

Επεξεργασία Εικόνας & Βίντεο



03. Φίλτρα Εικόνων

Εισηγητής: Νικόλαος Γιαννακέας

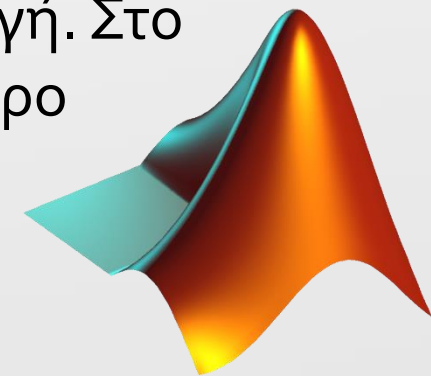
Επίκουρος Καθηγητής, Σημάτων & Συστημάτων





Εισαγωγή

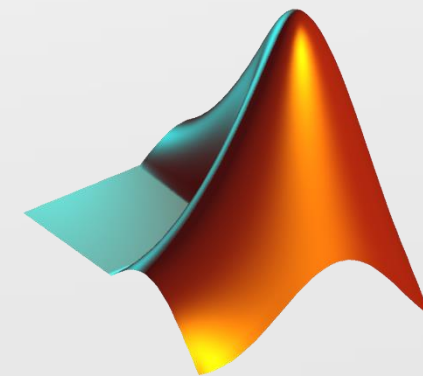
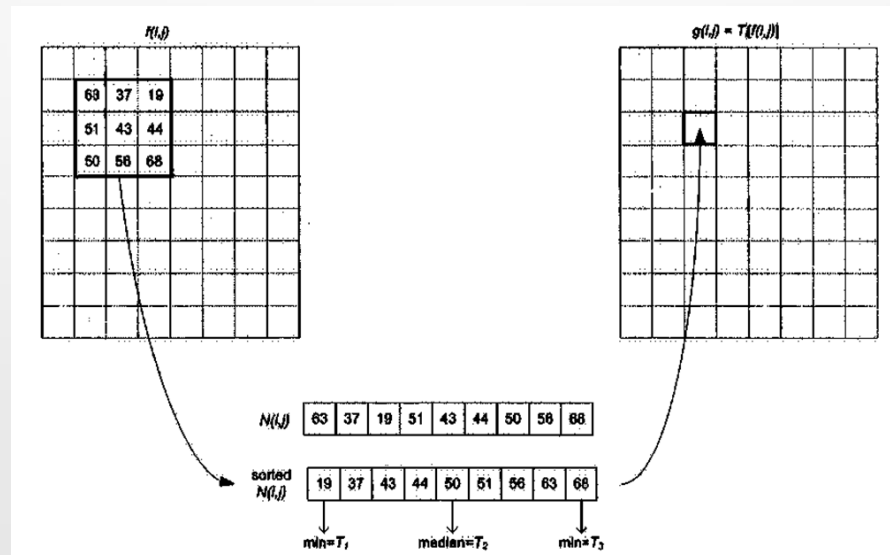
Η εφαρμογή φίλτρων είναι ίσως η πιο βασική μέθοδος προεπεξεργασίας (preprocessing) ή βελτίωσης (enhancement) μιας ψηφιακής εικόνας. Ανάλογα με την φύση του φίλτρου οι μέθοδοι μπορούν να τονίσουν ακμές, ή αντικείμενα, να «λειάνουν» περιοχές, ή και τα δύο ταυτόχρονα. Το φιλτράρισμα βασίζεται στην κίνηση ενός κυλιόμενου παραθύρου, το οποίο σαρώνει ολόκληρη την επιφάνεια της εικόνας και συνήθως αντικαθιστά την τιμή της φωτεινότητας του κεντρικού εικονοστοιχείου με τιμή η οποία υπολογίζεται ανάλογα με το φίλτρο. Το μέγεθος του παραθύρου είναι μια παράμετρος, η οποία μπορεί να διαφέρει ανάλογα με την εφαρμογή. Στο παρακάτω σχήμα φαίνεται για παράδειγμα η εφαρμογή ενός φίλτρο διαμέσου (median filter) με μέγεθος 3 επί 3.

