# EECS225B-Spring 2020 — PROBLEM SET 02

XIUQI DENG, SID 3035539480

## 1 PROBLEM 4.25

The dc term this the center white dot in the right image. The frequency term is the distance between two spikes.

#### 1.1

Since the wide of stripes double, the frequency of stripes became half of origin. As a result, the distance between the left spike and right spike will be half of origin. They will stay in horizontal. and the center one will stay in center.

#### 1.2

Because the stripe is vertical axis, components of the spectrum will be perpendicular to the stripe.

### 1.3

Since the wide of stripes becomes half, the frequency of stripes will double. As a result, the distance between the left spike and right spike will double as well. They will stay in horizontal. But since now they seem in the half point between edge and center spike, if we double the distance, they will go the the edge of the spectrum of the image. Matching the true that one pixel is the minimum width(as well as the maximum frequency) so the distance will be max possible position, which is the horizon edge of the image.

#### 1.4

No change.

Change in frequency will not affect the magnitude of image.

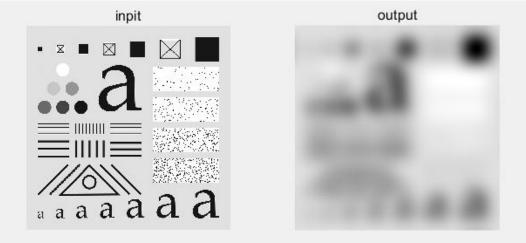
# 2 PROBLEM 4.26

From sampling theorem, we know that a function can be recovered completely from a set of its samples if the samples are acquired at a rate exceeding twice the highest frequency content of the function. The frequency of the image is steady to be 250 per line. So the resolution chip should at least be 500 per horizontal line. Since cost is an important consideration. We choose the smallest choice which is 500 per horizontal.

# 3 Project 4.6

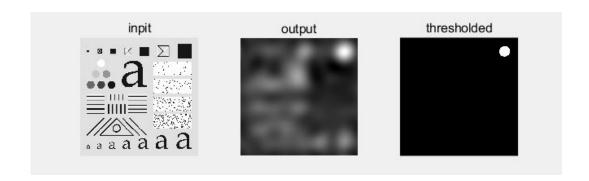
#### 3.1 a

```
project4 6 a.m × project4 6 b.m × intXform4e.m × +
        img = imread('testpattern1024.tif');
2 -
       img = double(img);
      [M, N] = size(img);
3 -
       m = (M + 1) / 2;
       n = (N + 1) / 2;
      sigma = 4;
6 —
      img_fft = fftshift(fft2(img));
      H = zeros(M, N);
9 —
      G = zeros(M, N);
10 -
     \Box for i=1:M
      for j=1:N
11 -
12 -
              d = (i-m)^2 + (j-n)^2;
               H(i, j) = \exp(-d/(2*sigma^2));
13 -
14 -
15 -
       - end
16 -
       G = H . * img_fft;
17 -
       img_result=ifft2(ifftshift(G));
18 —
      img_result=real(img_result);
        subplot(1, 2, 1); imshow(img, []); title('inpit');
19 -
        subplot(1, 2, 2); imshow(img_result, []); title('output')
20 -
21
                  inpit
                                                          output
```



#### 3.2 b

```
1 -
         img = imread('testpattern1024.tif');
 2 -
         [M, N] = size(img);
 3 -
      □ for i=1:M
 4 - 🗀
            for j=1:N
                 img.r(i, j) = 255 - img(i, j);
 5 -
 6 -
             end
 7 -
       - end
       img_fft = fftshift(fft2(img_r));
 8 -
       H = zeros(M, N);
 9 -
10 -
       D0 = 30:
       n = 2;
11 -
12 -
       m = M / 2;
13 -
       n = N / 2;
14 - ☐ for i=1:M
15 -
            for j=1:N
                d = (i-m)^2 + (j-n)^2;
16 -
                 H(i, j) = 1/(1 + (d / D0)^{(2 * n)};
17 -
18 -
             end
19 -
       end
       G = H . * img_fft;
20 -
        img_result=ifft2(ifftshift(G));
21 -
22 -
      img_result=real(img_result);
     t = max(img_result(:));
23 -
25 -
          for j = 1:N
              result(i, j) = img_result(i, j) > (t * 0.77);
26 -
27 -
          end
28 -
      - end
29
       subplot(1, 3, 1); imshow(img, []); title('inpit');
30 -
       subplot(1, 3, 2); imshow(img_result, []); title('output');
31 -
32 -
       subplot(1, 3, 3); imshow(result, []); title('thresholded');
```



