Table of Contents

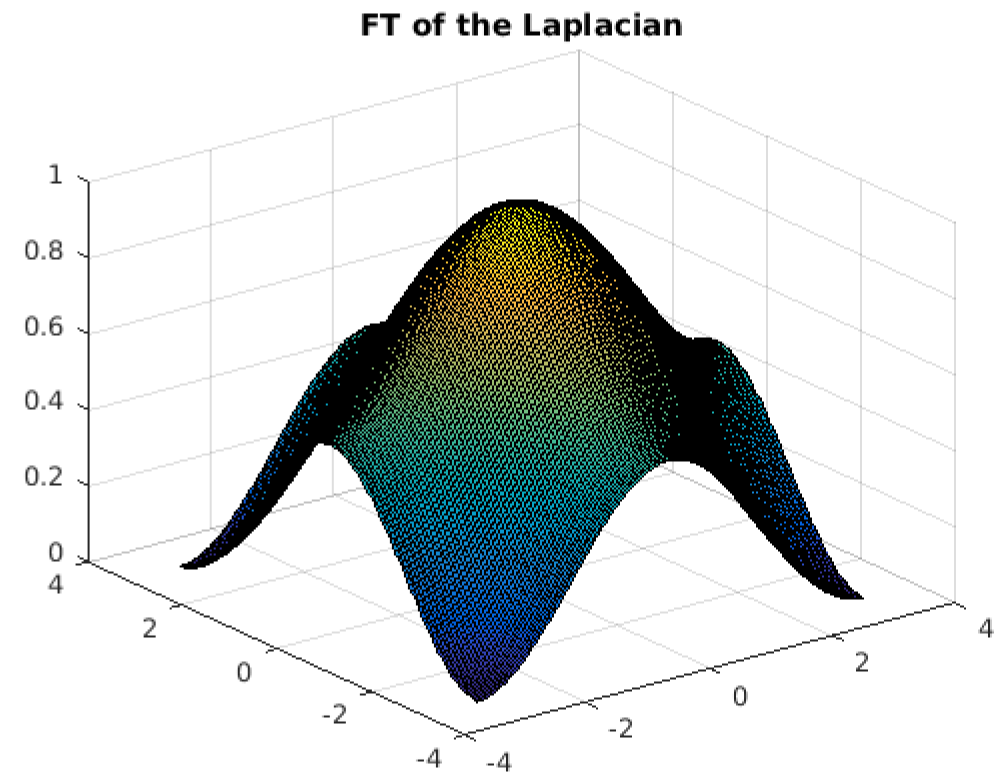
ECE 418 Digital Video Problems: Multidimensional Signal Processing DongKyu Kim

