



## Επεξεργασία Εικόνας και Βίντεο

### 1<sup>η</sup> Εργασία

### Φίλτρα εικόνων με MATLAB

Για τις Ασκήσεις θα χρησιμοποιήσετε τα ψηφία του Αριθμού Μητρώου Α.Μ.

Θεωρείστε το ΑΜ ως τετραψήφιο αριθμό με ψηφία: A B C D.

Δηλαδή ο ένας Φοιτητής με Α.Μ. 1343, θα έχει A = 1, B = 3, C = 4, D = 3

\*Οι απαντήσεις να δοθούν σε μορφή word

Προθεσμία: να παραδοθεί έως 17 Απριλίου

Νικόλαος Γιαννακέας



## Άσκηση 1<sup>η</sup>

Έστω ότι ο πίνακας είναι μια εικόνα: 
$$\begin{bmatrix} D & B & D \\ B & A & B \\ A & C & C \end{bmatrix}$$

A) Εφαρμόστε φίλτρο μέσου (mean filter), χρησιμοποιώντας την προσέγγιση zero padding για να διαχειριστείτε τα όρια της εικόνας.

B) Εφαρμόστε φίλτρο διάμεσου (median filter), χρησιμοποιώντας την προσέγγιση reflecting padding.

Γ) Επιβεβαιώσετε τα αποτελέσματα με κώδικα Matlab/Octave

**Υπόδειξη:** Τα δύο πρώτα ερωτήματα να λυθούν πραγματοποιώντας όλες τις απαιτούμενες πράξεις, ενώ για το τρίτο ερώτημα να δημιουργηθεί κώδικα. Μπορείτε να χρησιμοποιήσετε είτε έτοιμες συναρτήσεις MATLAB, είτε των κώδικα του εργαστηρίου.

## Άσκηση 2<sup>η</sup> (Με κώδικα Matlab/Octave)

Για καθεμία εικόνα (lenna1.png, lenna2.png, lenna3.png) να εκτιμήσετε κοιτάζοντας τις εικόνες το είδος υποβάθμισης που έχει υποστεί, να προτείνετε μια τεχνική βελτίωσης της εικόνας και να την υλοποιήσετε σε κατάλληλο λογισμικό (MATLAB/Octave). Στην απάντησή σας για κάθε περίπτωση να συμπεριλάβετε τον αντίστοιχο κώδικα κατάλληλα σχολιασμένο, καθώς και τη βελτιωμένη εικόνα.

## Άσκηση 3<sup>η</sup> (Με κώδικα Matlab/Octave)

Η εικόνα 'olumJ.png' έχει υποστεί παραμόρφωση, η οποία είναι περιοδική. Πρέπει να υπολογίσετε τον δισδιάστατο μετασχηματισμό της εικόνας και να εντοπίσετε στην εικόνα του τις συχνότητες που επικεντρώνεται η παραμόρφωση. Αφού τις εντοπίσετε να αφαιρέσετε τα σημεία εκείνα μηδενίζοντας τα και επαναφέρετε την εικόνα στο χωρικό πεδίο με την εφαρμογή του αντίστροφου μετασχηματισμού fourier. Συγκρίνεται την αρχική εικόνα με την βελτιωμένη.

**Υπόδειξη:**

1) Αφού εισάγετε την εικόνα, εφαρμόστε μετασχηματισμό fourier με την εντολή `fft2()`.

2) Χρησιμοποιήστε την εντολή `fftshift` για να ολισθήσετε τον μετασχηματισμός στο κέντρο της εικόνας

3) Εμφανίστε το εικόνα του χώρου των συχνοτήτων (προσοχή πρέπει να την μετατρέψετε σε 16-bit με την εντολή `uint16()`). Με την εντολή `imshow()`

4) Οι κορυφές βρίσκονται στις περιοχές: Περιοχή 1 (90:110,110:130), Περιοχή 2 (210:230,110:130). Μηδενίστε αυτές τις περιοχές στην εικόνα του μετασχηματισμού.

5) Αντιστρέψτε πρώτα την ολίσθηση με την εντολή `ifftshift` και μετά τον fourier με την εντολή `ifft2`

6) Εμφανίστε την τελική εικόνα. Με την εντολή `imshow()`



## ΑΠΑΝΤΗΣΕΙΣ

Για τη λύση των ασκήσεων χρειάζεται το Α.Μ. το οποίο είναι: A=1, B=8, C=7, D=3.

### Άσκηση 1<sup>η</sup>

Ο πίνακας είναι:  $\begin{bmatrix} D & B & D \\ B & A & B \\ A & C & C \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 8 & 3 \\ 8 & 1 & 8 \\ 1 & 7 & 7 \end{bmatrix}$

A) Εφαρμόστε φίλτρο μέσου (mean filter), χρησιμοποιώντας την προσέγγιση zero padding για να διαχειριστείτε τα όρια της εικόνας.