### 3.3 c

```
1 -
       img = imread('checkerboard1024-shaded.tif');
 2 -
       img = double(img);
 3 -
       [M, N] = size(img);
       m = (M + 1) / 2;
 4 -
       n = (N + 1) / 2;
 5 -
      sigma = 4;
       img_fft = fftshift(fft2(img));
 8 -
      H = zeros(M, N);
      G = zeros(M, N);
10 -
     □ for i=1:M
     for j=1:N
11 —
              d = (i-m)^2 + (j-n)^2;
12 -
               H(i, j) = \exp(-d/(2*sigma^2));
13 -
14 -
            end
15 -
       end end
16 -
       G = img_fft . * H;
       G=ifft2(ifftshift(G));
17 -
18 -
      G=rea1(G);
       img_result = img . / G;
19 -
20 -
      subplot(1, 2, 1); imshow(img, []); title('inpit');
        subplot(1, 2, 2); imshow(img_result, []); title('output');
21 -
                                                         output
                    inpit
```

# 4 Project 4.7

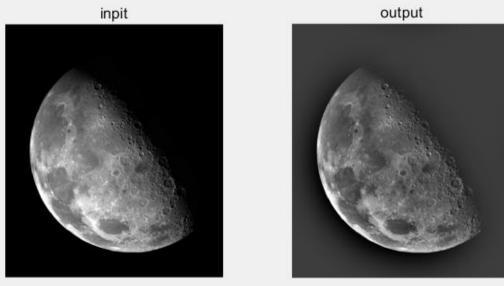
#### 4.1 a

```
img = double(img);
 2 -
        [M, N] = size(img);
 3 -
        m = (M + 1) / 2;
        n = (N + 1) / 2;
        sigma = 2;
        img_fft = fftshift(fft2(img));
        H = zeros(M, N);
 8 -
        G = zeros(M, N);
 9 -
      □ for i=1:M
10 -
11 -
           for j=1:N
               d = (i-m)^2 + (j-n)^2;
12 -
13 -
               H(i,j) = \exp(-d/(2*sigma^2));
14 -
            end
15 —
        end
        G = H . * img_fft;
16 -
17 -
        img_blur=ifft2(ifftshift(G));
18 -
        img_blur=real(img_blur);
19 -
        mask = img - img_blur;
20 -
        img_result = img + mask;
        subplot(1, 2, 1); imshow(img, []); title('inpit');
21 -
        subplot(1, 2, 2); imshow(img_result, []); title('output');
22 -
```



### 4.2 b

```
project4_7_a.m project4_7_b.m + 1 mg = imread( blurry-moon. til );
         img = double(img);
 3 -
         [M, N] = size(img);
        m = (M + 1) / 2;
 4 -
        n = (N + 1) / 2;
        sigma = 2;
 7 -
        img_fft = fftshift(fft2(img));
 8 -
        H = zeros(M, N);
 9 -
        G = zeros(M, N);
10 -
      □ for i=1:M
      for j=1:N
11 -
                d = (i-m)^2 + (j-n)^2;
12 -
                H(i,j) = \exp(-d/(2*sigma^2));
13 -
14 -
             end
15 -
       - end
16 -
        G = H . * img_fft;
17 -
         img_blur=ifft2(ifftshift(G));
18 -
         img_blur=real(img_blur);
19 -
         mask = img - img_blur;
         img_result = img + 5 * mask;
20 -
21 -
        subplot(1, 2, 1); imshow(img, []); title('inpit');
         subplot(1, 2, 2); imshow(img_result, []); title('output');
22 -
                 inpit
```



