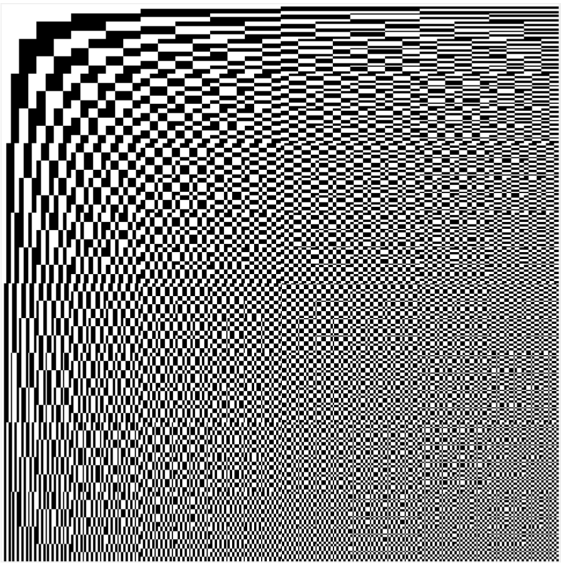
Parichehr Moradi
Biomedical Engineering Department,
Engineering Faculty, University of Isfahan,
Isfahan, Iran

INVESTIGATING THE EFFECT OF ORDERED HADAMRD TRANSFORM ON SAMPLE IMAGE

In this practice, we want to see the properties of the Hadamard domain in image processing. And also using some lowpass, highpass, bandpass, and bandstop non-ideal filters. In this work I have used non-ideal Laplacian filters with this characteristics for low-pass filters: sigma = 10 , 5 and 0.1 .And we know that we can create a highpass filter using 1-lowpass one. As we want to investigate the effect of two filters of every form so we need to create 3 lowpass and 3 highpass filters to create 2 bandpass and bandstop filters as following: bandstop filter = lowpass filter + high pass filter, as the stop frequency of lowpass filter must be smaller than pass frequency of the highpass one. So we can produce our bandstop filters as follow: bandstop1 = lowpass1 + highpass2 and bandstop2 = lowpass2 + highpass3. And then we can create our bandpass filters with 1-bandstop ones.

In the following we can see the results step by step:

1- Producing ordered Hadamard transform matrix:



2- Samples of our lowpass, highpass, bandpass and bandstop filters:

