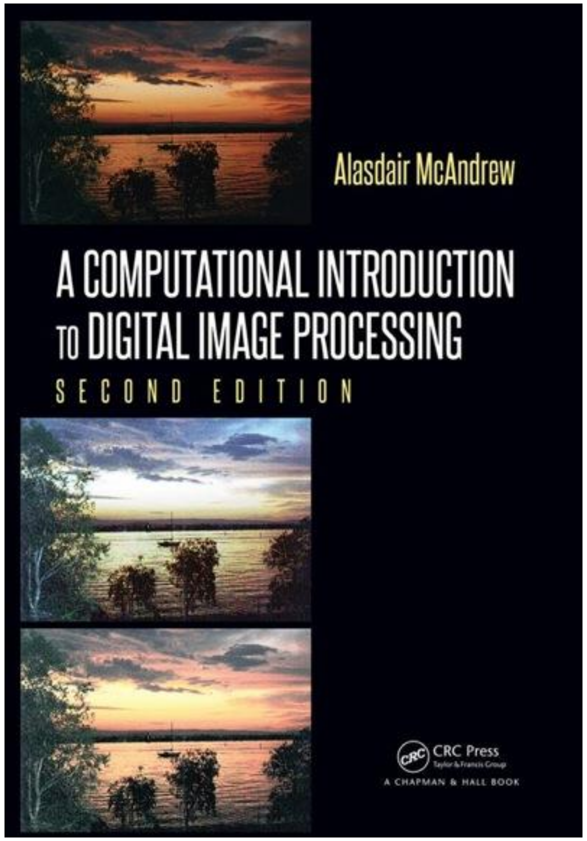
Chapter 8

Image Restoration



Contents

MIDWESTERN STATE UNIVERSITY

Introduction

- Concerns removal or reduction of degradations that occurred during image acquisition
  - Noise: errors in pixel values
  - Optical effects: out of focus blurring
  - Blurring due to camera motion
- Image restoration is one of the most important areas of image processing
- Focus: techniques for dealing with restoration

Noise

MIDWESTERN STATE UNIVERSITY

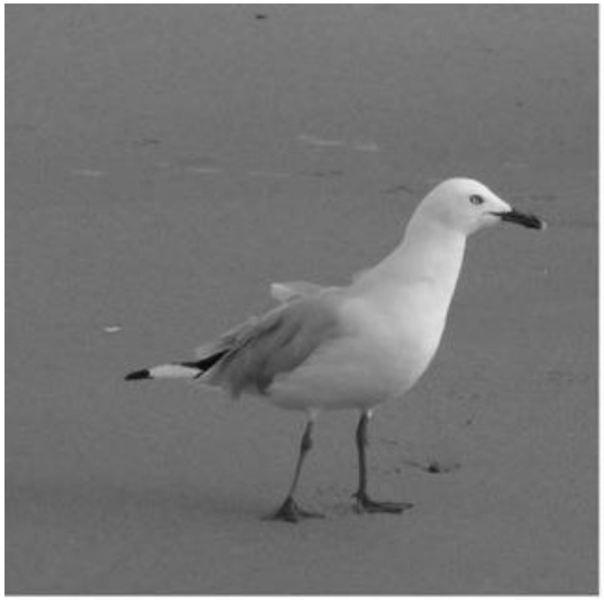
- Definition: any degradation in the image signal, caused by external disturbance
- Cleaning an image corrupted by noise is an important area of image restoration
- Look at 4 different noise types:
  - Salt and Pepper Noise
  - Gaussian Noise
  - Speckle Noise
  - Periodic Noise

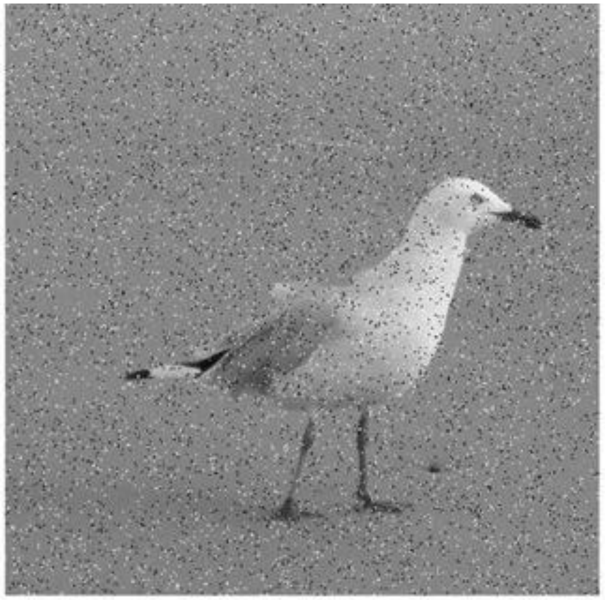
Clean using spatial filtering techniques, local degradations