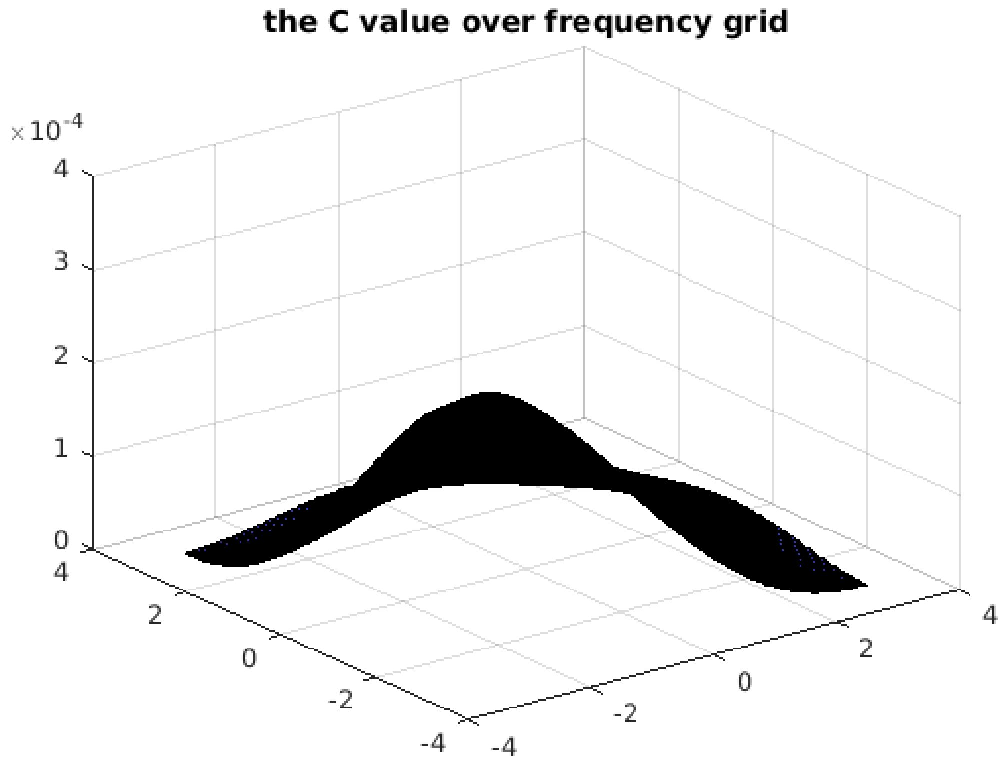
```
clear all; close all; clc;
```

1

(a)

```
l_hat = 1/8*[0,1,0;1,-4,1;0,1,0];

N1 = 128; N2 = 128;
C = zeros(N1,N2);
L_hat = C;
for i = 1:N1
  for ii = 1:N2
    L_hat(i,ii) = DFT_2(l_hat,i,ii,N1,N2);
    C(i,ii) = -L_hat(i,ii)/(i*i+ii*ii);
  end
end

axis = linspace(-pi,pi,N1);
figure(1);
surf(axis,axis,abs(L_hat));
title('FT of the Laplacian');

figure(2);
surf(axis,axis,abs(C));
title('the C value over frequency grid');
% I think I did something wrong here.
```


FT of the Laplacian
1
0.8
0.6
0.4
0.2
0
4
2
0
-2
-4
-4
-2
0
2
4

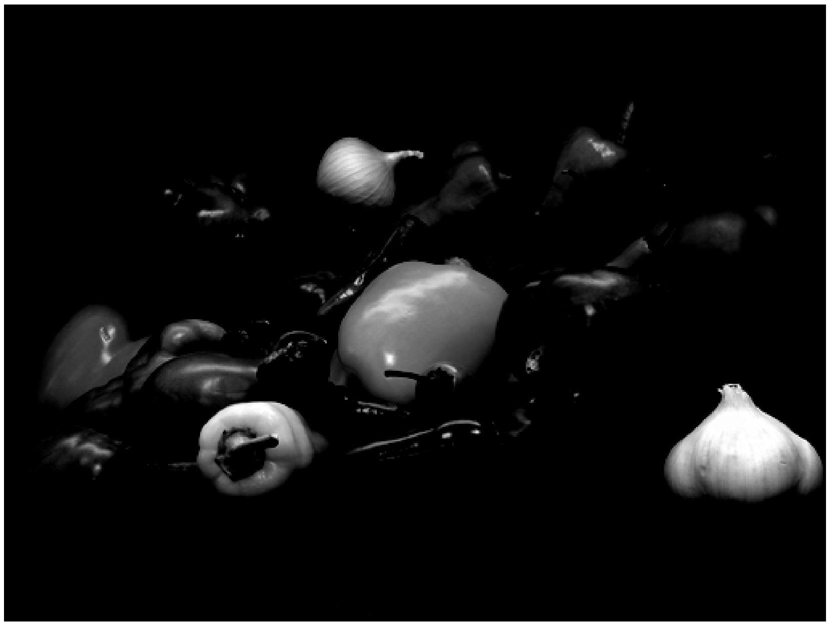

the C value over frequency grid
$\times 10^{-4}$
4
3
2
1
0
4
2
0
-2
-4
-4
-2
0
2
4