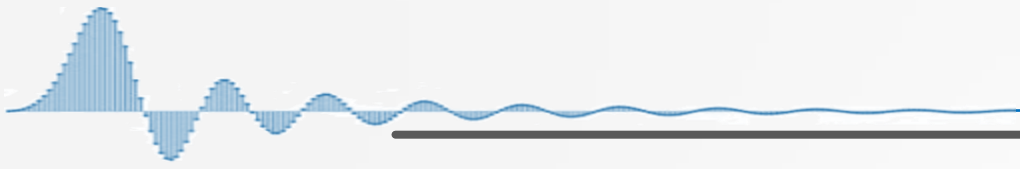




Εισαγωγή

Ένα τυχαίο τρέχον παράθυρο φαίνεται στο αριστερό μέρος την εικόνας, καθώς αυτή σαρώνεται. Από το παράθυρο αυτό λαμβάνονται όλες οι φωτεινότητες των εικονοστοιχείων, ταξινομούνται με αύξουσα σειρά και επιλέγεται η διάμεσος τιμή δηλαδή η κεντρική τιμή τις κατάταξης. Στην τελική φιλτραρισμένη εικόνα η τιμή της φωτεινότητας του εικονοστοιχείου που βρίσκεται στο κέντρο αυτού του παραθύρου αντικαθίσταται με την διάμεσο τιμή.

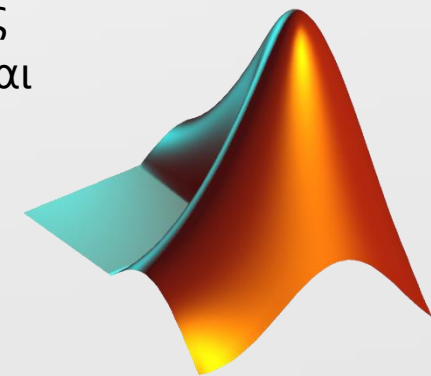


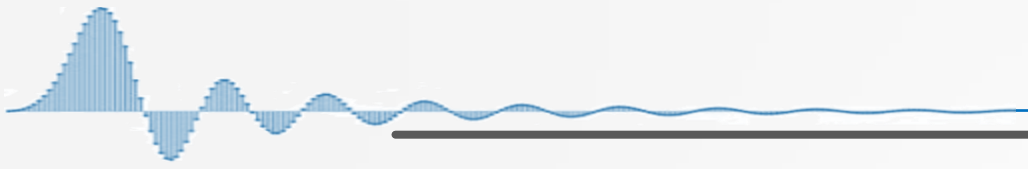


Εντολές Εισαγωγής-Εξαγωγής

	Φίλτρο	Εντολή MATLAB	Περιγραφή
1	Γκαουσιανό	imgaussfilt	2-D Gaussian filtering of images
2	Wiener	wiener2	2-D adaptive noise-removal filtering
3	Διάμεσου	medfilt2	2-D median filtering
4	Τυπικής Απόκλισης	stdfilt	Local standard deviation of image
5	Εύρους	rangefilt	Local range of image
6	Εντροπίας	entropyfilt	Local entropy of grayscale image

Με την ίδια ακριβώς λογική μπορεί να υλοποιηθεί το φίλτρο μέσης τιμής, το οποίο έχει ως αποτέλεσμα την λείανση περιοχών. Κάποια βασικά φίλτρα επεξεργασίας εικόνας φαίνονται στον παραπάνω πίνακα.

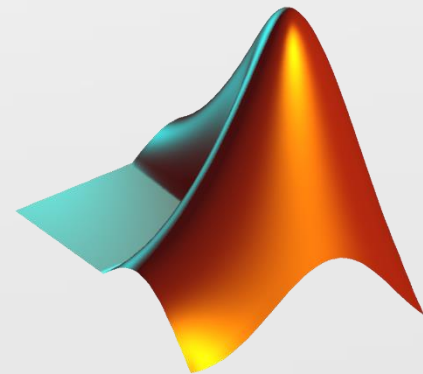




ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 1

Εισάγεται μια εικόνα από τον φάκελο datasets και υλοποιήστε ένα φίλτρο μέσης τιμής (mean filter). Ορίστε σε παράμετρο το μέγεθος του φίλτρου. Μετά την εφαρμογή του φίλτρου εμφανίστε και την αρχική εικόνα αλλά και την φιλτραρισμένη με σκοπό να φανούν οι διαφορές τους.





