

Chapter 1. Average Filter and Median Filter

1. Test the given pgm-reader program use 3 x 3 filter:

- Average Filter

Algorithm:

Input: an unsigned char array to store source image data.

Output: an unsigned char array to store output image data.

(i-1, j-1)	(i-1, j)	(i-1, j+1)
(i, j-1)	(i, j)	(i, j+1)
(i+1, j-1)	(i+1, j)	(i+1, j+1)

Each pixel in the picture is surrounded by **eight** pixels except for those at the edges which are ignored. And we recalculate the value of each pixel by taking the average of the **nine** pixels in figure. So we have the following algorithm:

$$Pixel(i, j) = data[i * Width + j], \quad i, j \geq 1, i < Height - 1, \text{ and } j < Width - 1.$$

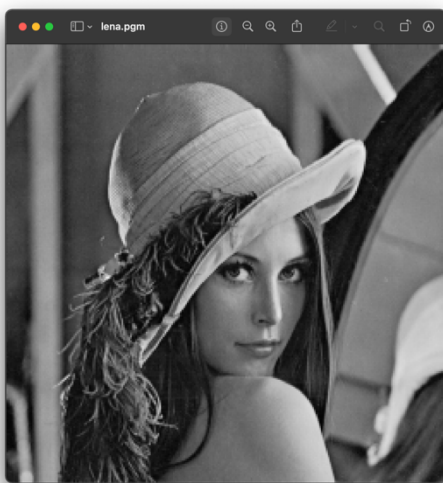
newPixel

$$(i, j) = \sum_{x=-1}^1 \sum_{y=-1}^1 originalPixel(i + x, j + y) / 9 = \sum_{x=-1}^1 \sum_{y=-1}^1 data[(i + x) * Width + (j + y)] / 9$$

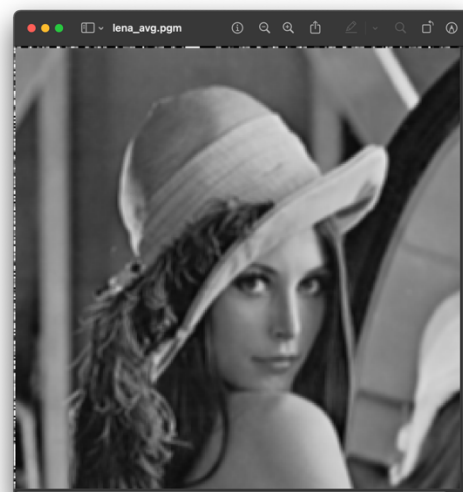
Results (including pictures):

Process result of "lena.pgm":

Source Image:



Result after average filter:

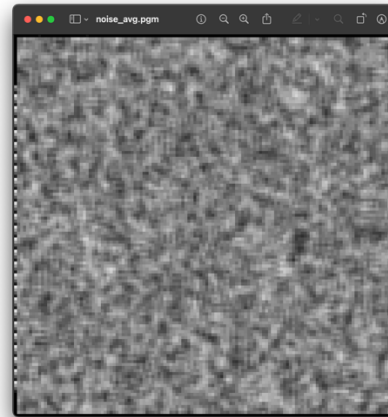


Process result of “noise.pgm”

Source Image:



Result after average filter:



Discussion:

1. We know each pixel's value is replaced by the average value of the surrounding 8 pixels and itself, so the differences between all pixels are reduced and the image will look smoother.
2. However, it does not protect the image details well, and it also destroys the details of the image while denoising the image, so that the image becomes blurred.

● Median Filter

Algorithm:

Input: an unsigned char array to store source image data.

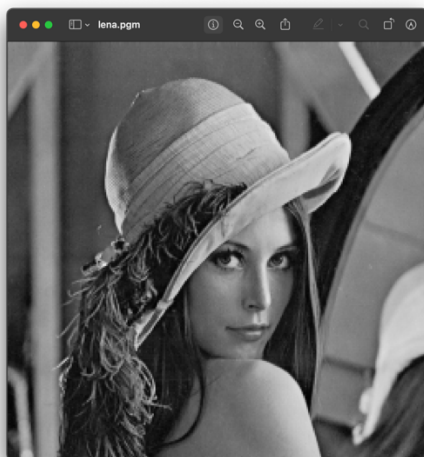
Output: an unsigned char array to store output image data.

Similar to the above question, but the value of each pixel is replaced by the **median** of the nine-square grid pixels instead of the average. I store the values of 9 surrounded pixels into an **array** and use the **Insertion Sort** method to find its median, `array[4]`, which will be assigned to `Pixel(i, j)`.

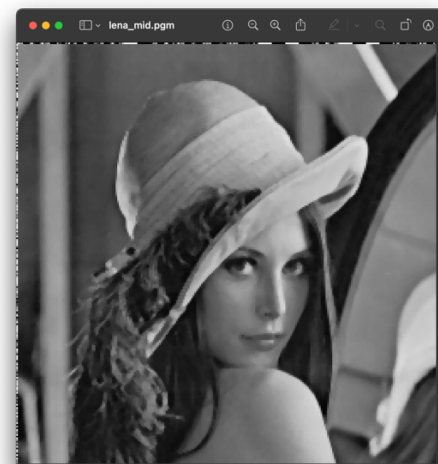
Results (including pictures):

Process result of “lena.pgm”:

Source Image:



Result after median filter:

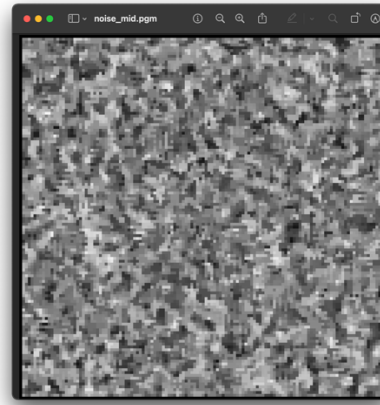


Process result of “noise.pgm”:

Source Image:



Result after median filter:



Discussion:

1. Each pixel's value is replaced by the median of the surrounding 8 pixels and itself, and the differences between all pixels are also reduced and make the image smoother.
2. Compared to the average filter, the median filter has sharper edges of objects after noise reduction. And the difference between pixels is bigger because it only processes the target pixel each time.