

題目敘述

離散傅立葉轉換 DFT 練習

撰寫傅利葉轉換程式(Forward Fourier Transform and Inverse Fourier Transform)將一張圖

像轉換至頻域後，將頻譜大小與相位角度各以灰階 256 色圖像方式呈現出，再呈現還

原後圖像。

開發環境

Windows10、Spyder (Python 3.8)、OpenCV 4.7.0

說明

1、讀取圖像並轉為灰階

2、用 `getOptimalDFTSize` 得到最適合進行傅里葉變換的圖像大小，然後用 `copyMakeBorder` 進行 padding，並轉成 `float32` 來計算

2、 $\text{fp}(x,y) \times (-1)^{(x+y)}$ to center the Fourier transform

```
for i in range(img.shape[0]):  
    for j in range(img.shape[1]):  
        padded_img[i][j] *= (-1)**(i+j)
```

3、`dft()`計算傅立葉轉換，然後用 `split()`將虛數與實數分開

`magnitude()`計算幅值，並取 `log` 乘上 20 得到振幅

`normalize()` 標準化，讓圖像以灰階 256 色圖像方式呈現，並轉成 `uint8` 型態(避免負數)

```
S = 20 * np.log(cv2.magnitude(planes[0], planes[1]))  
# normalize the output image to [0, 255]  
S = cv2.normalize(S, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```

`phase()` 計算相位

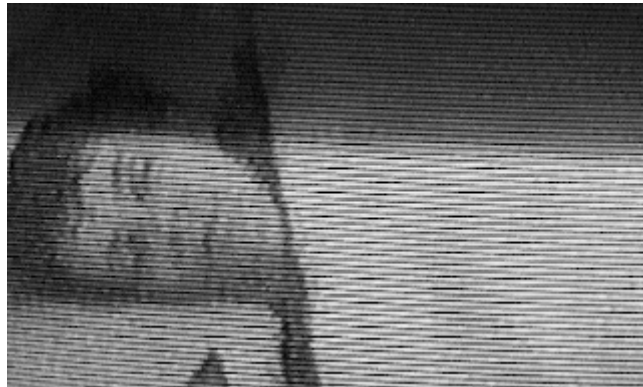
並同上進行標準化。

```
phase = cv2.phase(planes[0], planes[1], angleInDegrees=True)  
phase = cv2.normalize(phase, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```

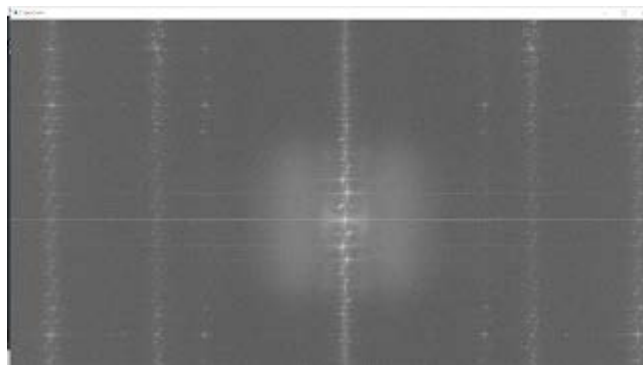
`idft()` 計算還原後圖像

magnitude()計算幅値
並同上進行標準化。

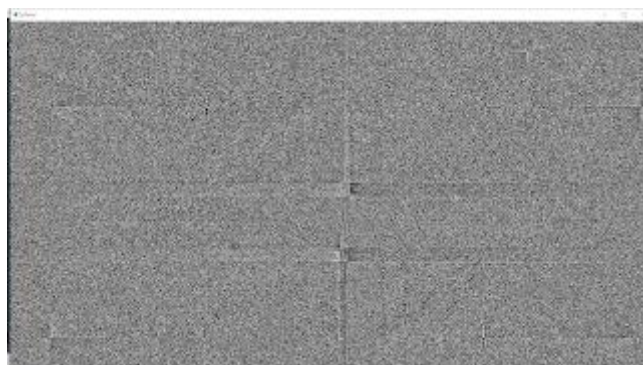
```
idft = cv2.idft(DFT)  
idft = cv2.magnitude(idft[:, :, 0], idft[:, :, 1])  
idft = cv2.normalize(idft, None, 0, 255, cv2.NORM_MINMAX).astype(np.uint8)
```



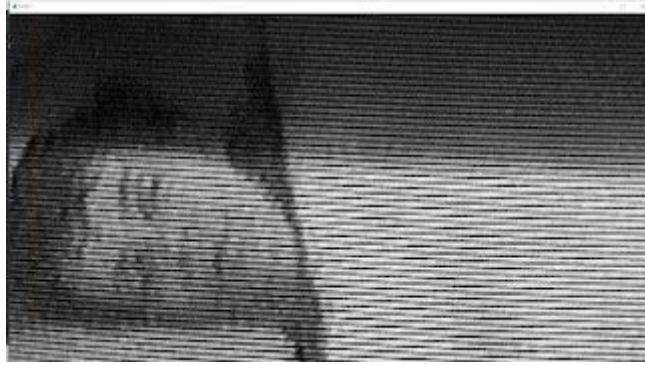
Original



Spectrum



Phase



IDFT