Clean using frequency domain filtering, global effect

Salt & Pepper Noise

- A.k.a. – impulse noise, shot noise, binary noise
- Cause
  - sharp, sudden disturbances in the image signal
- Appearance
  - Randomly scattered white or black (or both) pixels over the image


(a) Original image


(b) With added salt and pepper noise

Salt & Pepper Noise

MIDWESTERN STATE UNIVERSITY

To add salt and pepper noise in MATLAB or Octave, we use the MATLAB function imnoise, which takes a number of different parameters. To add salt and pepper noise:

```
>> gsp = imnoise(g,'salt_and_pepper')
```

MATLAB/Octave

The amount of noise added defaults to 10%; to add more or less noise we include an optional parameter, being a value between 0 and 1 indicating the fraction of pixels to be corrupted. Thus, for example

```
>> imnoise(g,'salt_and_pepper',0.2);
```

MATLAB/Octave

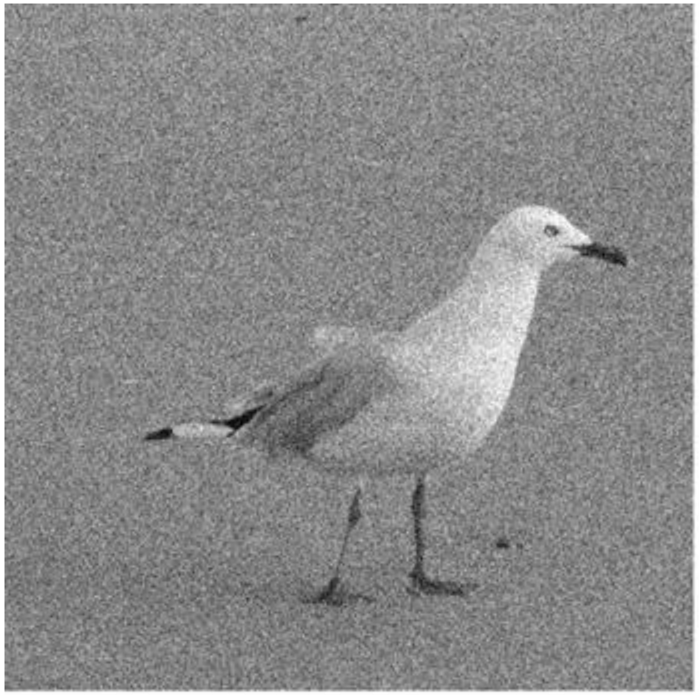
In Python, the method `noise.random_noise` from the `util` module of `skimage` provides this functionality:

```
In :  import skimage.util.noise as noise
In :  gn = noise.random_noise(g,mode='s&p')
In :  gn2 = noise.random_noise(g,mode='s&p',amount=0.2)
```

Python

Gaussian Noise

- Definition: idealized form of *white noise* that is normally distributed
- Can be modeled by random values added to an image
- Cause:
  - Random fluctuations in the signal
- Example:
  - Television slightly mistuned to a particular channel