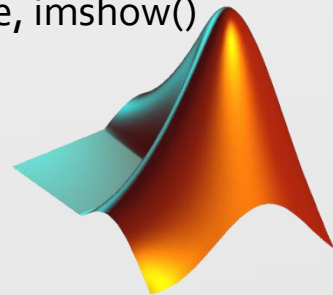
ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

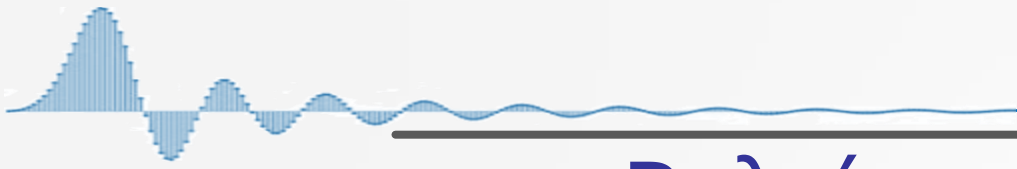
Άσκηση 1

Υπόδειξη: Ακολουθείστε τα βήματα της υπόδειξης για να βοηθηθείτε στην συγγραφή του κώδικά

1. Διαβάστε την Εικόνα από το αρχείο τις με την εντολή `imread()`.
2. Αρχικοποιήστε μια εικόνα με μηδενικές τιμές ίδιου μεγέθους με την αρχική, ώστε αυτή να είναι η τελική φιλτραρισμένη
3. Ορίστε σε μεταβλητή το μέγεθος του παραθύρου. Π.χ. αν το φίλτρο είναι 3×3 ορίστε του ίσο με 3.
4. Υπολογίστε από το μέγεθος του φίλτρου πόσο είναι το offset από το κεντρικό Pixel σε κάθε πλευρά.
5. Αρχικοποιήστε ένα μηδενικό παράθυρο διάστασης ίσης με το παράθυρο που θα σαρώσουμε την εικόνα.
6. Υλοποιήστε διπλό φωλιασμένο βρόχο `for` ώστε να τρέχει στις γραμμές και τις στήλες.
7. Για κάθε τρέχον παράθυρο μέσα στον δεύτερο φωλιασμένο βρόχο να υπολογίζεται η μέση τιμή της φωτεινότητας του παραθύρου.
8. Βάλτε την μέση τιμή που υπολογίσατε στην κεντρική θέση που «τρέχει» το παράθυρο, στην τελική φιλτραρισμένη εικόνα.
9. Τερματίστε τους δύο βρόχους
10. Εμφανίστε και την αρχική εικόνα και την τελική σε ξεχωριστά σχήματα με την εντολή `figure, imshow()`

Τι παρατηρείτε στην φιλτραρισμένη εικόνα σε σχέση με την αρχική;





