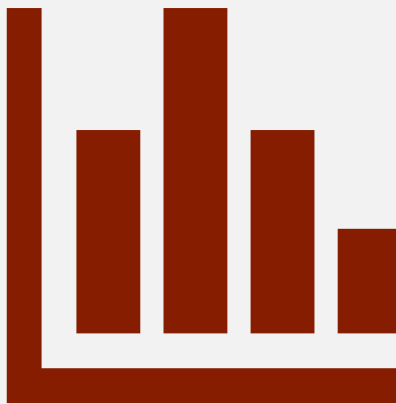




ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ

Αναφορά 2^{ης} Εργαστηριακής Άσκησης



















«ΑΠΛΕΣ ΤΕΧΝΙΚΕΣ ΣΤΗΝ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ»



 Καραμαϊλής Παντελής 2016030040
 Βλάχος Κωνσταντίνος 2016030042
 Γαλάνης Μιχάλης 2016030036

ΠΕΡΙΕΧΟΜΕΝΑ

*Οι σύνδεσμοι για τις παρακάτω ενότητες είναι διαδραστικοί.
Πιέστε πάνω στο επιθυμητό τμήμα της άσκησης για τη μετάβαση σε αυτό.*

	ΕΙΣΑΓΩΓΗ	2
	Σκοπός Εργαστηρίου	2
	1 – HISTOGRAM ANALYSIS	2
	Επισκόπηση & Θεωρητική Ανάλυση	2
	Υλοποίηση.....	2
	Αποτελέσματα & Συμπεράσματα	3
	2 – MEDIAN & GAUSSIAN FILTERS	4
	Επισκόπηση & Θεωρητική Ανάλυση	4
	Υλοποίηση.....	4
	Αποτελέσματα & Συμπεράσματα	5
	3 – NOISE REDUCT. USING FILTERS	6
	Επισκόπηση & Θεωρητική Ανάλυση	6
	Υλοποίηση.....	6
	Αποτελέσματα & Συμπεράσματα	7
	4 – FLOATING PARTICLES	1
	Υλοποίηση.....	1
	Αποτελέσματα & Συμπεράσματα	1
	ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ	2

ΕΙΣΑΓΩΓΗ

Σκοπός Εργαστηρίου

Σκοπός της άσκησης αυτής είναι η μελέτη με βασικές τεχνικές επεξεργασίας σε ψηφιακές εικόνες. Συγκεκριμένα θα ασχοληθούμε με:

- **Ιστογράμματα:** που επιτρέπουν την ανάλυση μιας εικόνας με την απεικόνιση της στοχαστικής κατανομής της.
- **Φίλτρα:** lowpass όπως median, gaussian & average που χρησιμοποιούνται για την ομαλοποίηση μιας εικόνας (ελάττωση του θορύβου) ή highpass για την όξυνση μιας εικόνας (αύξηση έντασης ορίων).

1 – HISTOGRAM ANALYSIS

Στο πρώτο ερώτημα μας ζητήθηκε να διαβάσουμε μια εικόνα της επιλογής μας που δεν έχει πετυχημένη αντίθεση και να την ισοσταθμίσουμε μελετώντας το ιστογράμμα της. Στη συνέχεια μας ζητήθηκε να τη μετατρέψουμε σε ασπρόμαυρη χρησιμοποιώντας κατάλληλο threshold.

Επισκόπηση & Θεωρητική Ανάλυση

Γνωρίζουμε από τη θεωρία, ότι το ιστογράμμα μιας ψηφιακής εικόνας είναι μια ένδειξη της στοχαστικής κατανομής της εικόνας. Πρακτικά, πολλά χαρακτηριστικά της εικόνας περιγράφονται σε αυτό το διάγραμμα. Η συγκέντρωση για παράδειγμα του ιστογράμματος κοντά σε μικρές ή μεγάλες τιμές αντικατοπτρίζουν τη φωτεινότητά της και το εύρος της συγκέντρωσης υποδηλώνει την αντίθεση της.

Μια εικόνα λέγεται ισοσταθμισμένη όταν η συγκέντρωσή του ιστογράμματος είναι ομοιόμορφα κατανομημένη σε όλα τα επίπεδα του γκρι. Για αυτό το λόγο, επιλέγουμε αρχικά μια εικόνα με μη ιδανικές συνθήκες για να παρατηρήσουμε μέγιστη διαφορά.

