Homework 3

3.6 (a) If set to zero the lower-order bit planes small values should be zeros on histogram. Large values should be regularly discrete.

(b) If set to zero the higher-order bit planes some large values should be zeros and some small values increase

3.34 (a) The two results are not equal. Because obviously the result of first one has some white or blacks. But the second one are all between white and black.

(b)

Value	amount
0	24
0.2222	2
0.3333	6
0.4444	4
0.6667	16
1	12

Result of figure left

Value	amount
0.2222	4
0.3333	24
0.4444	18
0.5556	18

Result of figure right

3.36 (a) The result is still Gaussian. Because the convolution of Gaussian is still a Gaussian.

(b) $\sigma^2 = 1.5^2 + 2^2 + 4^2 = 22.25$

$\sigma = \sqrt{\sigma^2} = 4.717$

(c) ~~If using zero padding, the result is 7×7~~

$(3+5-1)+7-1 = 13$ the result is 13×13

3.48 Yes. Because they are both convolution. It can be commutative.

3.35 (a) The Laplacian kernels are not separable.

(b) The Roberts cross-gradient kernels are not separable.

(c) The Sobel kernels are separable.

for (d) $vw^T = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}^T$

for (e) $vw^T = \begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^T$

Function explanation:

Function myLaplacian() receives a grayscale image stored in a matrix and return a sharped image. Its main method is to create a matrix of every location in the kernel and make it match the location of result image. For example, the top left location (1,1) is 0 when calculate (1,2) in result image, so its matrix (1,2) is 0. Obviously, (2,2)'s matrix is image itself.

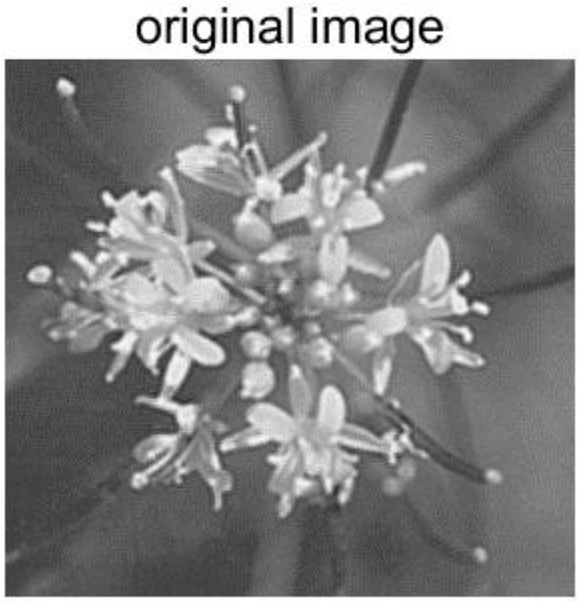
Then we just multiply corresponding paraments and add them together to get Laplacian result. And add it to the original image to get final result.

Function myUnsharp() receives a grayscale image stored in a matrix and return a sharped image. Firstly, I do a zero padding according to the filter size. Then I use a double loop to calculate the value of each pixel in mean filter result. The original image minus mean result to get the mask. Mask is multiplied by k and added to the original image to get final result.

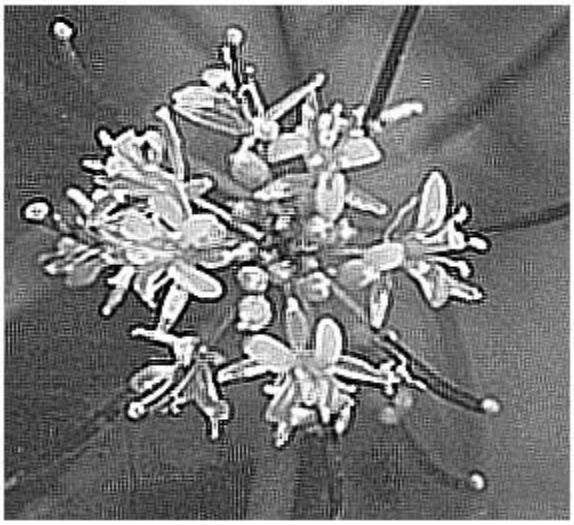
Results:

Both Laplacian and Unsharp perform well when sharping a blurry image. Results are obviously more clear and the edges are highlighted.