Βελτίωση στο πεδίο του Χώρου

Φίλτρα Εξομάλυνσης

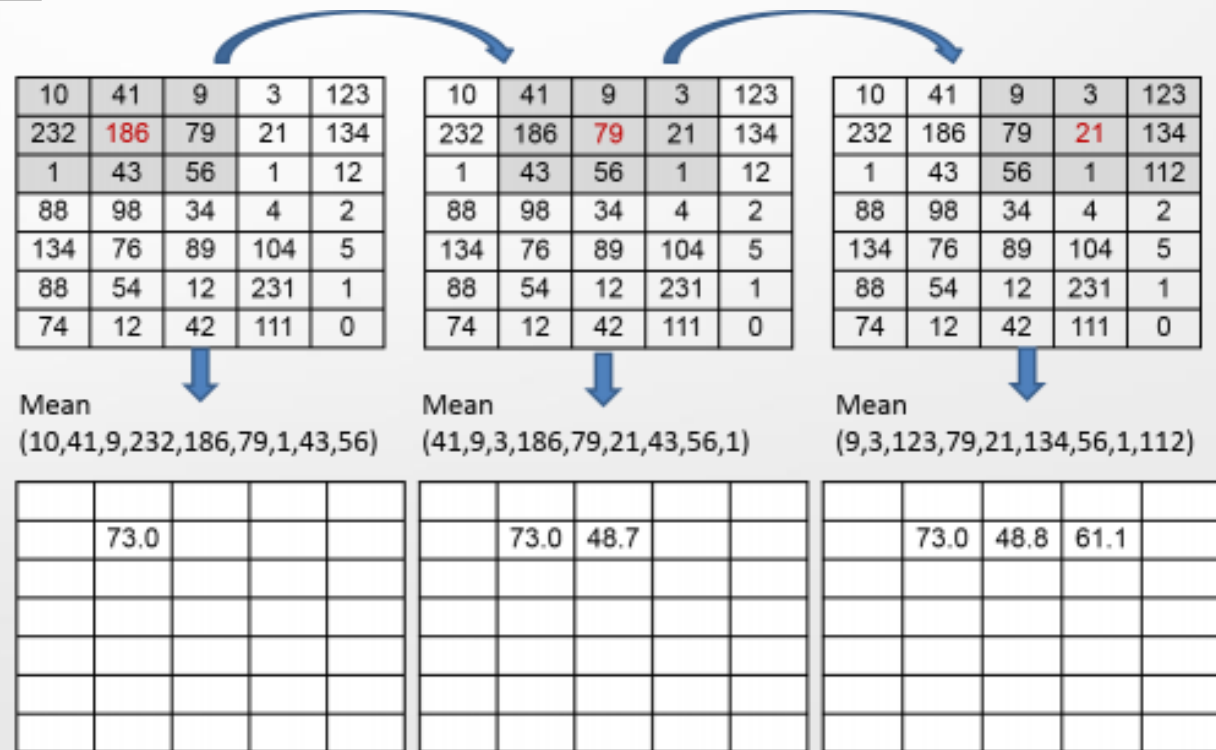
Mean Filter

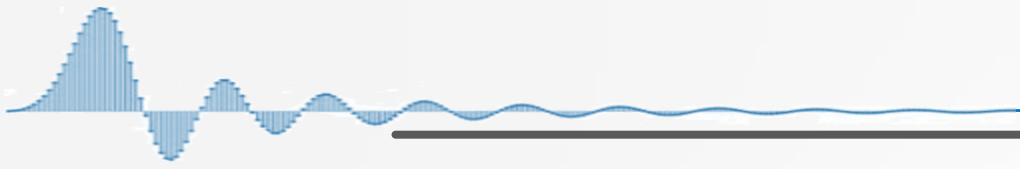
Φίλτρο Μέσου

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{x,y}} g(s, t)$$

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Στην πραγματικότητα γίνεται συνέλιξη με την μάσκα





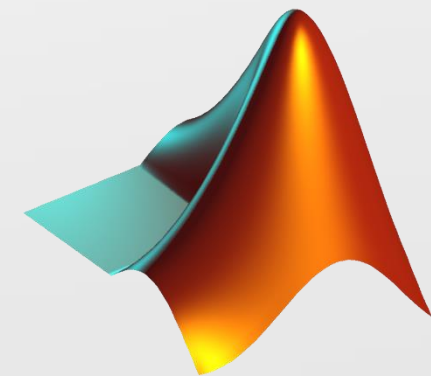
ΛΥΣΗ

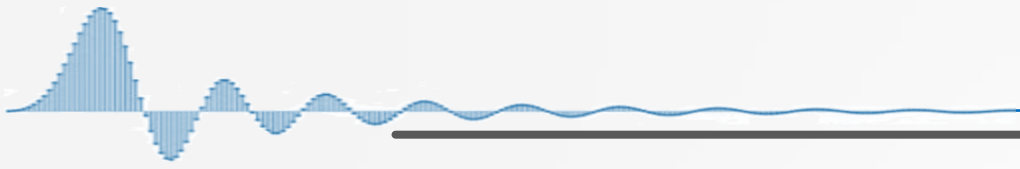
```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));
```