(b)

```
N2_n = [2,4,8,16,32,64];
N1_n = N2_n;
C_store = zeros(length(N1_n));
for k = 1:length(N1_n)
  for kk = 1:length(N1_n)
    N1_t = N1_n(k);
    N2_t = N2_n(k);
    C_t = zeros(N1_t,N2_t);
    temp = zeros(N1_t,N2_t);
    for i = 1:N1_t
      for ii = 1:N2_t
        temp(i,ii) = DFT_2(l_hat,i,ii,N1_t,N2_t);
        C_t(i,ii) = -temp(i,ii)/(i*i+ii*ii);
      end
    end
    C_store(k,kk) = mean(mean(C_t));
  end
end

% I am not too sure about the exact relationship, but it seems like C
 is dependent on N1.
% However my C seems to be wrong. C is dependent on k.
```

(c)

```
peppers = imread('peppers.png');
peppers = im2double(rgb2gray(peppers));
peppers = 2*(peppers-1/2);
imshow(peppers);
```

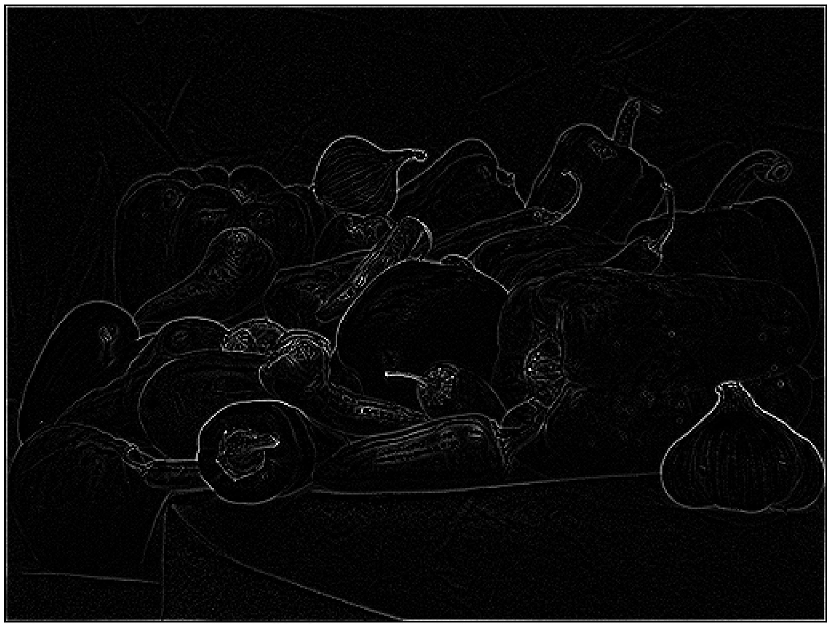


(d)

```
l_hat_norm = norm(l_hat);
l_hat_H_inf = max(max(real(L_hat))); % H inf norm is the peak gain.
% I can't do the question for l. But I am assuming it is higher than
 l_hat because it conveys more information.
% if I try what I did in (g)
laplacian_td = fspecial('laplacian'); % well this is another 3 by 3
 laplacian thing.....
l_norm = norm(laplacian_td); % l_norm is higher
laplacian_fd = fft2(double(laplacian_td),512,512);
l_norm_H_inf = max(max(real(laplacian_fd))); %H inf norm is higher
 than norm...
```

(e)