Όσον αφορά τη μετατροπή της σε ασπρόμαυρη, όλα τα pixel της εικόνας θα πρέπει να έχουν 2 δυνατές τιμές (μαύρο ή άσπρο). Για να αποφασίσουμε ποια θα πάρουν την κατάλληλη τιμή, ορίζουμε μια τιμή επιπέδου γκρι **threshold** (που καθορίζει την ευαισθησία της μετατροπής) τέτοια ώστε αν μετατρέψουμε την εικόνα μας σε grayscale, κάθε pixel που θα έχει τιμή μικρότερη ή μεγαλύτερη τιμή από αυτό θα μετατραπεί σε μαύρο ή άσπρο αντίστοιχα. Αυτή η τιμή εξαρτάται από το ιστογράμμα της κάθε εικόνας.

Υλοποίηση

Αρχικά, διαβάζουμε μια εικόνα της επιλογής μας. Μπορούμε να απεικονίσουμε το ιστογράμμα της ως διάγραμμα με τη συνάρτηση **imhist()**. Για να διορθώσουμε την αντίθεση της εικόνας, εφαρμόζουμε ισοστάθμιση με τη μέθοδο **histeq()** που παίρνει ως όρισμα την αρχική εικόνα και παράγει την ισοσταθμισμένη. Συγκρίνουμε οπότε τις δύο εικόνες με τα ιστογράμματά τους με subplot.

Η συνάρτηση **imhist()** μπορεί επίσης να εξάγει τα δεδομένα του ιστογράμματος στους πίνακες `binLocations` & `counts`. Ο πρώτος περιγράφει σε ποια επίπεδα του γκρι αντιστοιχεί τουλάχιστον ένα pixel, ενώ ο δεύτερος δείχνει αθροιστικά τη συνολική τιμή των pixel για κάθε επίπεδο του γκρι του πίνακα `binLocations`. Με βάση τα χαρακτηριστικά αυτά, υπολογίζουμε το `threshold` ως το επίπεδο του γκρι εκείνο όπου το συνολικό άθροισμα των τιμών των pixel να είναι ο μισός όλων των pixel. Με τον τρόπο αυτό, η διαδικασία δουλεύει για οποιαδήποτε εικόνα. Για τη μετατροπή της εικόνας σε ασπρόμαυρη, μετατρέπουμε πρώτα την εικόνα σε κλίμακα του γκρι έτσι ώστε να αποφύγουμε τα τρία χρωματικά κανάλια (οπότε δουλεύουμε με 2-διάστατους πίνακες) και διατρέχοντας όλα τα pixels της εικόνας θέτουμε τις νέες τιμές τους σύμφωνα με την διαδικασία που παρουσιάστηκε στη θεωρητική ανάλυση.

Μετάβαση στον κώδικα MATLAB ➡

Αποτελέσματα & Συμπεράσματα

Παρακάτω παραθέτουμε τις εικόνες και τα ιστογράμματα που προέκυψαν από τα ζητούμενα της εκφώνησης:

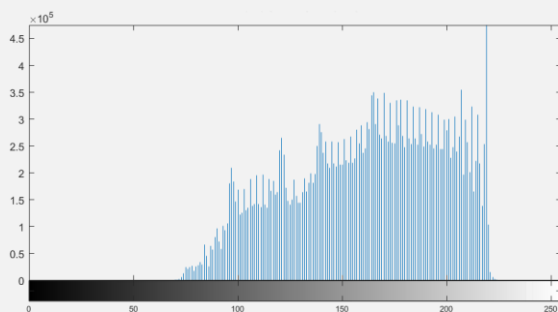
ΑΡΧΙΚΗ ΕΙΚΟΝΑ



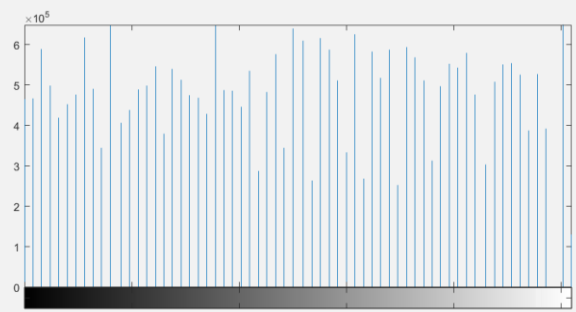
ΙΣΟΣΤΑΘΜΙΣΜΕΝΗ ΕΙΚΟΝΑ



ΙΣΤΟΓΡΑΜΜΑ ΑΡΧΙΚΗΣ ΕΙΚΟΝΑΣ



ΙΣΤΟΓΡΑΜΜΑ ΙΣΟΣΤΑΘΜΙΣΜΕΝΗΣ ΕΙΚΟΝΑΣ



ΑΡΧΙΚΗ ΕΙΚΟΝΑ**ΑΣΠΡΟΜΑΥΡΗ ΕΙΚΟΝΑ**

Παρατηρούμε ότι η αρχική εικόνα πάσχει από «κακή» αντίθεση καθώς το ιστόγραμμα της καταλαμβάνει μόνο ένα περιορισμένο εύρος των δυνατών επιπέδων του γκρι. Η ισοσταθμισμένη απ' την άλλη διορθώνει το πρόβλημα αυτό, κατανέμοντας τα *pixels* ομοιόμορφα σε όλα τα επίπεδα. Πρακτικά όπως φαίνεται παραπάνω, λαμβάνουμε μια πολύ περισσότερο - αισθητικά - ευχάριστη φωτογραφία με εντονότερα σκοτεινά αλλά και φωτεινά σημεία.

Σύμφωνα με το αρχικό ιστόγραμμα, εφαρμόζοντας τον αλγόριθμό που παρουσιάσαμε για τη μετατροπή της εικόνας σε ασπρόμαυρη λαμβάνουμε το παραπάνω αποτέλεσμα. Το *threshold* στη προκειμένη περίπτωση υπολογίστηκε ίσο με το επίπεδο γκρι 168. Παρατηρούμε ότι η μετατροπή μιας εικόνας σε ασπρόμαυρη δεν είναι ιδανική για φωτογραφίες με πολλά χρώματα. Αντιθέτως έχει βέλτιστα αποτελέσματα για εικόνες που έχουν εξαρχής λίγα χρώματα.

2 – MEDIAN & GAUSSIAN FILTERS

Στη δεύτερη άσκηση, ερχόμαστε για πρώτη φορά σε επαφή με φίλτρα ομαλοποίησης. Πιο συγκεκριμένα θα εφαρμόσουμε *median* & *gaussian* φίλτρα με διάφορα μεγέθη μάσκας και θα τα συγκρίνουμε μεταξύ τους.

Επισκόπηση & Θεωρητική Ανάλυση

Θεωρητικά, τα φίλτρα *media* & *gaussian* εξυπηρετούν τον ίδιο σκοπό. Να μειώσουν το θόρυβο «θολώνοντας» την εικόνα. Η κύρια διαφορά είναι ότι το *Gaussian* φίλτρο είναι γραμμικό σε αντίθεση με το *median*. Και τα δύο φίλτρα χρησιμοποιούν **μάσκες** που είναι ουσιαστικά η πληροφορία των γειτονικών *pixels* που χρησιμοποιείται για τον υπολογισμό της νέας τιμής του κάθε *pixel*. Όσο μεγαλύτερη είναι η μάσκα, τόσο εντονότερη γίνεται η θόλωση.

Υλοποίηση

Αρχικά υπολογίζουμε τις τιμές k_1 και k_2 με τον τρόπο που ζητήθηκε από την εκφώνηση. Για την ομάδα μας ισχύει:

$$k_1 = 9$$

$$k_2 = 11$$

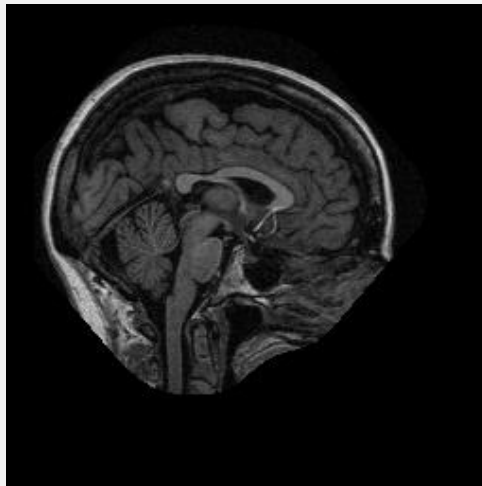
Διαβάζουμε την εικόνα "brain.gif". Με τη συνάρτηση **mefilt2()** δίνουμε ως όρισμα μια εικόνα και έναν πίνακα που δηλώνει το μέγεθος της μάσκας και παράγεται η τελική εικόνα με εφαρμοσμένο median φίλτρο. Για τη δημιουργία gaussian φίλτρου δίνουμε ως ορίσματα τα 'gaussian' και το μέγεθος του πίνακα στη συνάρτηση **fspecial()** και το παραγόμενο φίλτρο εισέρχεται ως όρισμα στη γενική μέθοδο **imfilter()** που το εφαρμόζει στην αρχική εικόνα. Στο τέλος συγκρίνουμε με subplot τις εφαρμογές των φίλτρων με 2 μεγέθη μασκών.