Μετατρέπουμε σε 8bit απρόσημο ακέραιο πίνακα

Διάβασμα εικόνας

Αρχικοποιούμε ένα πίνακα filtered με μέγεθος το μέγεθος του πίνακα I





ΛΥΣΗ

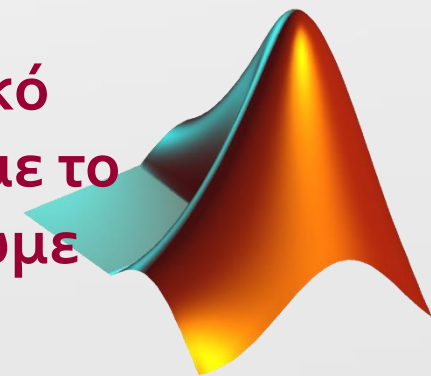
```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));  
megethos_parathyrou = 3;  
offset = megethos_parathyrou/2 - 0.5;  
parathyro = zeros(megethos_parathyrou,  
megethos_parathyrou);
```

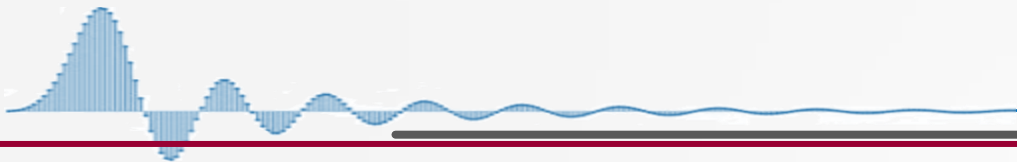
Ορίζουμε το μέγεθος του κυλιόμενου παραθύρου με 3

Ορίζουμε την ακτίνα του παραθύρου

Αρχικοποιούμε ένα μηδενικό παράθυρο διάστασης ίσης με το παράθυρο που θα σαρώσουμε την εικόνα.

10	41	9	3	123
232	186	79	21	134
1	43	56	1	12
88	98	34	4	2
134	76	89	104	5
88	54	12	231	1
74	12	42	111	0





```

I = imread('Noise_salt_and_pepper.png');
filtered = uint8(zeros(size(I,1),size(I,2)));
megethos_parathyrou = 3;
offset = megethos_parathyrou/2 - 0.5;
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);

```

`for i=1+offset:size(I,1)-offset` → Για τις γραμμές του πίνακα με βάση το offset

`for j=1+offset:size(I,2)-offset` → Για τις στήλες του πίνακα με βάση το offset

```
parathyro(:,j)=I(i-offset:i+offset,j-offset:j+offset);
```

```
filtered(i,j) = uint8(round(mean2(parathyro)));
```

```
end
```

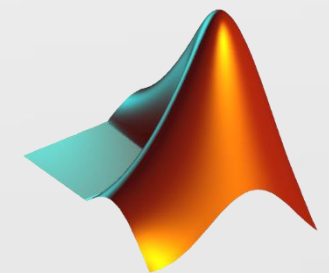
```
end
```

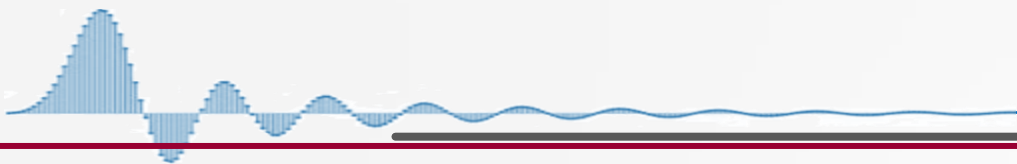
✓ Υλοποίηση for:

- ✓ Στο πίνακα `parathyro` εκχωρούμε το 3x3 με βήμα μετατόπισης το offset
- ✓ Στο πίνακα `filtered` εκχωρούμε την μέση τιμή από το 3x3 πίνακα `parathyro` στρογγυλοποιημένο μέσω της `round`