(a) Gaussian noise

Gaussian Noise

```
>> gg = imnoise(g,'gaussian');
```

MATLAB/Octave

or

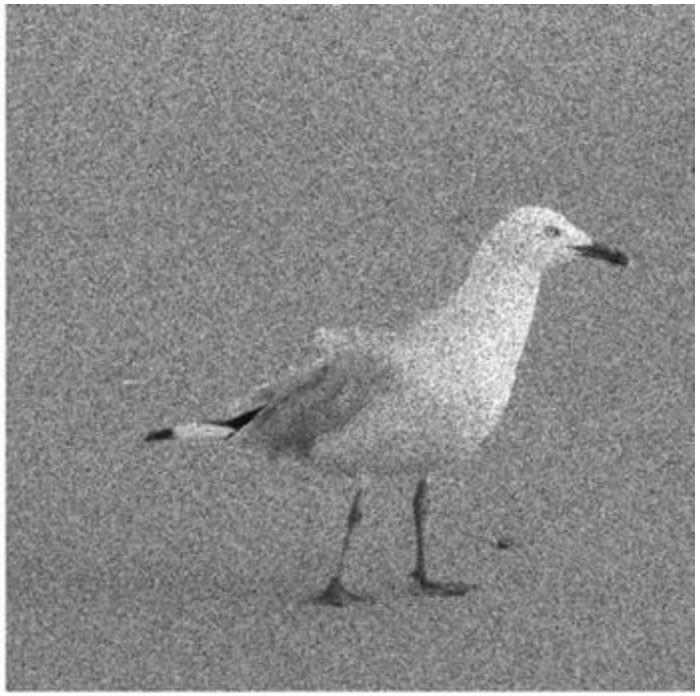
```
In :  gg = noise.random_noise(g,'gaussian')
```

Python

As with salt and pepper noise, the "`gaussian`" parameter also can take optional values, giving the mean and variance of the noise. The default values are 0 and 0.01, and the result is shown in Figure 8.2(a).

Speckle Noise

- A.k.a. – multiplicative noise
- Can be modeled by random values *multiplied* by pixel values
- Major problem in some radar applications