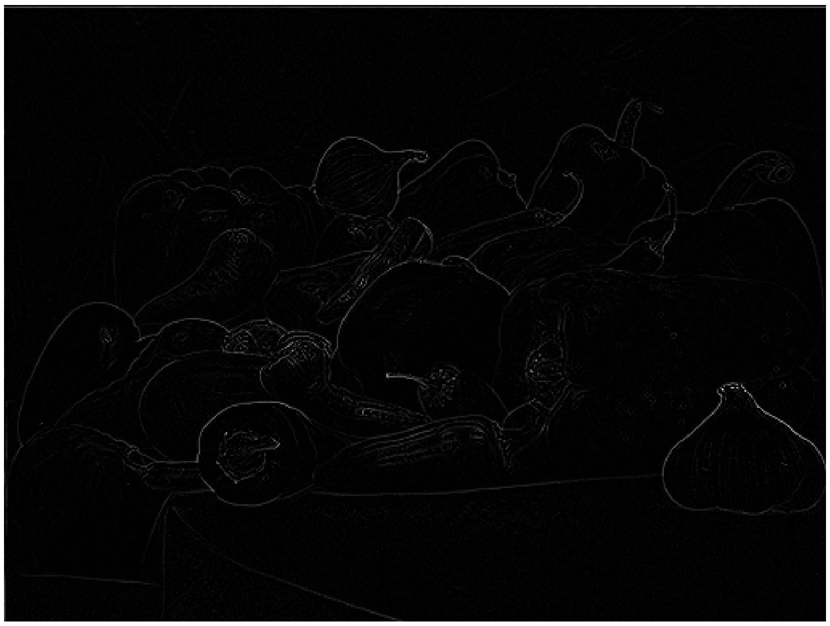
```
peppers_n = padarray(peppers,[1 1]); % single padding because mask is
 centered at origin.
peppers_filtered = imfilter(peppers_n,l_hat);
peppers_l_hat = 8*peppers_filtered;% scale by 8 because largest value
 is 0.125
figure;
imshow(peppers_l_hat);
```



(f)

```
peppers_padded = padarray(peppers,[64 0]);
L_hat2 = zeros(512,512);
for i = 1:512
    for ii = 1:512
        L_hat2(i,ii) = DFT_2(l_hat,i,ii,512,512);
    end
end
peppers_fft = fft2(peppers_padded);
peppers_filtered2 = peppers_fft.*L_hat2;
peppers_filtfin = ifft2(peppers_filtered2);
peppers_filtfin = peppers_filtfin(65:(end - 64),:);
scale = 1./max(max(real(peppers_filtfin)))
peppers_l_hat_freq = scale*real(peppers_filtfin);
figure;
imshow(peppers_l_hat_freq);

% I think we get the same result because we zero padded our edges are
 both weird.
% but I didn't really get the exactly same image..
```

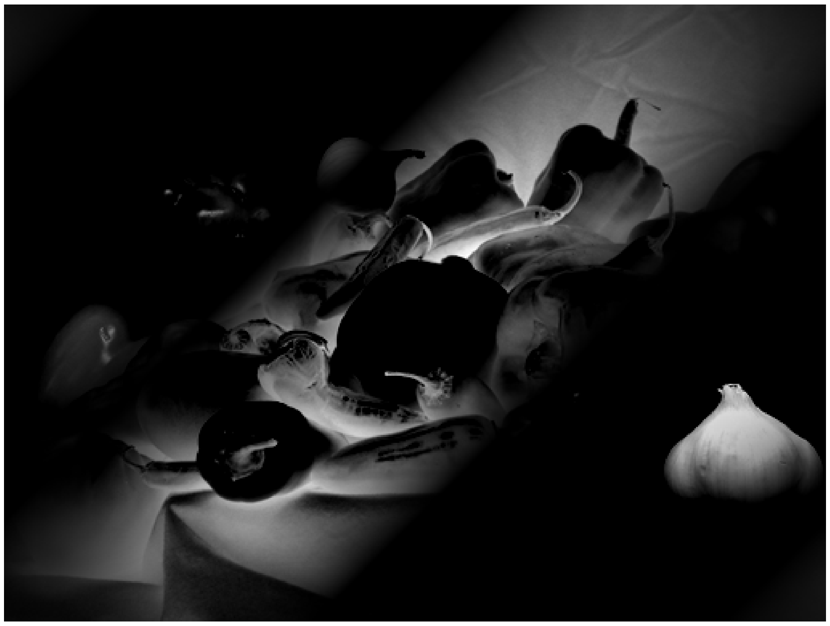
```
scale =

    3.4938
```



(g)

I couldn't really get the exact laplacian working because my C is weird from (b)

