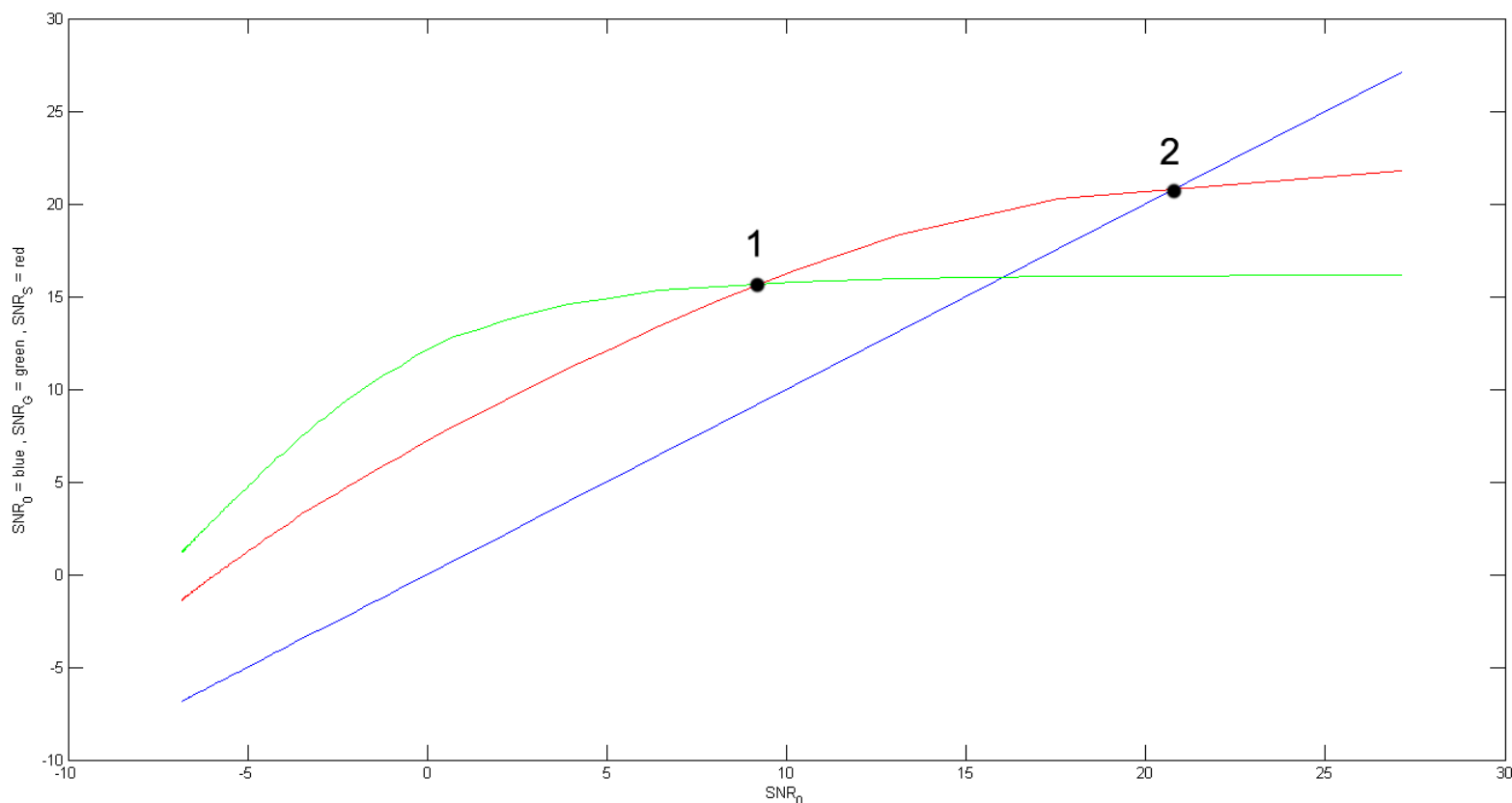




Ονοματεπώνυμο: Δανασής Παναγιώτης
Α.Μ.: 03109004
Τμήμα: Πέμπτης 15:15-17:00

Απάντηση στα θεωρητικά ερωτήματα:



Σχήμα1: Γραφική παράσταση SNR/SNR_0
' SNR_0 = blue , SNR_G = green , SNR_S = red'

Το διάγραμμα αυτό μας δείχνει την απόδοση των δύο φίλτρων ανάλογα με την διασπορά του θορύβου που έχουμε εφαρμόσει. Παρατηρούμε ότι για μικρές τιμές διασποράς, το φίλτρο Gaussian δίνει καλύτερα αποτελέσματα καθώς η γραφική παράσταση του SNR_G προς το SNR_0 βρίσκεται πιο πάνω από το SNR_0/SNR_0 και το SNR_S/SNR_0 (μέχρι το σημείο 1). Από κάποια τιμή της διασποράς θορύβου και μετά, καλύτερα αποτελέσματα δίνει το βαθυπερατό φίλτρο (από το σημείο 1 μέχρι το σημείο 2), ενώ για μεγάλες τιμές διασποράς, το φιλτράρισμα δεν έχει θετικά αποτελέσματα με κανένα από τα δύο φίλτρα.

Η περισσότερη πληροφορία της εικόνας βρίσκεται στις χαμηλές συχνότητες, συνεπώς αυτό που προσπαθούμε να κάνουμε με το φιλτράρισμα είναι να κόψουμε τον θόρυβο που βρίσκεται στις υψηλές συχνότητες έτσι ώστε να μην αλλοιώσουμε το περιεχόμενο της εικόνας.

Στις περιπτώσεις λοιπόν που έχουμε πολύ θόρυβο στις υψηλές συχνότητες, μας συμφέρει το φιλτράρισμα, καθώς κόβοντας τον θόρυβο δεν χαλάμε την εικόνα μας. Όμως στις περιπτώσεις που ο θόρυβος βρίσκεται και αυτός στις χαμηλές συχνότητες, με τις παραπάνω τεχνικές που είδαμε, δεν ωφελεί το φιλτράρισμα, καθώς για να κόψουμε τον θόρυβο θα χάσουμε και πολύτιμη πληροφορία από την εικόνα.

Παράθεση κώδικα Matlab:

```
function J = freqfilt( I, h )
    % Η συνάρτηση φιλτράρει την εικόνα I με το φίλτρο h.
    I = im2double( I );
    [M, N] = size( I );
    F = fft2( I, M+10, N+10 );
    H = fft2( h, M+10, N+10 );
    JF = abs( ifft2( F.*H ) );
    JF = JF( 6:end-5, 6:end-5 );
    J = JF;
End
```

```
function [SNR_0, SNR_G, SNR_S] = denoise( I, s )
    % Προσθήκη θορύβου.
    I = im2double( I );
    J = imnoise( I, 'gaussian', 0, (s^2) );

    % Δημιουργία ενούς 2-D φίλτρου Gaussian με  $\sigma=2$ .
    hg = fspecial( 'gaussian', 11, 2 );

    % Δημιουργία βαθυπερατού φίλτρου.
    [f1,f2] = freqspace( 11, 'meshgrid' );
    Hd = zeros( 11 );
    Hd( sqrt( f1.^2 + f2.^2 ) < 0.5 ) = 1;
    hl = fsamp2( Hd );

    % Εφαρμογή φίλτρων.
    Jg = freqfilt( J, hg );
    Jl = freqfilt( J, hl );

    % Υπολογισμός snr.
    SNR_0 = snr( I, J );
    SNR_G = snr( I, Jg );
    SNR_S = snr( I, Jl );
End
```

```
function [] = lab2()
    % Φόρτωση εικόνας.
    I = imread( 'house.tif' );

    y_0 = [];
    y_G = [];
    y_S = [];
```

```

% Apply noise and filters
for s = 0.01:0.02:2
    [SNR_0, SNR_G, SNR_S] = denoise( l, s );
    y_0 = [ y_0 SNR_0 ];
    y_G = [ y_G SNR_G ];
    y_S = [ y_S SNR_S ];
end

x = y_0;

% Εμφάνιση αποτελεσμάτων.
plot( x, y_0, 'blue' );
xlabel('SNR_0');
ylabel('SNR_0 = blue , SNR_G = green , SNR_S = red');
hold all
plot( x, y_G, 'green' );
hold all
plot( x, y_S, 'red' );
end

```