Μετάβαση στον κώδικα MATLAB ➔

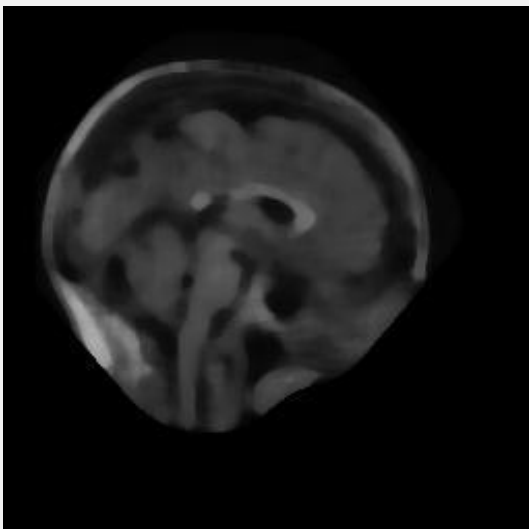
Αποτελέσματα & Συμπεράσματα

Παρακάτω εμφανίζονται όλα τα παραγόμενα αποτελέσματα των εφαρμογών των φίλτρων:

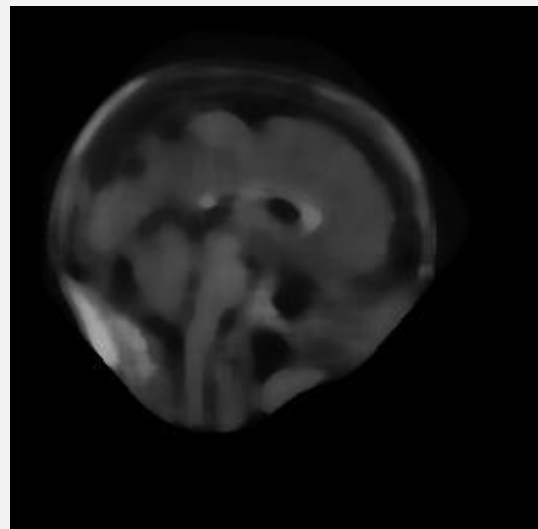
ΑΡΧΙΚΗ ΕΙΚΟΝΑ



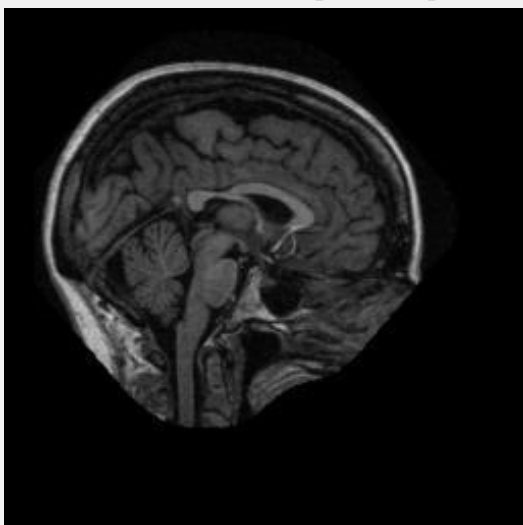
**MEDIAN FILTER
WITH MASK [k1 k1]**



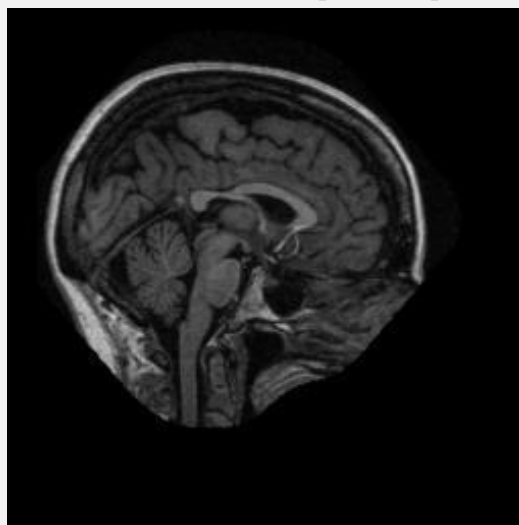
**MEDIAN FILTER
WITH MASK [k2 k2]**



**GAUSSIAN FILTER
WITH MASK [k1 k1]**



**GAUSSIAN FILTER
WITH MASK [k2 k2]**



Παρατηρούμε ότι η εφαρμογή και των δύο φίλτρων έχει ως αποτέλεσμα το θόλωμα της εικόνας. Όσο μεγαλύτερη είναι η μάσκα, τόσο πιο έντονο γίνεται το φαινόμενο καθώς περισσότερα γειτονικά pixels απαιτούνται για την εύρεση της νέας τιμής του pixel. Βλέπουμε επίσης ότι σε μια τέτοια βιοϊατρική εικόνα η διαφορά του Gaussian & Median φίλτρου είναι μεγάλη. Μεταξύ των δύο θα επιλέγαμε το Gaussian διότι διατηρεί περισσότερη λεπτομέρεια, κάτι που είναι σημαντικό στην επιστήμη αυτή.



3 – NOISE REDUCT. USING FILTERS