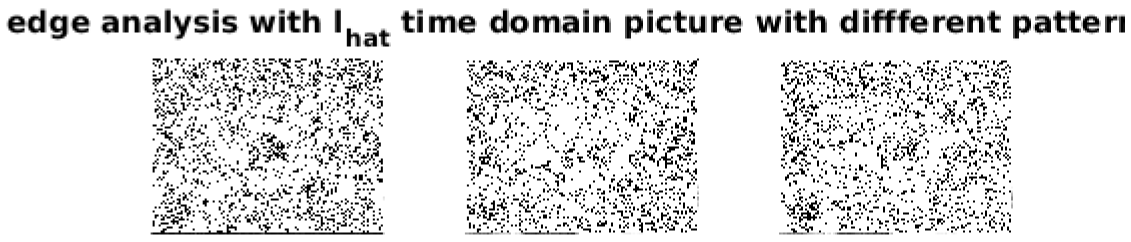
```
laplacian_td = fspecial('laplacian');
laplacian_fd = fft2(double(laplacian_td),512,512);
peppers_laplace = peppers_padded.*laplacian_fd;
peppers_laplace = peppers_laplace(65:(end - 64),:);
scale = 1./max(max(real(peppers_laplace)));
figure;
peppers_real_laplace = scale*real(peppers_laplace);
imshow(peppers_real_laplace);
```



(h)

```
pattern1 = [1,0,-1;1,0,-1;1,0,-1]; %left right
pattern2 = [1,1,0;1,0,-1;0,-1,-1]; %diagonal /
pattern3 = [0,-1,-1;1,0,-1;1,1,0]; %diagonal \
TH1_1 = conv2(pattern1,peppers_l_hat);
TH1_2 = conv2(pattern2,peppers_l_hat);
TH1_3 = conv2(pattern3,peppers_l_hat);
TH2_1 = conv2(pattern1,peppers_l_hat_freq);
TH2_2 = conv2(pattern2,peppers_l_hat_freq);
TH2_3 = conv2(pattern3,peppers_l_hat_freq);
TH3_1 = conv2(pattern1,peppers_real_laplace);
TH3_2 = conv2(pattern2,peppers_real_laplace);
TH3_3 = conv2(pattern3,peppers_real_laplace);
T = 0.02; % this is a value in which half the points are classified as
 edge points in TH1_1

for i = 1:384
    for ii = 1:512
        if abs(TH1_1(i,ii)) > T
            TH1_1(i,ii) = sign(TH1_1(i,ii));
        else
            TH1_1(i,ii) = 0;
        end
        if abs(TH1_2(i,ii)) > T
            TH1_2(i,ii) = sign(TH1_2(i,ii));
        else
            TH1_2(i,ii) = 0;
        end
        if abs(TH1_3(i,ii)) > T
            TH1_3(i,ii) = sign(TH1_3(i,ii));
        else
            TH1_3(i,ii) = 0;
        end
        if abs(TH2_1(i,ii)) > T
            TH2_1(i,ii) = sign(TH2_1(i,ii));
        else
            TH2_1(i,ii) = 0;
        end
        if abs(TH2_2(i,ii)) > T
            TH2_2(i,ii) = sign(TH2_2(i,ii));
        else
            TH2_2(i,ii) = 0;
        end
        if abs(TH2_3(i,ii)) > T
            TH2_3(i,ii) = sign(TH2_3(i,ii));
        else
            TH2_3(i,ii) = 0;
        end
        if abs(TH3_1(i,ii)) > T
            TH3_1(i,ii) = sign(TH3_1(i,ii));
        else
            TH3_1(i,ii) = 0;
        end
        if abs(TH3_2(i,ii)) > T
            TH3_2(i,ii) = sign(TH3_2(i,ii));
        else
            TH3_2(i,ii) = 0;
        end
        if abs(TH3_3(i,ii)) > T
            TH3_3(i,ii) = sign(TH3_3(i,ii));
        else
            TH3_3(i,ii) = 0;
        end
    end
end

% I put in the absolute value here because having an absolute value
 bigger than the threhold
% means it's an edge anyways.
figure;
subplot(1,3,1);
imshow(abs(TH1_1));
subplot(1,3,2);
imshow(abs(TH1_2));
title('edge analysis with l_{hat} time domain picture with diffferent
 patterns');
subplot(1,3,3);
imshow(abs(TH1_3));

figure;
subplot(1,3,1);
```

```
imshow(abs(TH2_1));
subplot(1,3,2);
imshow(abs(TH2_2));
title('edge analysis with l_{hat} freq domain picture with diffferent
 patterns');
subplot(1,3,3);
imshow(abs(TH2_3));

figure;
subplot(1,3,1);
imshow(abs(TH3_1));
subplot(1,3,2);
imshow(abs(TH3_2));
title('edge analysis with l_{given} freq domain picture with
 diffferent patterns');
subplot(1,3,3);
imshow(abs(TH3_3));

% They are all in general bad. At least for TH2_1~3 you can kind of
 tell that it's the same picture.
% Pattern 2 seems to work the best.
% l_hat in frequency domain picture works the best.
```