(b) Speckle noise

Speckle Noise

```
>> gs = imnoise(g,'speckle');
```

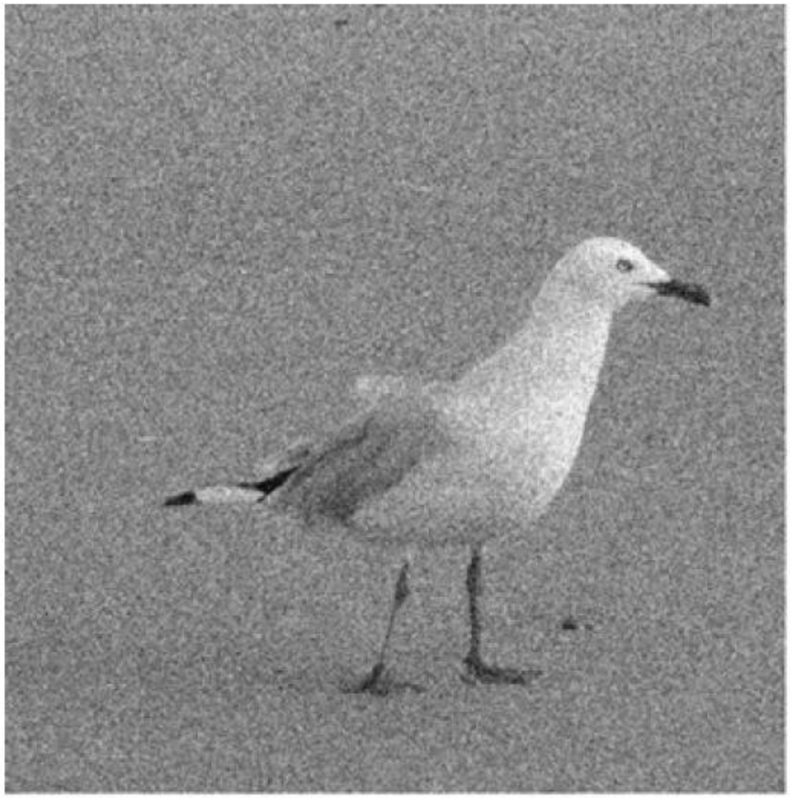
MATLAB/Octave

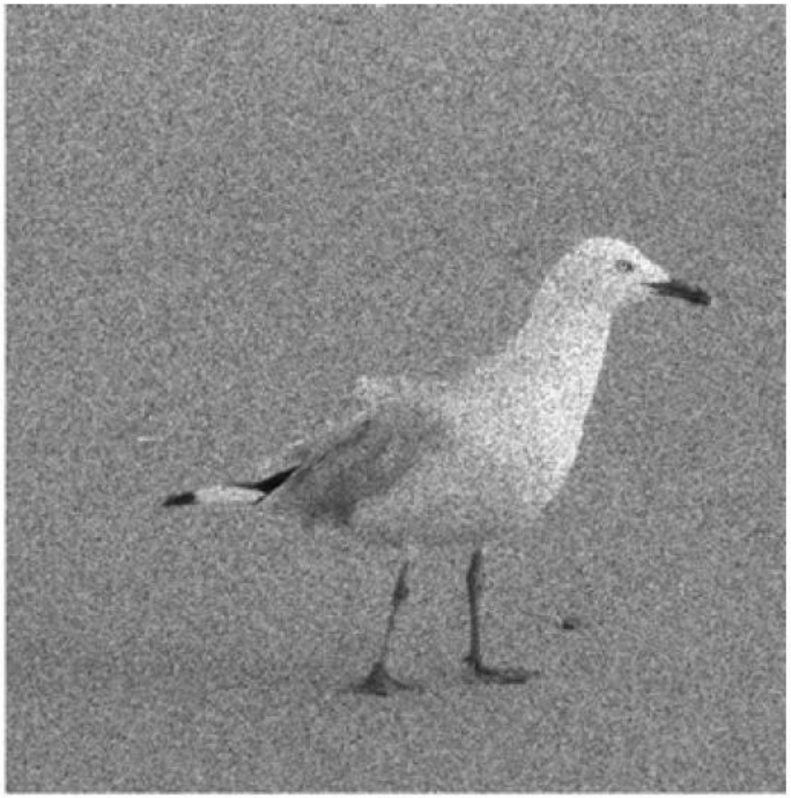
and

```
In :   gs = noise.random_noise(g,'speckle')
```

Python

Gaussian vs. Speckle Noise


(a) Gaussian noise


(b) Speckle noise

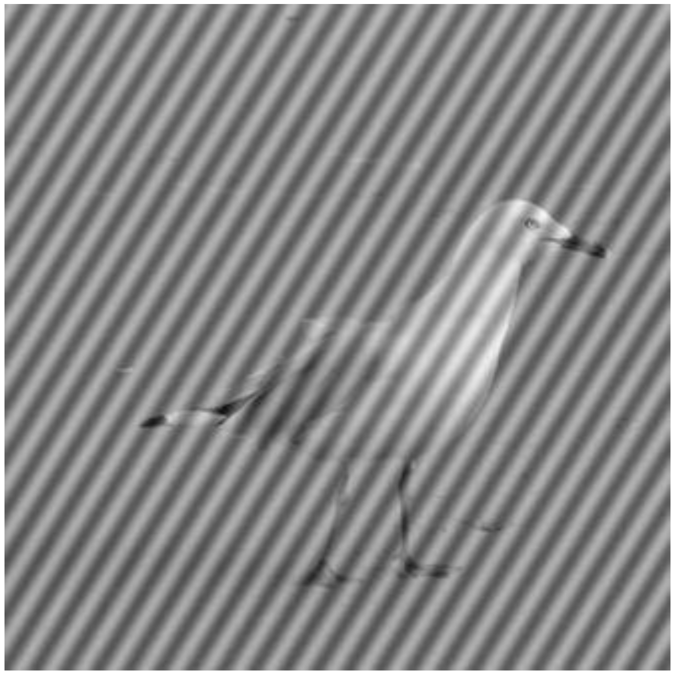
Although Gaussian noise and speckle noise appear superficially similar, they are formed by two totally different methods, and, as we shall see, require different approaches for their removal.

Periodic Noise

Occurs when an image is subject to a periodic disturbance

- Effect of bars over the image

Figure 8.3: The gull image corrupted by periodic noise



Periodic Noise

MIDWESTERN STATE UNIVERSITY

Neither function imnoise or random_noise has a periodic option, but it is quite easy to create such noise, by adding a periodic matrix (using a trigonometric function) to the image.

```
>> [rs,cs] = size(g);
>> [x,y] = meshgrid(1:rs,1:cs);
>> p = sin(x/3+y/5)+1;
>> gp = (2*im2double(g)+p/2)/3;
```