Η τρίτη άσκηση μοιάζει κάπως με την προηγούμενη, αυτή τη φορά όμως προσθέτουμε αρχικά θόρυβο στην εικόνα οπότε εφαρμόζουμε τα φίλτρα πάνω σε αυτή και συγκρίνουμε τα αποτελέσματα.

👁 Επισκόπηση & Θεωρητική Ανάλυση

Σε αυτό το ερώτημα δεν υπάρχει κάποιο καινούργιο φαινόμενο της θεωρίας. Αξίζει να σημειωθεί ότι ο λόγος που προσθέτουμε θόρυβο στην εικόνα είναι να μελετήσουμε ποιο φίλτρο προσφέρει την καλύτερη ελάττωση θορύβου χωρίς να έχουμε σημαντική μείωση της λεπτομέρειάς της.



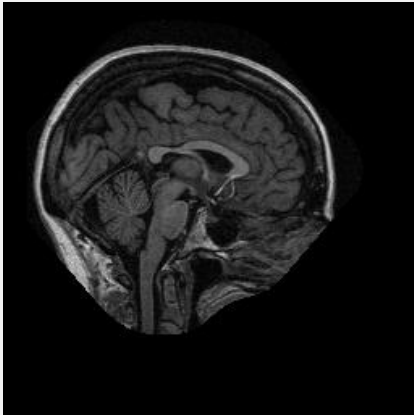
Υλοποίηση

Χρησιμοποιούμε και πάλι την brain.gif για να μελετήσουμε τα αποτελέσματα. Προσθέτουμε θόρυβο σε μια εικόνα χρησιμοποιώντας τη συνάρτηση `imnoise()` με μια συγκεκριμένη διασπορά. Μεγαλύτερη διασπορά σημαίνει πιο έντονο φαινόμενο θορύβου. Για το πείραμά μας επιλέξαμε μια πολύ μεγάλη τιμή ($=0.3$) για να ξεχωρίσει το φίλτρο με τα καλύτερα αποτελέσματα. Η εφαρμογή των φίλτρων έχει ακριβώς την ίδια εφαρμογή με τη προηγούμενη άσκηση, απλώς αυτή τη φορά χρησιμοποιούμε ως δεύτερο φίλτρο average αντί για gaussian.