### Αρχική Εικόνα

3	8	3
8	1	8
1	7	7

### Zero Padding

$$1. I(1,1) = \frac{0+0+0+3+8+0+0+8+1}{9} = 2$$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0



2.  $I(1,2) = \frac{0+0+0+3+8+3+8+1+8}{9} = 3$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

3.  $I(1,3) = \frac{0+0+0+8+3+0+1+8+0}{9} = 2$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

4.  $I(2,1) = \frac{0+3+8+0+8+1+0+1+7}{9} = 3$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0



0	1	7	7	0
0	0	0	0	0

5.  $I(2,2) = \frac{3+8+3+8+1+8+1+7+7}{9} = 5$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

6.  $I(2,3) = \frac{8+3+0+1+8+0+7+7+0}{9} = 4$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

7.  $I(3,1) = \frac{0+8+1+0+1+7+0+0+0}{9} = 2$



0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

8.  $I(3,2) = \frac{8+1+8+1+7+7+0+0+0}{9} = 3$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0

9.  $I(3,3) = \frac{1+8+0+7+7+0+0+0+0}{9} = 2$

0	0	0	0	0
0	3	8	3	0
0	8	1	8	0
0	1	7	7	0
0	0	0	0	0



Άρα η τελική εικόνα με Zero Padding θα είναι:

2	3	2
3	5	4
2	4	3

Β) Εφαρμόστε φίλτρο διάμεσου (median filter), χρησιμοποιώντας την προσέγγιση reflecting padding.

### Reflecting Padding

1.  $I(1,1) = 1, 3, 3, 3, 3, 8, 8, 8, 8 = 3$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

2.  $I(1,2) = 1, 3, 3, 3, 3, 8, 8, 8, 8 = 3$

3	3	8	3	3
---	---	---	---	---



3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

3.  $I(1,3) = 1, 3, 3, 3, 3, 8, 8, 8, 8 = 3$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

4.  $I(2,1) = 1, 1, 1, 3, 3, 7, 8, 8, 8 = 3$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7





5.  $I(2,2) = 1, 1, 3, 3, 7, 7, 8, 8, 8 = 7$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

6.  $I(2,3) = 1, 3, 3, 7, 7, 7, 8, 8, 8 = 7$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

7.  $I(3,1) = 1, 1, 1, 1, 1, 7, 7, 8, 8 = 1$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7



8.  $I(3,2) = 1, 1, 1, 7, 7, 7, 7, 8, 8 = 7$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

9.  $I(3,3) = 1, 7, 7, 7, 7, 7, 7, 8, 8 = 7$

3	3	8	3	3
3	3	8	3	3
8	8	1	8	8
1	1	7	7	7
1	1	7	7	7

Άρα η τελική εικόνα με Reflecting Padding θα είναι:

3	3	3
3	7	7
1	7	7



Γ) Επιβεβαιώσετε τα αποτελέσματα με κώδικα Matlab/Octave

## Zero Padding

```
I = [0,0,0,0,0;0,3,8,3,0;0,8,1,8,0;0,1,7,7,0;0,0,0,0,0]; % η εικόνα μας
filtered = double(zeros(size(I,1),size(I,2))); % φτιαχνει έναν μηδενικό πίνακα που θα είναι το αποτέλεσμα
megethos_ypopinaka = 3; % το μέγεθος κάθε υποπίνακα
offset = megethos_ypopinaka/2 - 0.5; % η ακτίνα του
ypopinakas = zeros(megethos_ypopinaka,megethos_ypopinaka);
for i=1+offset:size(I,1)-offset % γραμμές
    for j=1+offset:size(I,2)-offset % στήλες
        ypopinakas(:,j)=I(i-offset:i+offset,j-offset:j+offset) % δηλώνει τον υποπίνακα που θα τρέχει κάθε φορά
        filtered(i,j) = uint8(round(mean2(ypopinakas))) % η πράξη της διαιρέσης με τον αριθμό του υποπίνακα
    end
end
```