# 5 Project 4.8

#### 5.1 a

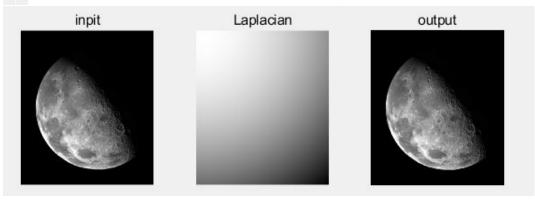
```
project4 8 a.m 💥 🛨
         img = imread('Fig0457(a) (thumb_print).tif');
         subplot(1, 3, 1); imshow(img, []); title('inpit');
 2 -
 3
 4 -
      = for i = 1:M
      -
 5 -
            for j = 1:N
                  img(i, j) = 255 - img(i, j);
 6 -
             end
       - end
 8 -
 9 -
        img = double(img);
       [M, N] = size(img);
10 -
11 -
      m = (M + 1) / 2;
      n = (N + 1) / 2;
12 -
      sigma = 10;
13 -
      img_fft = fftshift(fft2(img));
14 -
15 -
      H = zeros(M, N);
       G = zeros(M, N);
     \equiv for i=1:M
17 -
18 -
     for j=1:N
               d = (i-m)^2 + (j-n)^2:
19 -
                H(i, j) = d > sigma;
20 -
21 -
             end
00
      G = H .* img_fft;
23 —
      img_r=real(ifft2(ifftshift(G)));
25 —
    subplot(1, 3, 2); imshow(img_r, []); title('result of filtering');
26 —
    t = max(img_r(:));
27 - ☐ for i=1:M
28 - for j=1:N
29 —
            img_result(i, j) = img_r(i, j) > 10;
30 —
          end
31 —
      subplot(1, 3, 3); imshow(img_result, []); title('output');
          inpit
                                result of filtering
                                                                output
```

### **5.2** b

```
| function H = laplacianTF4e(P, Q)
| H = zeros(P, Q);
| for i = 1:P
| for j = 1:Q
| H(i, j) = -4 *(pi^2) * (i^2 + j^2);
| end |
| end |
| Laplacian
```

### 5.3 c

```
1 —
        img = imread('blurry-moon.tif');
      [M, N] = size(img);
2 -
3 —
        H = 1aplacianTF4e(M, N);
4 —
       img_fft = fftshift(fft2(double(img)));
        G = H . * img_fft;
6 —
        img_la = real(ifft2(ifftshift(G)));
7 —
      img_la = img_la / max(img_la(:));
        img_result = double(img) - img_la;
8 -
9
10 —
        subplot(1, 3, 1); imshow(img, []); title('inpit');
11 —
        subplot(1, 3, 2);imshow(H, []);title('Laplacian');
        subplot(1, 3, 3); imshow(img_result, []); title('output');
12 —
```



#### 5.4 d

```
+1 | laplacianTF4e.m | x | project4_8_c.m | x | project4_8_d.m | x | +
       img = imread('Fig0459(a)(orig_chest_xray).tif');
 2 -
       subplot(2, 2, 1); imshow(img, []); title('a');
 3
 4 -
       img = double(img);
       [M, N] = size(img);
 5 —
      m = (M + 1) / 2;
 6 -
 7 -
       n = (N + 1) / 2;
 8 -
       d0 = 170;
 9 -
       img_fft = fftshift(fft2(img));
      H = zeros(M, N);
10 -
11 -
      G = zeros(M, N);
12 - ☐ for i=1:M
13 - ☐ for j=1:N
14 -
             d = (i-m)^2 + (j-n)^2;
             H(i, j) = 1 - \exp(-d/(2*(sigma^2)));
15 -
16 -
           end
      end
17 -
18 -
      K = img_fft .* H;
19 -
       G = img fft .* (0.5 + 0.75 * H):
       img_rl=real(ifft2(ifftshift(G)));
20 -
       img_r2=real(ifft2(ifftshift(K)));
21 -
       subplot(2, 2, 2); imshow(img_r2, []); title('b');
22 -
        subplot(2, 2, 3); imshow(img_r1, []); title('c');
23 -
24 -
       img rl = uint8(img rl);
25 -
       h(1:256) = 0:
     \Box for I = 1:256
26 -
27 -
             idx = find(img_rl == I - 1);
28 -
             h(I) = numel(idx);
29 -
      - end
       h = h / (M * N):
30 -
31 -
       s = zeros(1, 256);
32 -
      \Box for i = 1:256
33 -
             for j = 1:i
                  s(i) = s(i) + h(j);
34 -
35 -
              end
36 -
      - end
       s2 = round(s * 256);
37 -
38 -
       p = zeros(1, 256);
```

```
37 —
      s2 = round(s * 256);
38 -
      p = zeros(1, 256);
39 - \Box \text{ for i} = 1:256
40 -
            p(i) = sum(h(s2 == i));
41 -
       - end
42 -
       img_result = img_r1;
43 —
      \Box for i = 0:255
44 —
            img_result(img_rl == i) = s2(i + 1);
45 -
46
      subplot(2, 2, 4); imshow(img_result, []); title('output');
47 -
                                                               b
                                                            output
                   C
```