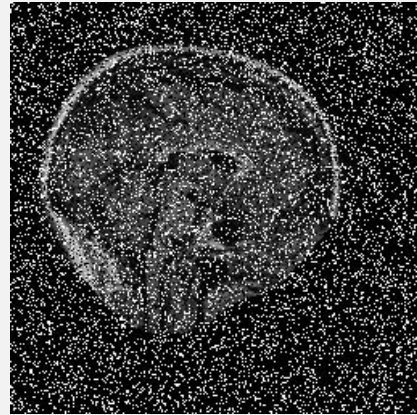
Μετάβαση στον κώδικα MATLAB ➡

Αποτελέσματα & Συμπεράσματα

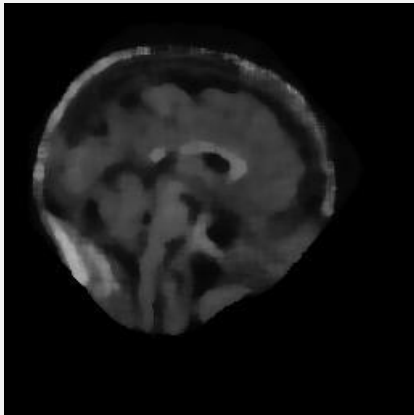
ΑΡΧΙΚΗ ΕΙΚΟΝΑ



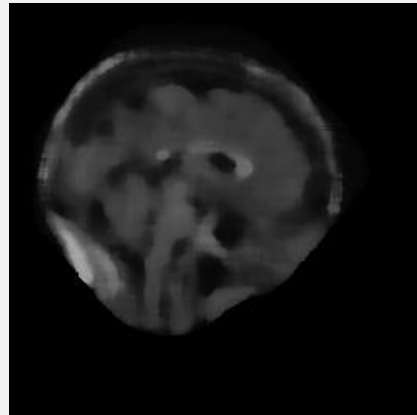
ΕΙΚΟΝΑ ΜΕ ΘΟΡΥΒΟ



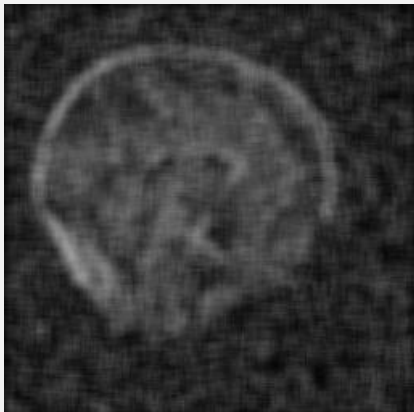
ΕΛΑΤΤΩΣΗ ΘΟΡΥΒΟΥ ΜΕ
MEDIAN [k1 k1]



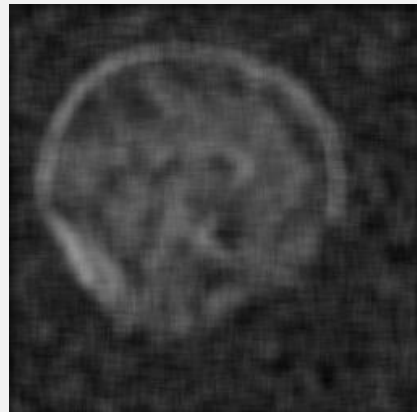
ΕΛΑΤΤΩΣΗ ΘΟΡΥΒΟΥ ΜΕ
MEDIAN [k2 k2]



ΕΛΑΤΤΩΣΗ ΘΟΡΥΒΟΥ ΜΕ
AVERAGE [k1 k1]



ΕΛΑΤΤΩΣΗ ΘΟΡΥΒΟΥ ΜΕ
AVERAGE [k2 k2]




Βλέπουμε πολύ ενδιαφέροντα αποτελέσματα. Ενώ η εφαρμογή average φίλτρου έχει αναμενόμενα αποτελέσματα για μια τόσο θορυβώδη εικόνα, το φίλτρο median έχει φέρει πολύ εντυπωσιακή ελάττωση θορύβου με μικρή – μόνο – απώλεια πληροφορίας.

4 – FLOATING PARTICLES

Στο τέταρτο και τελευταίο ερώτημα της εργαστηριακής άσκησης ασχολούμαστε με εικόνες περιβαλλοντικού ενδιαφέροντος και συγκεκριμένα η παρατήρηση κίνησης φερτών υλικών στη ροή ενός ποταμού χρησιμοποιώντας δύο διαδοχικά frames ενός video.

Υλοποίηση

Όπως συνήθως, ξεκινάμε διαβάζοντας τα δύο διαδοχικά frames. Τα μετατρέπουμε σε ασπρόμαυρες εικόνες καθώς η εφαρμογή των φίλτρων προϋποθέτει δυσδιάστατες εικόνες. Τέλος εφαρμόζουμε το median και το custom φίλτρο. Η στις εικόνες και παράγουμε τα αποτελέσματα που αναλύονται στη συνέχεια.

