HW2

Houlin Li

June 28, 2023

Exercise 1

1. 2-D continuous Fourier transform.

   We know the 2-D continuous Fourier transform can be shown below:

$$
\begin{aligned}
F(\mu, \nu) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t,z) e^{-2j\pi(\mu t + z\nu)} dt dz \\
&= \int_{-\infty}^{\infty} e^{-2j\pi \nu z} dz [\int_{-\infty}^{\infty} f(t,z) e^{-2j\pi\mu t} dt] \\
&= \int_{-\infty}^{\infty} F(\mu, z) e^{-2j\pi\nu z} dz
\end{aligned} \tag{1.1}
$$

   Because the $F(\mu, z)$ is a 1-D Fourier transform, which is linear. Therefore, we have:

$$
\begin{aligned}
F[af_i(t,z) + bf_J(t,z)] &= \int_{-\infty}^{\infty} (aF_i(\mu, z) + bF_j(\mu, z)) e^{-2j\pi\nu z} dz \\
&= a \int_{-\infty}^{\infty} F_i(\mu, z) e^{-2j\pi\nu z} dz + b \int_{-\infty}^{\infty} F_j(\mu, z) e^{-2j\pi\nu z} dz \\
&= aF_i(\mu, \nu) + bF_j(\mu, \nu)
\end{aligned} \tag{1.2}
$$

   And $F_i(\mu, \nu)$ means $\mathscr{F}[f_i(t,z)]$ So, 2-D continuous Fourier transform is linear.

2. 2-D discrete Fourier transform.

$$
\begin{aligned}
F(u, v) &= \sum_{x=0}^{M-1} \sum_{y=0} N - 1 f(x,y) e^{-2j\pi(ux/M + vy/N)} \\
&= \sum_{x=0}^{M-1} F(x, u) e^{-2j\pi ux/M}
\end{aligned} \tag{1.3}
$$

   Similar as continuous Fourier transform:

$$
F[af_i(x,y) + bf_J(x,y)] = aF_i[(x,y)] + bF_j[(x,y)] \tag{1.4}
$$

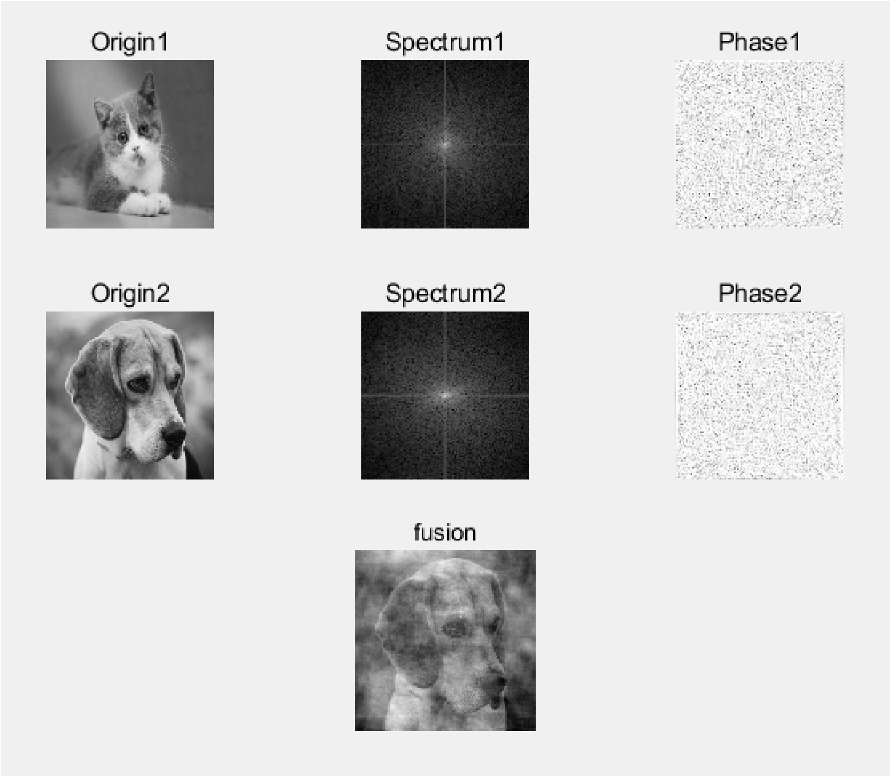
   So the 2-D discrete Fourier function is linear.

Exercise 2

the code is below:

```
Image1 = imread('D:\1.jpg');
Image2 = imread('D:\2.jfif');
if size(Image1,3) > 1
    G1 = im2double(rgb2gray(Image1));
else
    G1 = im2double(Image1);
end
if size(Image2,3) > 1
    G2 = im2double(rgb2gray(Image2));
else
    G2 = im2double(Image2);
end
F1 = fft2(G1);
F2 = fft2(G2);
imS1 = log10(abs(fftshift(F1))+1);
imP1 = log(angle(F1)*180/pi);
imS2 = log10(abs(fftshift(F2))+1);
imP2 = log(angle(F2)*180/pi);
subplot(3,3,1), imshow(G1), title('Origin1');
subplot(3,3,2), imshow(imS1,[]), title('Spectrum1');
subplot(3,3,3), imshow(imP1,[]), title('Phase1');
subplot(3,3,4), imshow(G2), title('Origin2');
subplot(3,3,5), imshow(imS2,[]), title('Spectrum2');
subplot(3,3,6), imshow(imP2,[]), title('Phase2');
%the new image is fused by the spectrum of 1 and phase angle of 2
F3 = abs(F1).*cos(angle(F2))+abs.*sin(angle(F2)).* 1j;
imF3 = ifft(F3);
subplot(3,3,[7,8,9]), imshow(imF3), title('fusion')
```