### Command Window

ypopinakas =

1	8	0
7	7	0
0	0	0

filtered =

0	0	0	0	0
0	2	3	2	0
0	3	5	4	0
0	2	4	3	0
0	0	0	0	0



## Reflecting Padding

```
I = [3  3  8  3  3;
      3  3  8  3  3;
      8  8  1  8  8;
      1  1  7  7  7;
      1  1  7  7  7]; % η εικόνα μας

filtered = (zeros(size(I,1),size(I,2))); % φτιαχνει την τελικη εικόνα με μηδενικά στην αρχη
megethos_ypopinaka = 3; % το μεγεθος του υποπινακα κάθε φορά
offset = megethos_ypopinaka/2 - 0.5; % η ακτινα του
ypopinakas = zeros(megethos_ypopinaka,megethos_ypopinaka); % ο υποπινακας της εικόνας κάθε φορά
for i=1+offset:size(I,1)-offset % γραμμές
    for j=1+offset:size(I,2)-offset % στήλες
        ypopinakas(:, :) = I(i-offset:i+offset, j-offset:j+offset); % ο τρεχον υποπινακας κάθε φορά
        line_ypopinaka = ypopinakas(:);
        median_ypopinaka = median(line_ypopinaka) % το φίλτρο
        filtered(i,j) = median_ypopinaka
    end
end
```

0	0	0	0	0
0	3	3	3	0
0	3	7	7	0
0	1	7	0	0
0	0	0	0	0

median\_ypopinaka =

7

filtered =

0	0	0	0	0
0	3	3	3	0
0	3	7	7	0
0	1	7	7	0
0	0	0	0	0



## Άσκηση 2<sup>η</sup>

Για καθεμία εικόνα (lenna1.png, lenna2.png, lenna3.png) να εκτιμήσετε κοιτάζοντας τις εικόνες το είδος υποβάθμισης που έχει υποστεί, να προτείνετε μια τεχνική βελτίωσης της εικόνας και να την υλοποιήσετε σε κατάλληλο λογισμικό (MATLAB/Octave). Στην απάντησή σας για κάθε περίπτωση να συμπεριλάβετε τον αντίστοιχο κώδικα κατάλληλα σχολιασμένο, καθώς και τη βελτιωμένη εικόνα.

- Lenna 1

Από όσο μπορούμε να καταλάβουμε, εκτιμάμε ότι η εικόνα αυτή έχει υποστεί λευκό θόρυβο διότι παρατηρούμε λευκά σημάδια να παραμορφώνουν την εικόνα αυτή.



```
I = imread('lenna1.png');
filtered = uint8(zeros(size(I,1),size(I,2))); % φτιαχνει την τελικη εικονα με μηδενικα στην αρχη
megethos_ypopinaka = 3; % το μεγεθος του υποπινακα
offset = megethos_ypopinaka/2 - 0.5; % η ακτινα του
ypopinakas = zeros(megethos_ypopinaka,megethos_ypopinaka); % ο τυπος του υποπινακα καθε φορα
for i=1+offset:size(I,1)-offset % γραμμές
    for j=1+offset:size(I,2)-offset % στήλες
        ypopinakas(:, :) = I(i-offset:i+offset, j-offset:j+offset); % ο τρεχον υποπινακας καθε φορα
        line_ypopinaka = ypopinakas(:);
        median_ypopinaka = median(line_ypopinaka); % το φιλτρο
        filtered(i,j) = median_ypopinaka;
    end
end
figure, imshow(I)
figure, imshow(filtered)
```

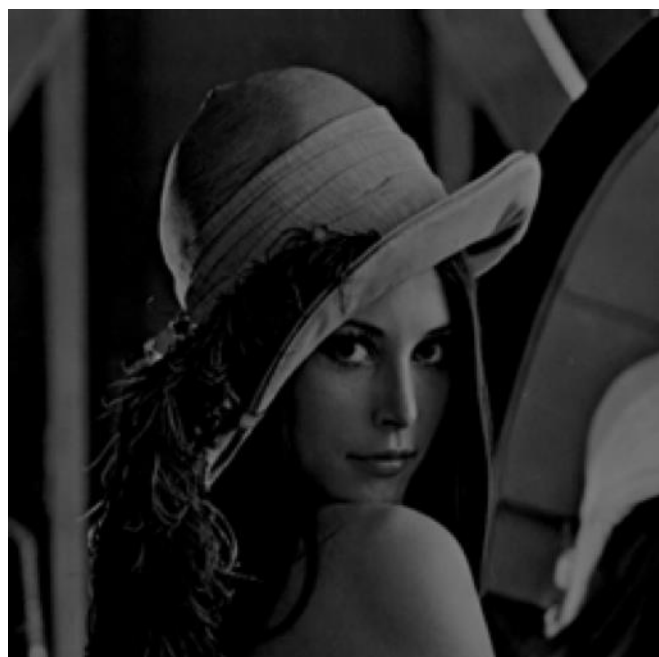


Χρησιμοποιώντας το median filter πάνω στην εικόνα, παρατηρούμε ότι είναι σαφώς πιο βελτιωμένη σε σχέση με την αρχική.



- Lenna 2

Από όσο μπορούμε να καταλάβουμε, εκτιμάμε ότι η εικόνα αυτή έχει υποστεί υψηλά επίπεδα θορύβου κι αυτό έχει ως αποτέλεσμα να υπάρχει χαμηλός φωτισμός στην εικόνα.