Μετάβαση στον κώδικα MATLAB 

Αποτελέσματα & Συμπεράσματα

ΑΡΧΙΚΟ FRAME 1



ΑΡΧΙΚΟ FRAME 2



MEDIAN FILTERED FRAME 1



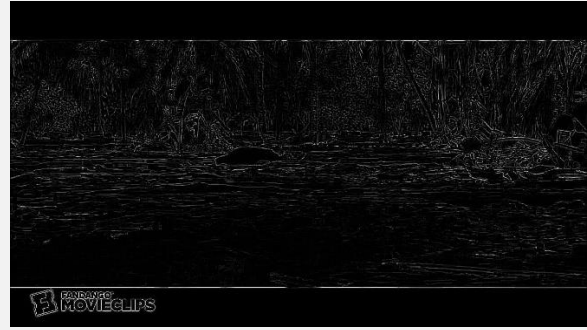
MEDIAN FILTERED FRAME 2



H FILTERED FRAME 1



H FILTERED FRAME 2



Παρατηρούμε ότι τα δυο φίλτρα παρουσιάζουν μεγάλη διαφορά στα αποτελέσματά τους. Το median φίλτρο είναι χαρακτηριστικό για την ελάττωση του θορύβου μιας εικόνας ενώ το custom φίλτρο H τονίζει τα πλαίσια κάθε αντικειμένου οπότε φαίνονται πιο ξεκάθαρα μεταξύ τους. Πιστεύουμε ότι για τα συγκεκριμένα frames του βίντεο κανένα από τα δύο φίλτρα δε βοηθούν αρκετά για την απομόνωση της πληροφορίας των φερτών.

Αξίζει να σημειώσουμε ότι το H είναι ένα γραμμικό χωρικό φίλτρο ενώ το median φίλτρο είναι μη-γραμμικό.

ΠΑΡΑΡΤΗΜΑ – ΚΩΔΙΚΑΣ

Σε περίπτωση που χρειαστεί ο κώδικας τον παραθέτουμε παρακάτω:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BASIC INFORMATION %
% Course: Digital Image Processing - Lab 2 %
% Deadline: 26-03-2019 %
% LAB31239720: Pantelis Karamailis, 2016030040 %
% Kostantinos Vlachos, 2016030041 %
% Mixalis Galanis, 2016030036 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Clearing things up
close all;
clearvars;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Exercise 1 - HISTOGRAM, EQUALIZATION, B-W CONVERSION %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1.1 - Displaying Histogram of an overexposed image
Initial_image_1 = imread('overexposed_image.jpg'); %Gathering Image
(Resolution: 3960x2640)
% 1.2 - Equalizing image
Equalized_image = histeq(Initial_image_1); %Applying Equalizer
figure(1);
subplot(2,2,1); imshow(Initial_image_1); title("IMAGE (BEFORE)");
subplot(2,2,2); imshow(Equalized_image); title("IMAGE (AFTER)");
subplot(2,2,3); imhist(Initial_image_1); title("HISTOGRAM (BEFORE)");
subplot(2,2,4); imhist(Equalized_image); title("HISTOGRAM (AFTER)");
```

```

% 1.3 - Converting image to Black & White
[counts,binLocations] = imhist(Initial_image_1); %Gathering Histogram Data
%Finding Threshold
half_channels = sum(counts)/2;
sum_of_counts = 0; Threshold = 0;
for i = 1:length(counts)
    sum_of_counts = sum_of_counts + counts(i);
    if (sum_of_counts >= half_channels)
        Threshold = i;
        break;
    end
end
%Conversion
bw_image_1 = rgb2gray(Initial_image_1);
for i = 1 : length(bw_image_1(:,1))
    for j = 1 : length(bw_image_1(1,:))
        if (bw_image_1(i,j) < Threshold)
            bw_image_1(i,j) = 0;
        else
            bw_image_1(i,j) = 255;
        end
    end
end
end
imwrite(bw_image_1, 'bw_image.jpg'); %Saving image to drive
%Showing side by side comparison
figure(2);
subplot(2,1,1); imshow(Initial_image_1); title("IMAGE (BEFORE)");
subplot(2,1,2); imshow(bw_image_1); title("BLACK & WHITE IMAGE (AFTER)");
imwrite(X1, grayscale_map, 'image_grayscale.png');

```

Μετάβαση στα αποτελέσματα της άσκησης 1 ➡

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Exercise 2 - MEDIAN & GAUSSIAN FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Initial_image_2 = imread('brain.gif'); %Gathering Image (Resolution: 256x256)
k1 = 7 + 2 + 0; %k1 = 9
k2 = k1 + 2; %k2 = 11
%Applying median filters
Filtered_image_2_1 = medfilt2(Initial_image_2, [k1 k1]);
Filtered_image_2_2 = medfilt2(Initial_image_2, [k2 k2]);
%Applying gaussian filters
h_1 = fspecial('gaussian', [k1 k1]);
h_2 = fspecial('gaussian', [k2 k2]);
Filtered_image_2_3 = imfilter(Initial_image_2, h_1);
Filtered_image_2_4 = imfilter(Initial_image_2, h_2);
%Displaying and comparing images
figure(3);
subplot(2,1,1); imshow(Initial_image_2); title("ORIGINAL IMAGE");
subplot(4,2,5); imshow(Filtered_image_2_1); title("IMAGE WITH MEDIAN FILTER
[k1 k1]");