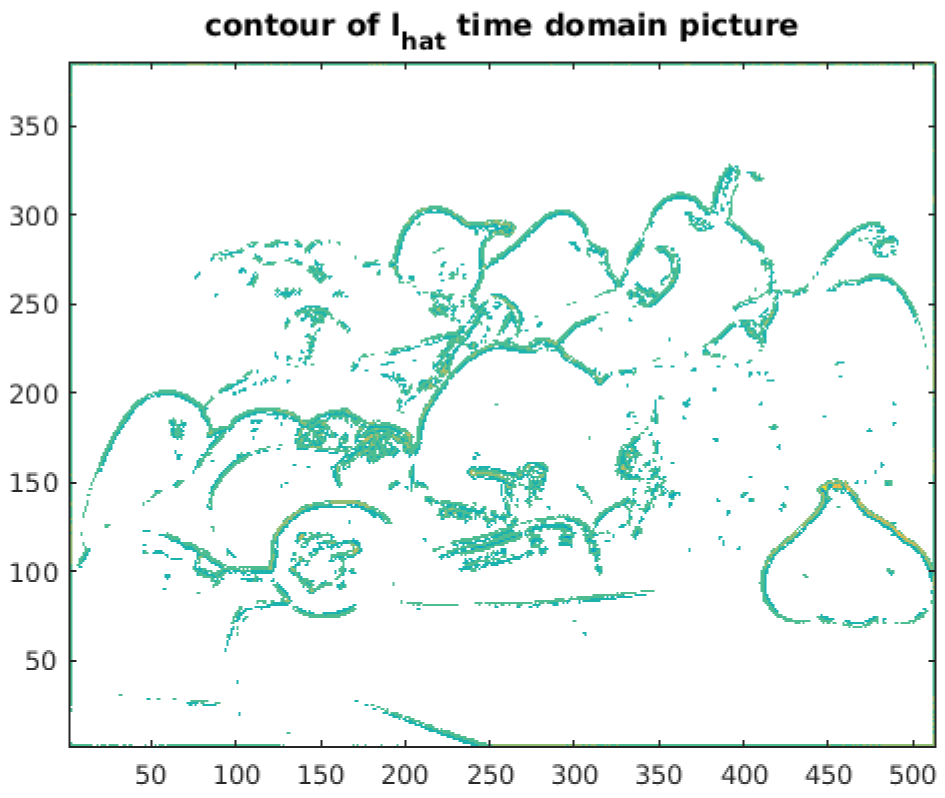
edge analysis with I_{hat} freq domain picture with diffferent patterr