ΛΥΣΗ

10	41	9	3	123	10	41	9	3	123	10	41	9	3	123
232	186	79	21	134	232	186	79	21	134	232	186	79	21	134
1	43	56	1	12	1	43	56	1	12	1	43	56	1	112
88	98	34	4	2	88	98	34	4	2	88	98	34	4	2
134	76	89	104	5	134	76	89	104	5	134	76	89	104	5
88	54	12	231	1	88	54	12	231	1	88	54	12	231	1
74	12	42	111	0	74	12	42	111	0	74	12	42	111	0
Mean					Mean					Mean				
(10,41,9,232,186,79,1,43,56)					(41,9,3,186,79,21,43,56,1)					(9,3,123,79,21,134,56,1,112)				
	73.0					73.0	48.7				73.0	48.8	61.1	





```
I = imread('Noise_salt_and_pepper.png');
filtered = uint8(zeros(size(I,1),size(I,2)));
megethos_parathyrou = 3;
offset = megethos_parathyrou/2 - 0.5;
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);
for i=1+offset:size(I,1)-offset
    for j=1+offset:size(I,2)-offset
        parathyro(:,j)=I(i-offset:i+offset,j-offset:j+offset);
        filtered(i,j) = uint8(round(mean2(parathyro)));
    end
end
```

figure, imshow(I)



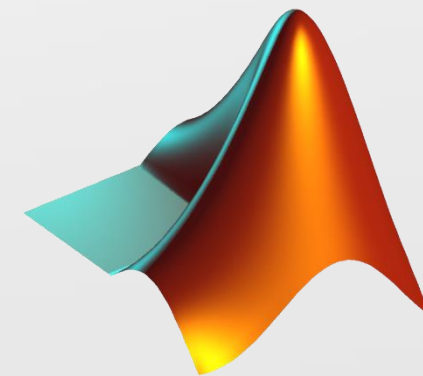
Εμφάνιση αρχικής εικόνας

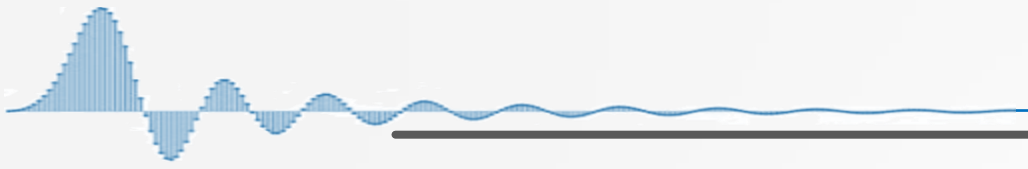
figure, imshow(filtered)



Εμφάνιση φιλτραρισμένης
εικόνας

ΛΥΣΗ





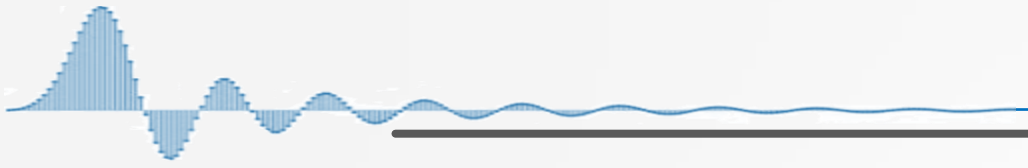
ΛΥΣΗ

```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));  
megethos_parathyrou = 3;  
offset = megethos_parathyrou/2 - 0.5;  
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);  
for i=1+offset:size(I,1)-offset  
    for j=1+offset:size(I,2)-offset  
        parathyro(:, :) = I(i-offset:i+offset, j-offset:j+offset);  
        filtered(i, j) = uint8(round(mean2(parathyro)));  
    end  
end
```

figure, imshow(I)

figure, imshow(filtered)



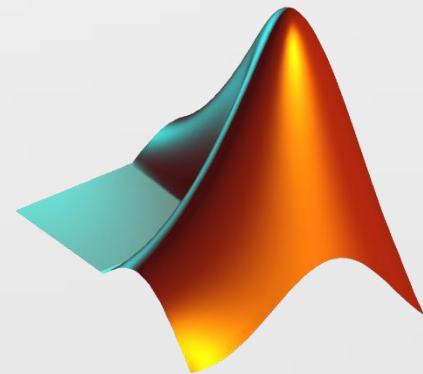


ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 2

Υλοποιήστε κατά αντιστοιχία ένα φίλτρο διάμεσου.
Εφαρμόστε το σε εικόνα με salt and pepper θόρυβο και δείτε τα αποτελέσματα. Τι παρατηρείτε;

Υπόδειξη: Χρησιμοποιήστε την ίδια ακριβώς δομή με την πρώτη άσκηση αλλάζοντας μόνο τον κώδικα μέσα στους βρόχους ώστε να υπολογίζεται η διάμεσος τιμή αντί της μέσης.





