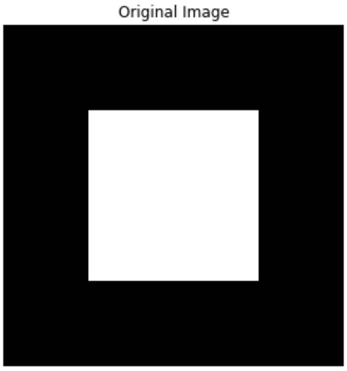
數位影像處理 HW4

電機系控制組碩一 R13921109 陳柏丞

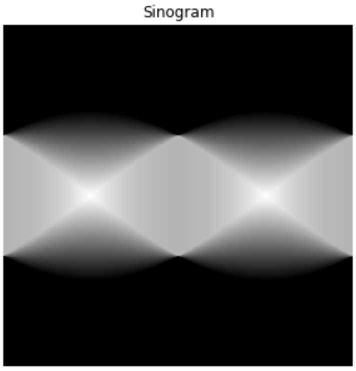
Problem 1:

The purpose of this homework is to exercise how an image can be reconstructed using filtered backprojections and a ramp limited by a Hamming window as described in Example 5.17 of the textbook for an image of 600x600 pixels with black background and a central white square of 300x300 pixels. Generate your result using 0.5° increments of rotation and 849 projection rays as described in the textbook.

1. 建立一個 600x600 pixels 的黑底圖像，中央為 300x300 的白色方塊

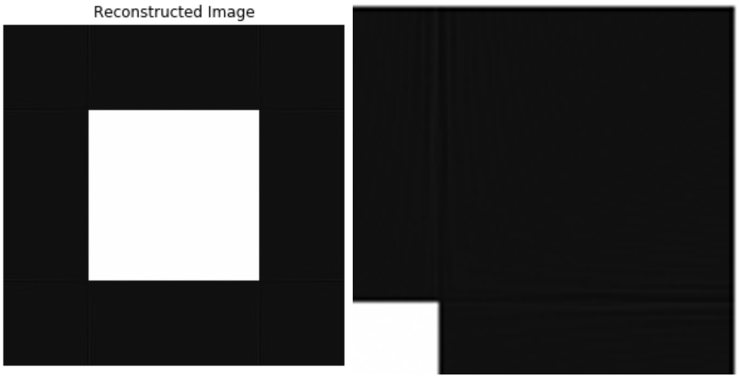


2. 使用 radon 函數對影像進行 Radon 轉換，以生成不同角度(0.5˚旋轉量和 849 條投影光線產生結果)的 sinogram 圖像

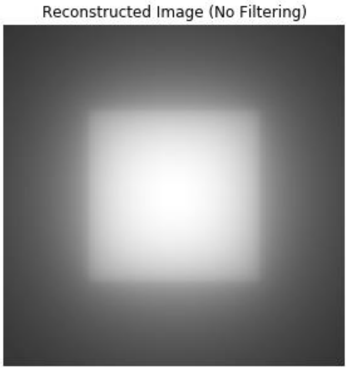


(849x600)

3. 利用 iradon 函數對濾波後的投影資料進行回放，生成重建影像



3-1. 未進行濾波直接進行反投影的結果



4. 結果分析:

i. 邊緣呈現直橫條紋: 由於重建過程中的有限角度投影或濾波過程產生的，雖然 849 條投影數量相對較多，但若角度分布過於間隔或沒有充分涵蓋所有角度，這可能會導致邊緣條紋，若增加投影角度的密度如(0.25˚或 0.125˚)

ii. 在進行反投影（iradon）時，scipy 中的 iradon 函數預設會根據投影數量來決定重建影像的大小，通常會將影像 padding 到一個較大的矩陣來避免邊界效應，這樣重建影像的大小可能會比原影像稍大，從而避免邊緣資訊的丟失。scipy 函數會根據投影的數量擴展輸出影像矩陣，可能會導致最終輸出影像為 849x849，最後在擷取中心 600x600 作為輸出結果。

Problem2:

JPEG uses the Huffman code defined in Table K.5 of the CCITT T.81 standard document to encode AC coefficients. A copy of the document is provided on NTU COOL for your information. Given the following quantized image block in the DCT domain:

-10	-3	-6	2	-1	0	0	0
0	-2	2	0	1	0	0	0
4	3	5	-1	0	0	0	0
0	0	0	0	0	0	0	0
1	0	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

a) Determine the code stream for the AC coefficients of the image block.

先取得Zig-Zag數列

Zig-Zag Sequence: -10,-3,0,4,-2,-6,2,2,3,0……

計算AC code stream，三個關鍵數字1.zero-run 2.Size 3.nonzero cofficient

根據Run/Size的值去查詢table，得到前面的code word，再加上二進制的值

AC coefficients:

First nonzero cofficient = -3; zero-run= 0, (0,2)(-3) = **0100**

2nd nonzero cofficient = 4; zero-run= 1, (1,3)(4) = **1111001100**

3rd nonzero cofficient = -2; zero-run= 0, (0,2)(-2) = **0101**

4th nonzero cofficient = -6; zero-run= 0, (0,3)(-6) = **100001**

5th nonzero cofficient = 2; zero-run= 0, (0,2)(2) = **0110**

6th nonzero cofficient = 2; zero-run= 0, (0,2)(2) = **0110**

7th nonzero cofficient = 3; zero-run= 0, (0,2)(3) = **0111**

8th nonzero cofficient = 1; zero-run= 1, (1,1)(1) = **11001**

9th nonzero cofficient = 5; zero-run= 1, (1,3)(5) = **1111001101**

10th nonzero cofficient = -1; zero-run= 1, (1,1)(-1) = **11000**

11th nonzero cofficient = 1; zero-run= 1, (1,1)(1) = **11001**

12th nonzero cofficient = -1; zero-run= 0, (0,1)(-1) = **000**

13th nonzero cofficient = -1; zero-run= 2, (2,1)(-1) = **111000**

14th nonzero cofficient = -1; zero-run= 2, (2,1)(-1) = **111000**

END **1010**

b) Suppose the DC component of the previous image block is -5. Determine the code stream for the DC coefficient of the image block. (10%)

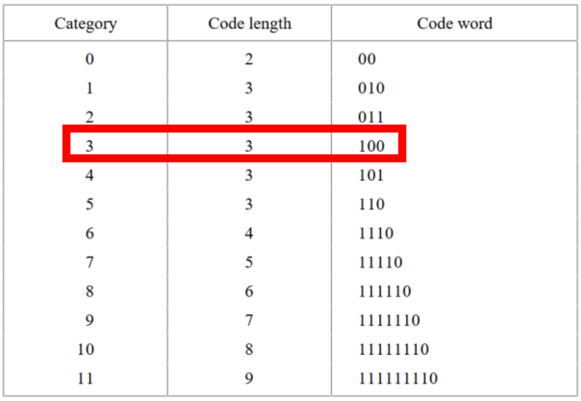
DC coefficient:

i. Assuming previous DC cofficient = -5

ii. -10 - (-5) = -5

iii. Diff SIZE =3, AMPLITUDE=-5

Table K.3 – Table for luminance DC coefficient differences



Category	Code length	Code word
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

-5 -> 010(1 補數)

DC code stream: **100010**