edge analysis with I_{given} freq domain picture with diffferent patterr

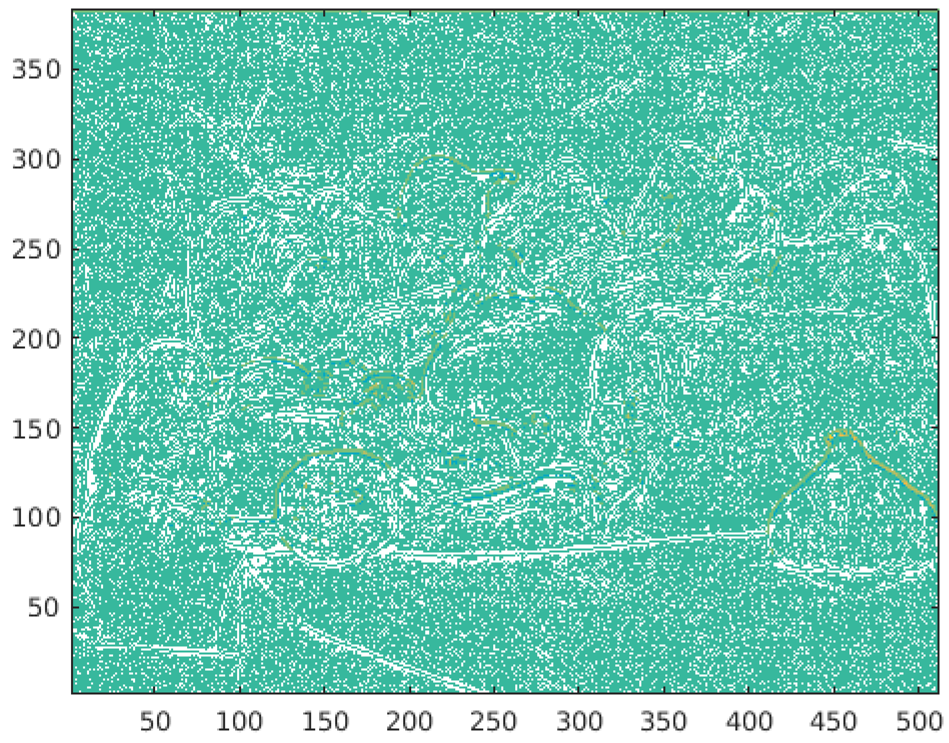
(i)

my final pictures are already scaled to -1 ~ 1

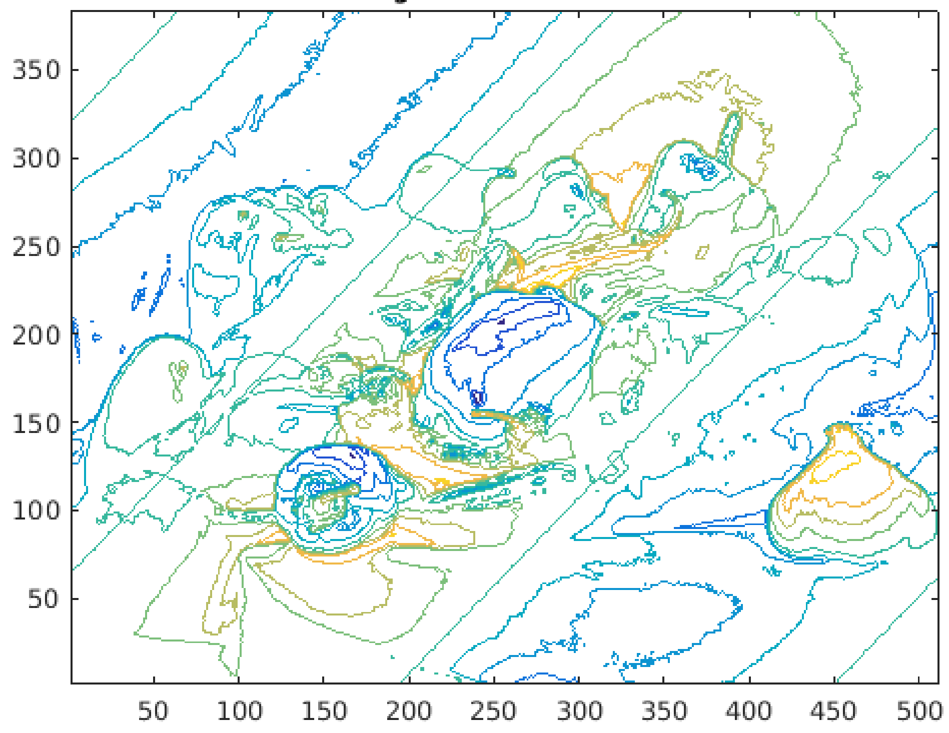
```
laplace1 = (peppers_l_hat+1)/2;
laplace2 = (peppers_l_hat_freq+1)/2;
laplace3 = (peppers_real_laplace+1)/2;
figure;
contour(flipud(laplace1))
title('contour of l_{hat} time domain picture')
figure;
contour(flipud(laplace2));
title('contour of l_{hat} freq domain picture')
figure;
contour(flipud(laplace3));
title('contour of l_{given} freq domain picture')
```



