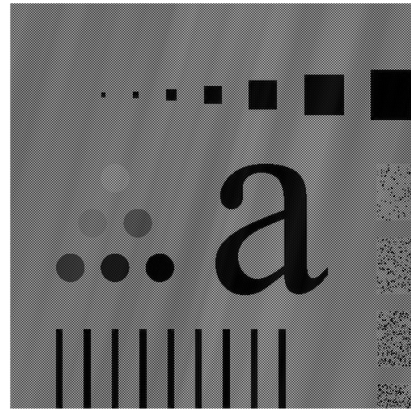


Όπως φαίνεται από τις εικόνες δεν επικαλύπτονται τα φίλτρα καθόλου καθώς υπάρχει παντού 1 σε όλα τα στοιχεία του πίνακα(η πρόσθεση των φίλτρων θα έδινε αριθμούς μεγαλύτερους από 1 αν υπήρχε επικάλυψη). Σγχρόνως όμως φαίνεται ένα bug καθώς μια ευθεία δεν γεμίζεται από τον αλγόριθμο. Δεν κατάφερα να βρω που οφείλεται το συγκεκριμένο πρόβλημα και συνέχισα με αυτήν την υλοποίηση.

Μέσα στο demo3.m αφού καλεστεί η completeCircle() και η myLowPassIdeal() και υλοποιηθεί το φιλτράρισμα και με τους δυο τρόπους προκύπτουν οι παρακάτω τελικές εικόνες για την εικόνα 1 :



Χρησιμοποιώντας Lowpass Ideal



Χρησιμοποιώντας Directional

Όπως φαίνεται και από τις εικόνες η μοναδική διαφορά τους οφείλεται στο χωρικό ημίτονο που δημιουργείται λόγω της συγκεκριμένης ευθείας. Αυτό επιβεβαιώνεται καθώς έχει την ίδια κατεύθυνση στο χώρο της εικόνας με την ευθεία. Αν είχα καταφέρει να λύσω το πρόβλημα του αλγορίθμου μου τότε η εικόνα θα ήταν ακριβώς η ίδια αφού δεν υπήρχαν επικαλύψεις.

Για την εικόνα 2 προκύπτουν οι παρακάτω εικόνες :



Χρησιμοποιώντας Lowpass Ideal



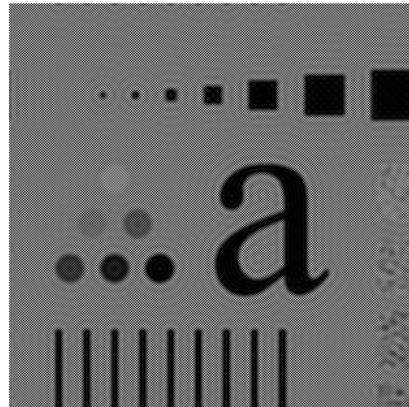
Χρησιμοποιώντας Directional

Στη δεύτερη εικόνα δεν φαίνεται τόσο πολύ η επίδραση του χωρικού ημιτόνου καθώς δεν έχει μεγάλη επιφάνεια με κοινή απόχρωση του γκρι ώστε να φανεί εκεί όπως στην εικόνα 1. Παρουσιάζεται όμως πολύ μικρή διαφορά στις δυο εικόνες από την παρουσία του.

Για να συνεχίσει το demo3.m στην κατασκευή των υπόλοιπων φίλτρων ζητείται ο χρήστης να πατήσει 1+ENTER. Ακολουθούν οι συγκρίσεις των φιλτραρισμάτων με χρήση κατευθυντικών και μη κατευθυντικών φίλτρων με δεδομένο ότι η χρήση των κατευθυντικών δημιουργεί το bug που ανέφερα παραπάνω. Άρα περιμένω να δω διαφορά μόνο στο χωρικό ημίτονο και σε τίποτα άλλο σε σχέση με το demo2.m . Οι δεξιά εικόνες είναι με τα μη κατευθυντικά φίλτρα ενώ οι αριστερά είναι με τα κατευθυντικά:

Lowpass filters

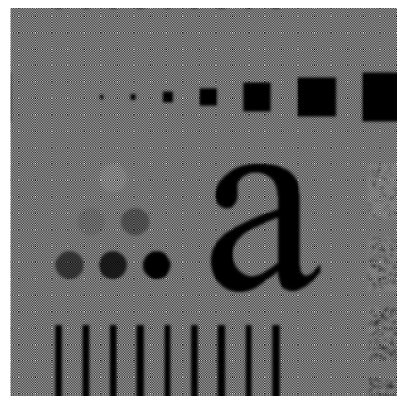
Lowpass ideal filter:



Lowpass Butterworth filter:



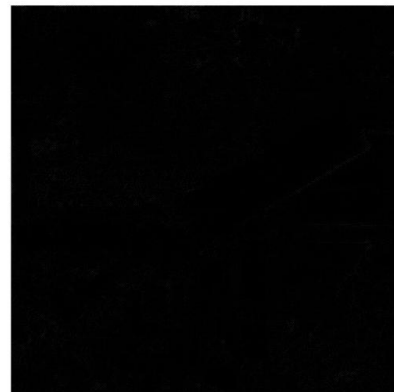
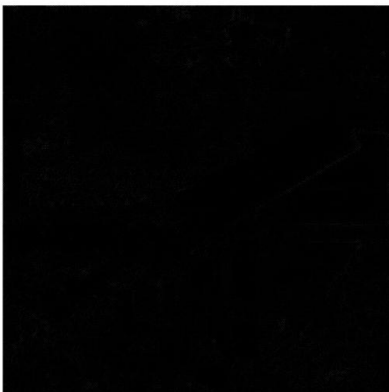
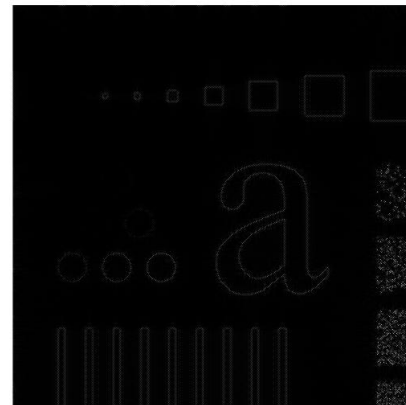
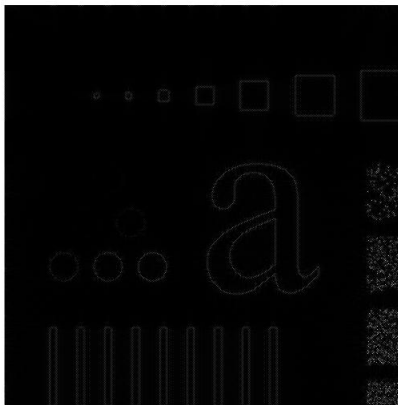
Lowpass Gauss filter :





Highpass filters

Highpass ideal filter:



Σημείωση 1: Συγκρίνοντας τις εικόνες παρατηρείται η ομοιότητα που περιμένα . Ομοίως συμβαίνει και στα υπόλοιπα και για αυτό το λόγο παραλείπονται.

Σημείωση 2: Το bug θα μπορούσε να διορθωθεί βαζοντας 1 στα πιξελς που έχουν τιμή 0 αλλά κάτι τέτοιο δεν θα ήταν κομμάτι του αλγορίθμου καθώς η υλοποίηση αυτή θα γινόταν αφού έχει δημιουργηθεί το φίλτρο και άρα δεν θα ήταν πραγματική επίλυση του προβλήματος. Προτίμησα λοιπόν να το αφήσω καθώς η μόνη διαφορά είναι το χωρικό ημίτονο.