Για να διορθωθεί ο χαμηλός φωτισμός και να ενισχυθεί η φωτεινότητα της εικόνας πρέπει να χρησιμοποιηθούν τεχνικές αφαίρεσης θολότητας οι οποίες περιλαμβάνουν 3 βήματα:

```
A = imread('lenna2.png');  
figure, imshow(A);  
AInv = imcomplement(A);  
figure, imshow(AInv);  
BInv = imreducehaze(AInv);  
figure, imshow(BInv);  
B = imcomplement(BInv);  
figure, imshow(B);
```

#### 1. Αντιστροφή της αρχικής εικόνας



#### 2. Εφαρμογή αλγορίθμου αφαίρεσης θολότητας της εικόνας με ανεστραμμένο χαμηλό φωτισμό





3. Αντιστροφή της εικόνας αυτής -> Τελική βελτιωμένη εικόνα







- Lenna 3

Από όσο μπορούμε να καταλάβουμε, εκτιμάμε ότι η εικόνα αυτή έχει υποστεί θόρυβο salt & pepper, καθώς υπάρχουν λευκά και μαύρα σημάδια πάνω στην εικόνα.



```
I=imread('lenna3.png');  
I_1 = imnoise(I, 'salt & pepper', 0.05);  
I_filtered = I_1;  
for layer=1:3  
    I_filtered(:,:,layer) = wiener2(I_1(:,:,layer), [5 5]);  
end  
figure, imshow(I_1)  
figure, imshow(I_filtered)
```

Χρησιμοποιώντας το imnoise filter πάνω στην εικόνα, παρατηρούμε ότι είναι σαφώς πιο βελτιωμένη σε σχέση με την αρχική.



### Άσκηση 3<sup>η</sup>

```
I=imread('oIumj.png'); % εισαγωγή εικόνας
I_1=fft2(I); % μετασχηματισμός fourier
I_2=fftshift(I_1); % ολισθήση του μετασχηματισμού στο κέντρο

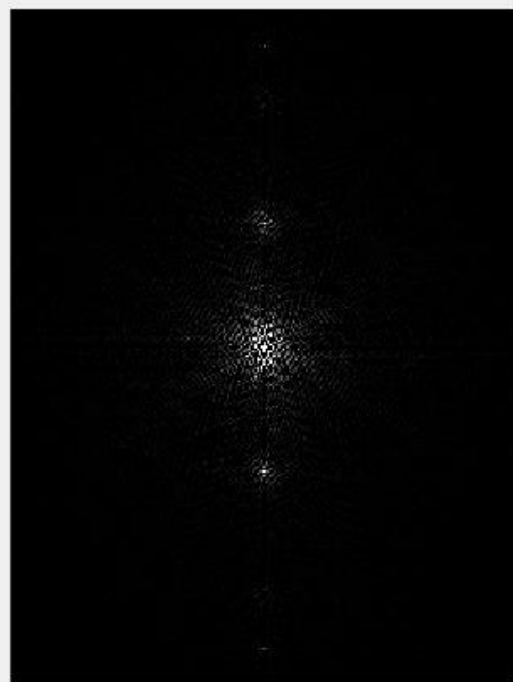
figure, imshow(I);
figure, imshow(uint16(I_2)); % μετατροπή της εικόνας σε 16-bit

% μηδενισμός των 2 περιοχών που βρίσκονται οι κορυφές της εικόνας
I_2(90:110,110:130)=0;
I_2(210:230,110:130)=0;

I_3=ifftshift(I_2); % αντιστροφή της ολισθήσης
I_4=ifft2(I_3); % αντιστροφή του fourier
figure, imshow(uint8(I_4))
```



Η συγκεκριμένη εικόνα έχει υποστεί παραμόρφωση, η οποία είναι περιοδική. Με βάση τα βήματα υλοποίησης του κώδικα υπολογίζεται ο δισδιάστατος μετασχηματισμός της εικόνας. Επιπλέον, όσον αφορά τις συχνότητες της εικόνας, είναι εμφανές ότι αυτές που επικεντρώνονται στην συγκεκριμένη παραμόρφωση επικεντρώνονται κυρίως στο κέντρο της εικόνας. Έτσι για τη βελτίωση της εικόνας, αφαιρέθηκαν τα σημεία αυτά μηδενίζοντάς τα και επανήλθε η εικόνα στο χωρικό πεδίο με την εφαρμογή του αντίστροφου μετασχηματισμού Fourier. Παρατηρούμε ότι η τελική είναι εμφανώς πιο βελτιωμένη.

**Αρχική****Συχνότητες****Τελική**