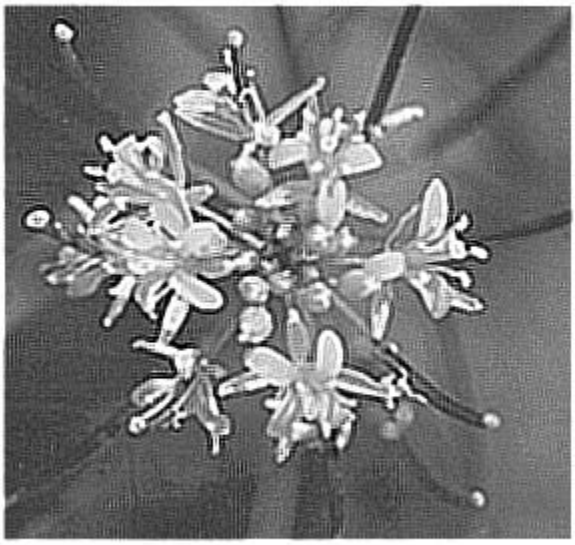
Flower.pgm



Result of Laplacian



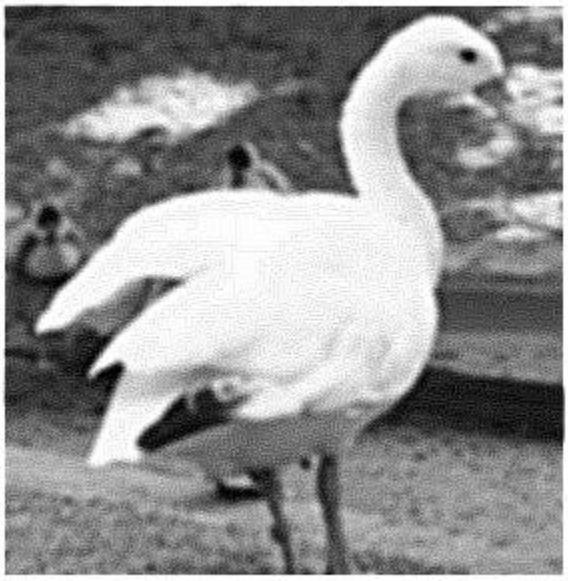
Result of Unsharp k = 5



Swan.pgm:

original image

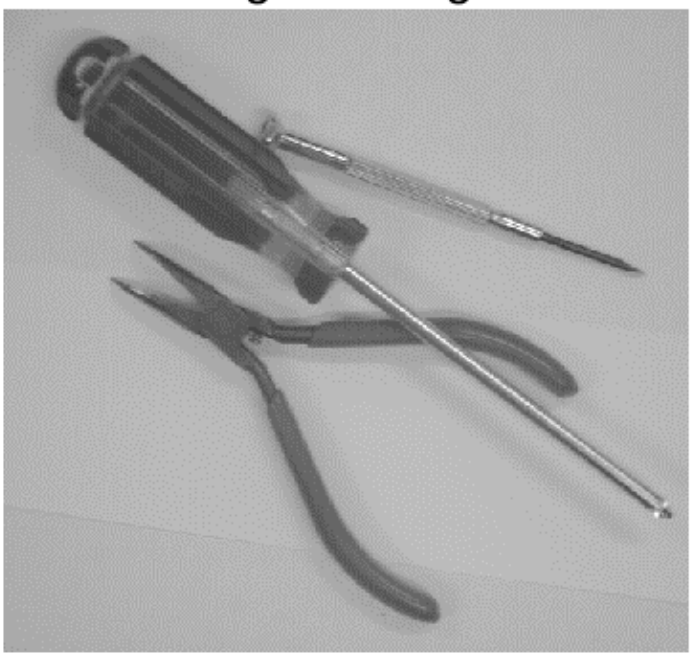
Result of Laplacian



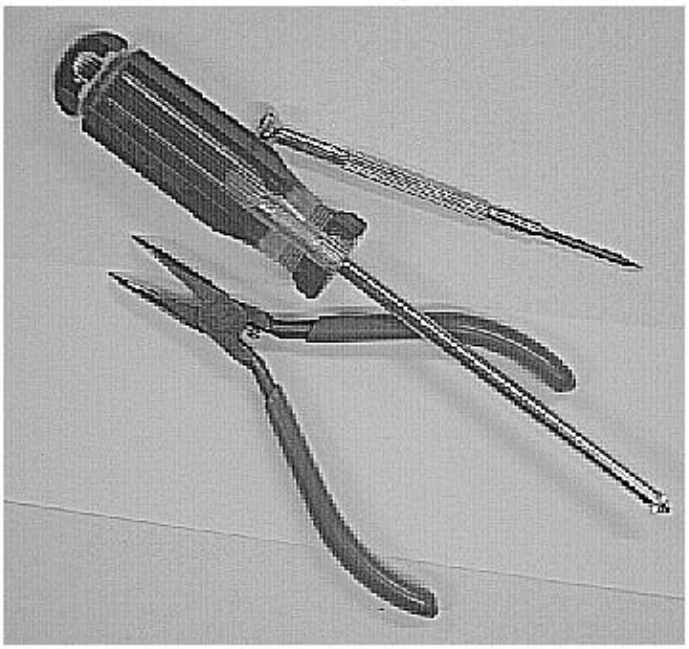
Result of Unsharp k = 5

Tools.pgm:

original image



Result of Laplacian



Result of Unsharp k = 5