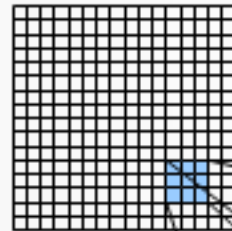
Βελτίωση στο πεδίο του Χώρου

Φίλτρα Εξομάλυνσης

Median Filter (μη γραμμικό)
Φίλτρο Διαμέσου

$$\hat{f}(x, y) = \underset{(s,t) \in S(x,y)}{\text{median}} \{g(s, t)\}$$

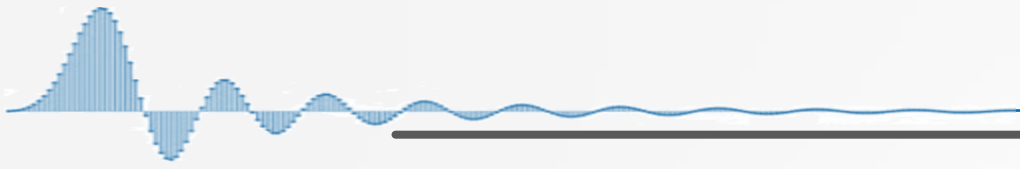
Αποδοτικότερο σε
κρουστικό θόρυβο



101	69	0
56	255	87
123	96	157

Για κάθε γειτονιά από εικονοστοιχεία οι φωτεινότητες ταξινομούνται κατά αύξουσα σειρά και λαμβάνεται η διάμεσος τιμή

0	56	69	87	96	101	123	157	255
---	----	----	----	----	-----	-----	-----	-----



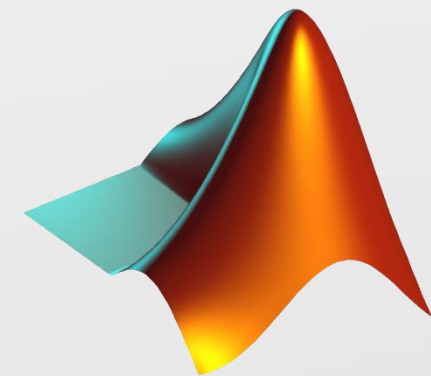
ΛΥΣΗ

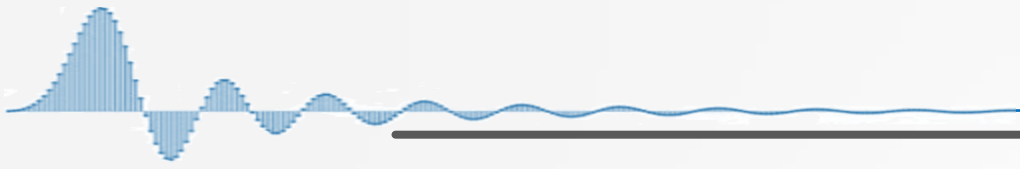
```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));
```

Μετατρέπουμε σε 8bit απρόσημο ακέραιο πίνακα

Διάβασμα εικόνας

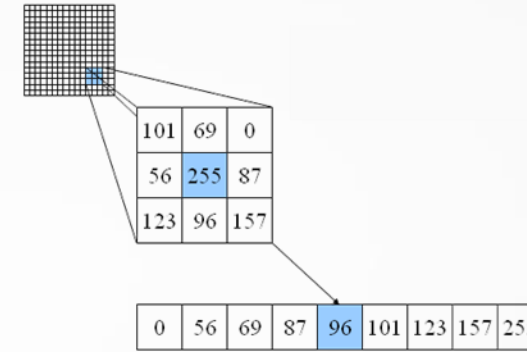
Αρχικοποιούμε ένα πίνακα filtered με μέγεθος το μέγεθος του πίνακα I





ΛΥΣΗ

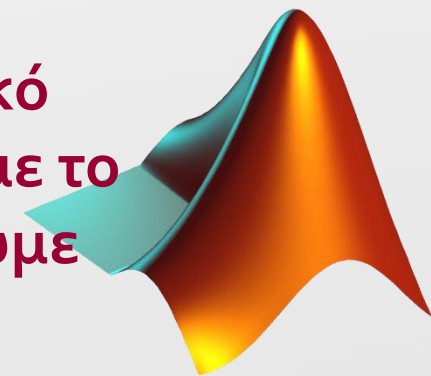
```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));  
megethos_parathyrou = 3;  
offset = megethos_parathyrou/2 - 0.5;  
parathyro = zeros(megethos_parathyrou,  
megethos_parathyrou);
```