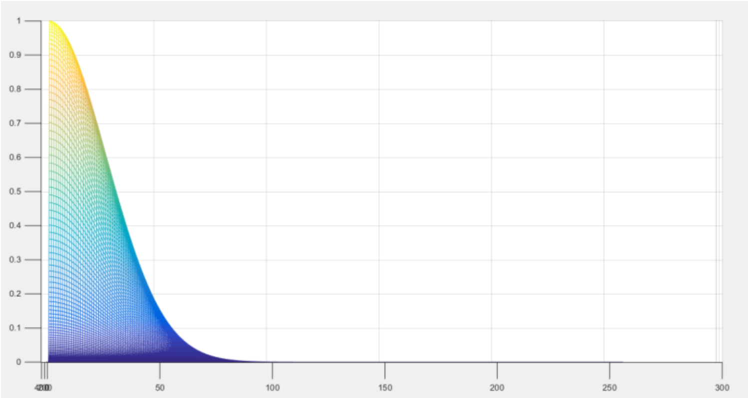


Fig.a : Laplacian lowpass filter with sigma=10

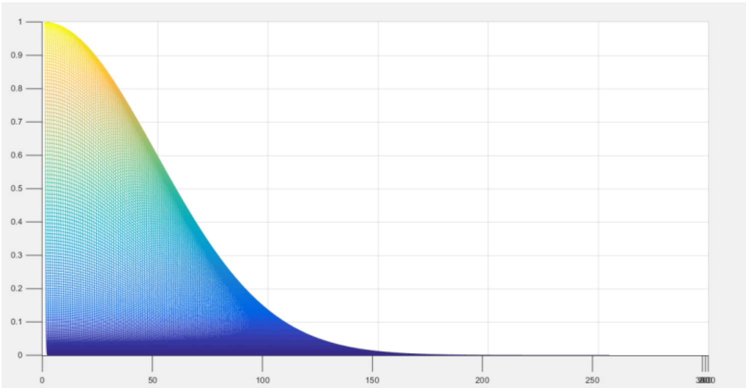


Fig.b : Laplacian lowpass filter with sigma=5

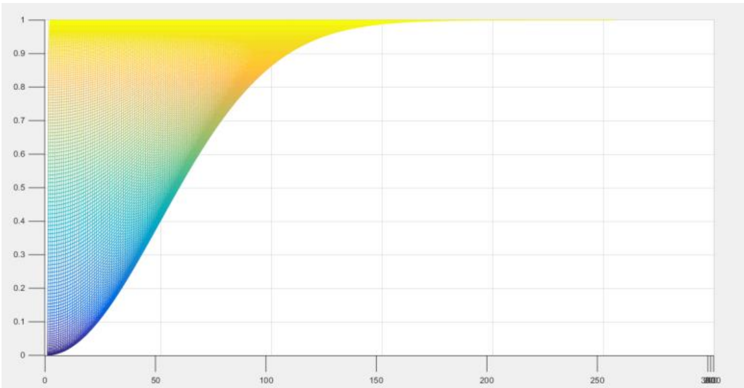


Fig.c : Laplacian highpass filter

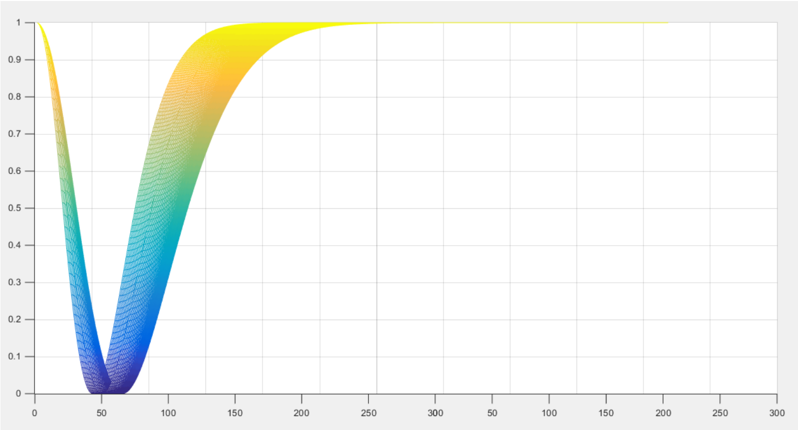


Fig.d : Laplacian bandstop filter

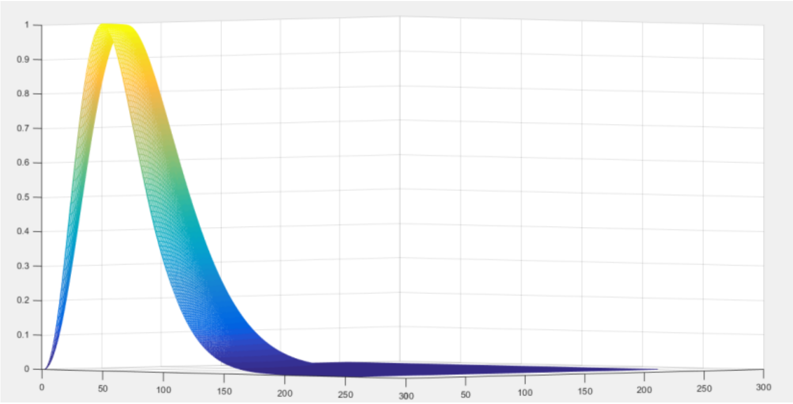


Fig.e : Laplacian bandpass filter

3- At last the output of these filters:

3.1- Two lowpass filters:

3.2- Two highpass filters:

3.3- Two bandstop filters:

4.4- Two bandpass filters:

Appendix:

Matlab code:

```
clear all; close all; clc;
I = im2double(imread('cameraman.tif'));
N = 256;  % Length of Walsh (Hadamard) functions
hadamardMatrix = hadamard(N);
HadIdx = 0:N-1;                          % Hadamard index
M = log2(N)+1;                           % Number of bits to represent the index
binHadIdx = fliplr(dec2bin(HadIdx,M))-'0'; % Bit reversing of the binary index
binSeqIdx = zeros(N,M-1);                  % Pre-allocate memory
for k = M:-1:2
    % Binary sequency index
    binSeqIdx(:,k) = xor(binHadIdx(:,k),binHadIdx(:,k-1));
end
SeqIdx = binSeqIdx*pow2((M-1:-1:0)');    % Binary to integer sequency index
W = hadamardMatrix(SeqIdx+1,:); % 1-based indexing
J = W*I*W;
n = 512;
[e,f] = meshgrid(-1:1/((n-1)/2):1,-1:1/((n-1)/2):1);
LOW1 = 1/(2*pi*10^2)*exp(-(e.^2+f.^2)/2*10^2);        % sigma=10
LOW1 = LOW1(257:512,257:512);
LOW1 = (LOW1-min(LOW1(:)))/(max(LOW1(:)-min(LOW1(:))));
LOW2 = 1/(2*pi*5^2)*exp(-(e.^2+f.^2)/2*5^2);        % sigma=5
LOW2 = LOW2(257:512,257:512);
LOW2 = (LOW2-min(LOW2(:)))/(max(LOW2(:)-min(LOW2(:))));
LOW3 = 1/(2*pi*0.1^2)*exp(-(e.^2+f.^2)/2*0.1^2);        % sigma=0.1
LOW3 = LOW3(257:512,257:512);
LOW3 = (LOW3-min(LOW3(:)))/(max(LOW3(:)-min(LOW3(:))));
HI1 = 1 - LOW1;HI1 = (HI1-min(HI1(:)))/(max(HI1(:)-min(HI1(:))));
HI2 = 1 - LOW2;HI2 = (HI2-min(HI2(:)))/(max(HI2(:)-min(HI2(:))));
HI3 = 1 - LOW3;HI3 = (HI3-min(HI3(:)))/(max(HI3(:)-min(HI3(:))));
BS1 = LOW1 + HI2 ; BS1 = (BS1-min(BS1(:)))/(max(BS1(:)-min(BS1(:))));
BS2 = LOW2 + HI3 ; BS2 = (BS2-min(BS2(:)))/(max(BS2(:)-min(BS2(:))));
BP1 = 1 - BS1 ; BP1 = (BP1-min(BP1(:)))/(max(BP1(:)-min(BP1(:))));
BP2 = 1 - BS2 ; BP2 = (BP2-min(BP2(:)))/(max(BP2(:)-min(BP2(:))));
YL1 = LOW1.*J ; figure,imshow(W*YL1*W/256^2);
YL2 = LOW2.*J ; figure,imshow(W*YL2*W/256^2);
YH1 = HI1.*J ; figure,imshow(W*YH1*W/256^2);
YH2 = HI2.*J ; figure,imshow(W*YH2*W/256^2);
YBP1 = BP1.*J ; figure,imshow(W*YBP1*W/256^2);
YBP2 = BP2.*J ; figure,imshow(W*YBP2*W/256^2);
YBS1 = BS1.*J ; figure,imshow(W*YBS1*W/256^2);
YBS2 = BS2.*J ; figure,imshow(W*YBS2*W/256^2);
```