contour of I_{hat} freq domain picture



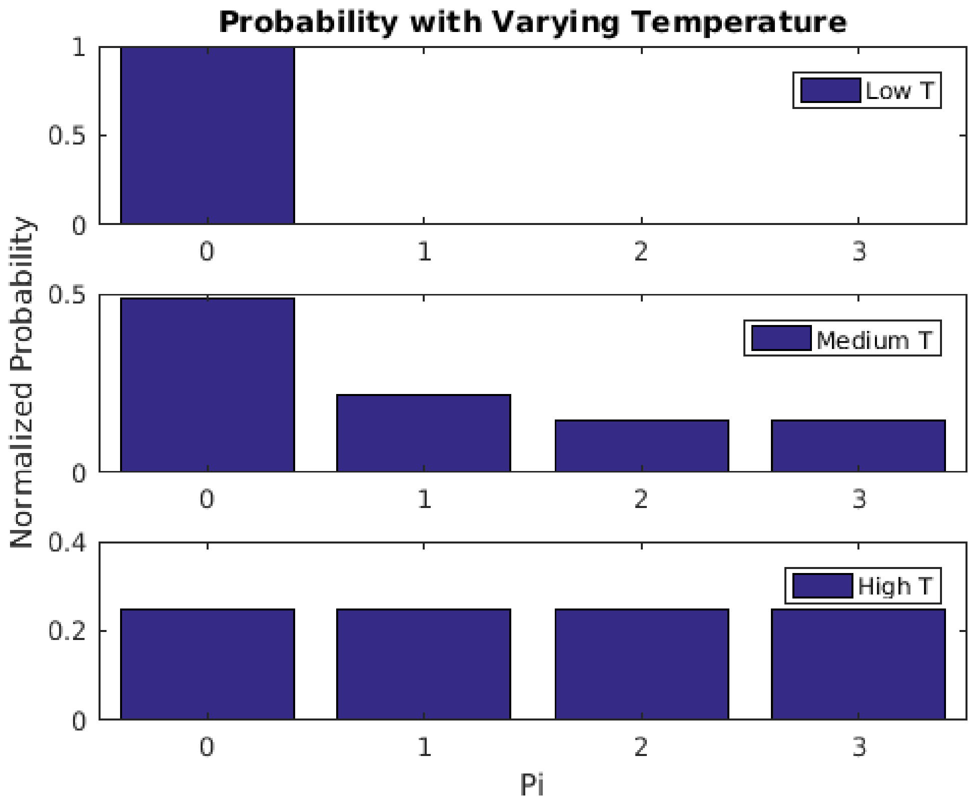
contour of I_{given} freq domain picture



2

(d)

```
beta = 1;
T = [0.1;5;1e5];
Q = 1;

alpha = exp(beta./T);

P = zeros(3,4);
P(:,1) = alpha.^(-4)./Q;
P(:,2) = alpha.^(-8)./Q;
P(:,3) = alpha.^(-10)./Q;
P(:,4) = alpha.^(-10)./Q;
P_sum = sum(P,2);
P_norm = P./repmat(P_sum,1,4);

figure;
subplot(3,1,1);
bar([0,1,2,3],P_norm(1,:));
legend('Low T')
title('Probability with Varying Temperature');

subplot(3,1,2);
bar([0,1,2,3],P_norm(2,:));
ylabel('Normalized Probability');
legend('Medium T')

subplot(3,1,3);
bar([0,1,2,3],P_norm(3,:));
xlabel('Pi');
legend('High T')
```



Published with MATLAB® R2016b