Ορίζουμε το μέγεθος του κυλιόμενου παραθύρου με 3

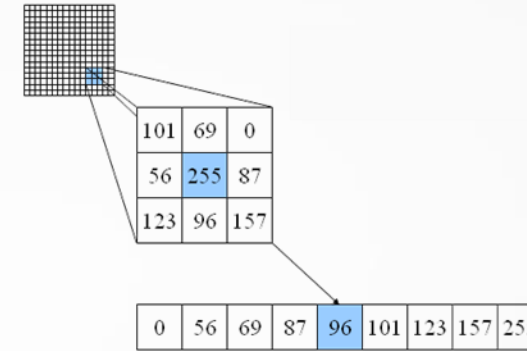
Ορίζουμε την ακτίνα του παραθύρου

Αρχικοποιούμε ένα μηδενικό παράθυρο διάστασης ίσης με το παράθυρο που θα σαρώσουμε την εικόνα.





```
I = imread('Noise_salt_and_pepper.png');
filtered = uint8(zeros(size(I,1),size(I,2)));
megethos_parathyrou = 3;
offset = megethos_parathyrou/2 - 0.5;
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);
for i=1+offset:size(I,1)-offset
    for j=1+offset:size(I,2)-offset
        parathyro(:, :) = I(i-offset:i+offset, j-offset:j+offset);
        line_parathyro = parathyro(:);
        median_parathyroy = median(line_parathyro);
        filtered(i,j) = median_parathyroy;
    end
end
```



ΛΥΣΗ



✓ Υλοποίηση for:

- ✓ Στο πίνακα `parathyro` εκχωρούμε το 3x3 με εύρος μετατόπισης το `offset`
- ✓ Στο πίνακα `line_parathyro` εκχωρούμε το πίνακα σε μορφή στήλης
- ✓ Μέσω της εντολής `median` βρίσκουμε την διάμεσο και την εκχωρούμε στο πίνακα `filtered`



```
I = imread('Noise_salt_and_pepper.png');
filtered = uint8(zeros(size(I,1),size(I,2)));
megethos_parathyrou = 3;
offset = megethos_parathyrou/2 - 0.5;
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);
for i=1+offset:size(I,1)-offset
    for j=1+offset:size(I,2)-offset
        parathyro(:, :) = I(i-offset:i+offset, j-offset:j+offset);
        line_parathyro = parathyro(:);
        median_parathyro = median(line_parathyro);
        filtered(i, j) = median_parathyro;
    end
end
```

figure, imshow(I)



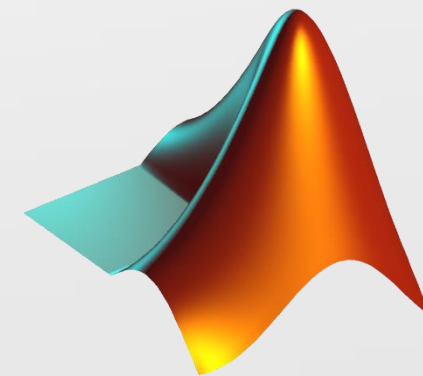
Εμφάνιση αρχικής εικόνας

figure, imshow(filtered)



Εμφάνιση φιλτραρισμένης
εικόνας

ΛΥΣΗ





```
I = imread('Noise_salt_and_pepper.png');  
filtered = uint8(zeros(size(I,1),size(I,2)));  
megethos_parathyrou = 3;  
offset = megethos_parathyrou/2 - 0.5;  
parathyro = zeros(megethos_parathyrou,megethos_parathyrou);  
for i=1+offset:size(I,1)-offset  
    for j=1+offset:size(I,2)-offset  
        parathyro(:,:)=I(i-offset:i+offset,j-offset:j+offset);  
        line_parathyro = parathyro(:);  
        median_parathyroy = median(line_parathyro);  
        filtered(i,j) = median_parathyroy;  
    end  
end  
  
figure, imshow(I)  
figure, imshow(filtered)
```