```

```
subplot(4,2,6); imshow(Filtered_image_2_2); title("IMAGE WITH MEDIAN FILTER
[k2 k2]");
subplot(4,2,7); imshow(Filtered_image_2_3); title("IMAGE WITH GAUSSIAN FILTER
[k1 k1]");
subplot(4,2,8); imshow(Filtered_image_2_4); title("IMAGE WITH GAUSSIAN FILTER
[k2 k2]");
```

Μετάβαση στα αποτελέσματα της άσκησης 2 ➡

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Exercise 3 - NOISE & NOISE REDUCTION USING FILTERS %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Initial_image_3 = imread('brain.gif'); %Gathering Image (Resolution: 256x256)
%k1, k2 remain the same
%Adding Noise to the Image
Noise_image_3 = imnoise(Initial_image_3,'salt & pepper', 0.3); % very high
noise density
%Applying noise reduction filter using median
Filtered_noise_image_3_1 = medfilt2(Noise_image_3, [k1 k1]);
Filtered_noise_image_3_2 = medfilt2(Noise_image_3, [k2 k2]);
%Applying noise reduction filter using average
h_3 = fspecial('average', [k1 k1]);
h_4 = fspecial('average', [k2 k2]);
Filtered_noise_image_3_3 = imfilter(Noise_image_3, h_3);
Filtered_noise_image_3_4 = imfilter(Noise_image_3, h_4);
%Displaying and comparing images
figure(4);
subplot(2,2,1); imshow(Initial_image_3); title("ORIGINAL IMAGE");
subplot(2,2,2); imshow(Noise_image_3); title("IMAGE WITH NOISE");
subplot(4,2,5); imshow(Filtered_noise_image_3_1); title("DENOISED IMAGE WITH
MEDIAN FILTER - [k1 k1]");
subplot(4,2,6); imshow(Filtered_noise_image_3_2); title("DENOISED IMAGE WITH
MEDIAN FILTER - [k2 k2]");
subplot(4,2,7); imshow(Filtered_noise_image_3_3); title("DENOISED IMAGE WITH
AVERAGE FILTER - [k1 k1]");
subplot(4,2,8); imshow(Filtered_noise_image_3_4); title("DENOISED IMAGE WITH
AVERAGE FILTER - [k2 k2]");
```

Μετάβαση στα αποτελέσματα της άσκησης 3 ➡

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Exercise 4 - FLOATING PARTICLES %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
frame_4_1 = imread('initial_frame_1.jpg'); %Gathering Image (Resolution:
720x1080)
frame_4_2 = imread('initial_frame_2.jpg'); %Gathering Image (Resolution:
720x1080)
%k1, k2 remain the same
%Converting images to grayscale
gray_frame_4_1 = rgb2gray(frame_4_1);
```

```

gray_frame_4_2 = rgb2gray(frame_4_2);
%Applying noise reduction filter using median
Filtered_frame_4_1 = medfilt2(gray_frame_4_1, [k1 k1]);
Filtered_frame_4_2 = medfilt2(gray_frame_4_2, [k1 k1]);
%Applying h filter
h = [1 1 1; 1 -8 1; 1 1 1];
Filtered_frame_4_3 = imfilter(gray_frame_4_1, h);
Filtered_frame_4_4 = imfilter(gray_frame_4_2, h);
%Displaying and comparing images
figure(5);
subplot(3,2,1); imshow(gray_frame_4_1); title("ORIGINAL FRAME 1");
subplot(3,2,2); imshow(gray_frame_4_2); title("ORIGINAL FRAME 2");
subplot(3,2,3); imshow(Filtered_frame_4_1); title("MEDIAN FILTERED FRAME 1");
subplot(3,2,4); imshow(Filtered_frame_4_2); title("MEDIAN FILTERED FRAME 2");
subplot(3,2,5); imshow(Filtered_frame_4_3); title("H FILTERED FRAME 1");
subplot(3,2,6); imshow(Filtered_frame_4_4); title("H FILTERED FRAME 2");

```

Μετάβαση στα αποτελέσματα της άσκησης 4 ➡