So run the codes above, we can see the result:



Exercise 3

We know the complement of RGB colour is : R' = 1-R, G' = 1-G, B' = 1-B

So the saturation component of the of a color image is :

$$\begin{aligned} S' &= 1 - \frac{3}{R' + G' + B'} min\{R', G', B'\} \\ &= 1 - \frac{3}{3 - (R + G + B)}(1 - max\{R, G, B\}) \end{aligned} \tag{3.1}$$

And the saturation component of the input image is :

$$S = 1 - \frac{3}{R + G + B} min\{R, G, B\} \tag{3.2}$$

Apparently, we can't know the value of max{ R,G,B}. So there is no a funtion that can calculate s' from s.

Let's give an example: assume there is an input image whose RGB is (0,0.7,1) , and another input image's RGB is (0.5,0,0.5). We can easily know the values below:

- the saturation is $s_1 = 1$, and $s_2 = 1$.
- the complement of the input images' RGB are (1,0.3,0) and (0.5,1,0.5)
- the saturation of complement images are $s'_1 = 1$ and $s'_2 = 0.25$

So, although the saturation component of the input images are the same, the saturation component of their complement images are different.

Exercise 4

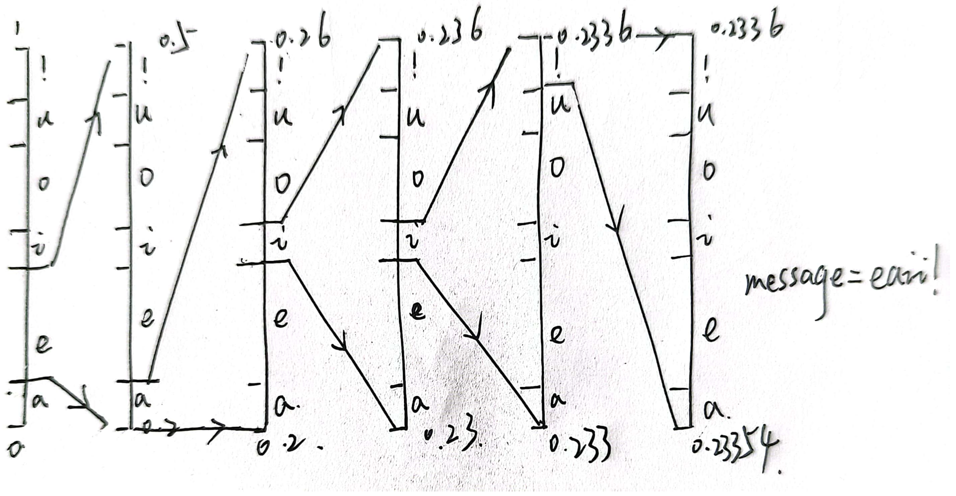


Figure 1: Decode

As shown in the 1 above. At the start of the coding process, the message is assumed to occupy the entire half-open interval [0, 1). We noticed that the message 0.23355 is within [0.2, 0.5). So the first symbol a1 is ‘e’.

Second, after we’ve determined a1, it narrows the sub-interval to [0.2, 0.5). Similarly, we can get a2 should be ‘a’.

In the same way, we can know a3, a4, a5 should be ’i’, ’i’, ’!’.

So we can get ‘eaii!’. Now the range is narrowed to [0.23354, 0.23360). It has the same number of decimal digits with the encoded message 0.23355. Besides, the symbol ‘!’ is usually used as a end-of message indicator.

Therefore, it is possible that the origin message has 5 symbols, which is ‘eaii!’

Exercise 5

Because Huffman code has no ambiguity. And the code column in the table is the corresponding code of each character. So we can easily know string is $a_3a_6a_6a_2a_5a_2a_2a_2a_4$.

Exercise 6

From the definition of blurring function $H(u, v)$, we know:

$$H(u, v) = \int_0^T e^{-j2\pi[ux(t)+vy(t)]}dt \tag{6.1}$$

Because the coordinates are :

$$x(t) = \frac{at^2}{2}, \quad y(t) = 0 \tag{6.2}$$

So, substitute the above formula into 6.1, we have:

$$H(u, v) = \int_0^T e^{-j\pi uat^2} dt \tag{6.3}$$

Exercise 7

From the definition of random operator, we know that:

$$g(\rho, \theta) = RT[f(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(x \cos\theta + y \sin\theta - \rho)dxdy \tag{7.1}$$

So when the input is $a_1 f_1(x, y) + a_2 f_2(x, y)$, we have:

$$\begin{gathered} RT[a_1 f_1(x, y) + a_2 f_2(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (a_1 f_1(x, y) + a_2 f_2(x, y))\delta(x \cos\theta + y \sin\theta - \rho)dxdy \\ = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_1 f_1(x, y)\delta(x \cos\theta + y \sin\theta - \rho)dxdy + \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} a_2 f_2(x, y)\big)\delta(x \cos\theta + y \sin\theta - \rho)dxdy \\ = a_1 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_1(x, y)\delta(x \cos\theta + y \sin\theta - \rho)dxdy + a_2 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_2(x, y)\big)\delta(x \cos\theta + y \sin\theta - \rho)dxdy \\ = a_1 RT[f_1(x, y)] + a_2 RT[f(_2(x, y)] \end{gathered} \tag{7.2}$$

So the random operation is a linear operation.