MATLAB/Octave

and in Python:

```
In :  r,c = g.shape
In :  x,y = np.mgrid(0:r,0:c].astype('float32')
In :  p = np.sin(x/3+y/3)+1.0
In :  gp = (2*skimage.util.img_as_float(g)+p/2)/3
```

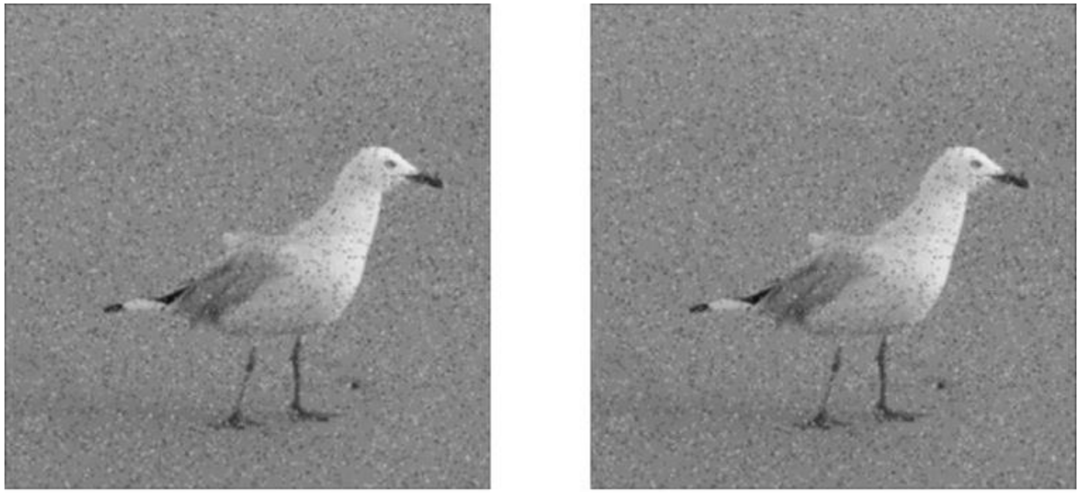
Python

Cleaning
Salt & Pepper Noise

Low Pass Filtering

- Pixels corrupted by salt & pepper noise are high frequency components of an image, can expect a low pass filter should reduce them

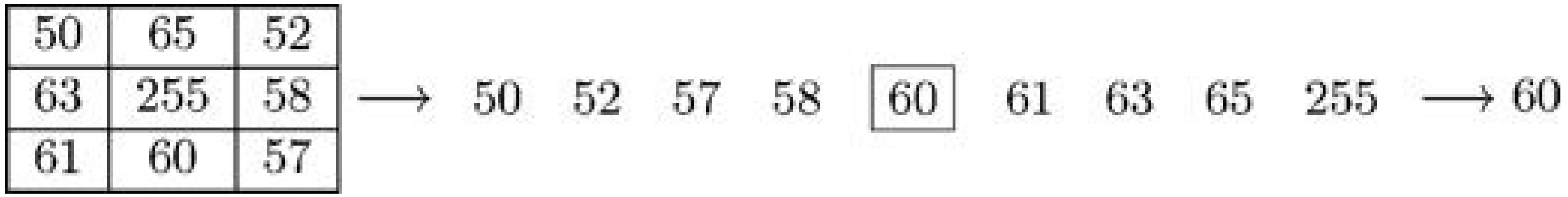
Figure 8.4: Attempting to clean salt and pepper noise with average filtering


(a) 3×3 averaging (b) 7×7 averaging

Cleaning
Salt & Pepper Noise

Median filtering

- Almost tailor made for removal of salt & pepper noise
- example of a non-linear spatial filter
- Using 3 x 3 mask, output value is median of values in the mask



50	65	52
63	255	58
61	60	57

$$\longrightarrow 50 \quad 52 \quad 57 \quad 58 \quad \boxed{60} \quad 61 \quad 63 \quad 65 \quad 255 \longrightarrow 60$$

- Very large or very small values (noisy values) end up at the top or bottom of the sorted list
- Replaces noisy value with one closer to its surroundings

Cleaning Salt & Pepper Noise

MIDWESTERN STATE UNIVERSITY

In MATLAB, median filtering is implemented by the `medfilt2` function:

```
>> gm3 = medfilt2(gsp);
```

MATLAB/Octave

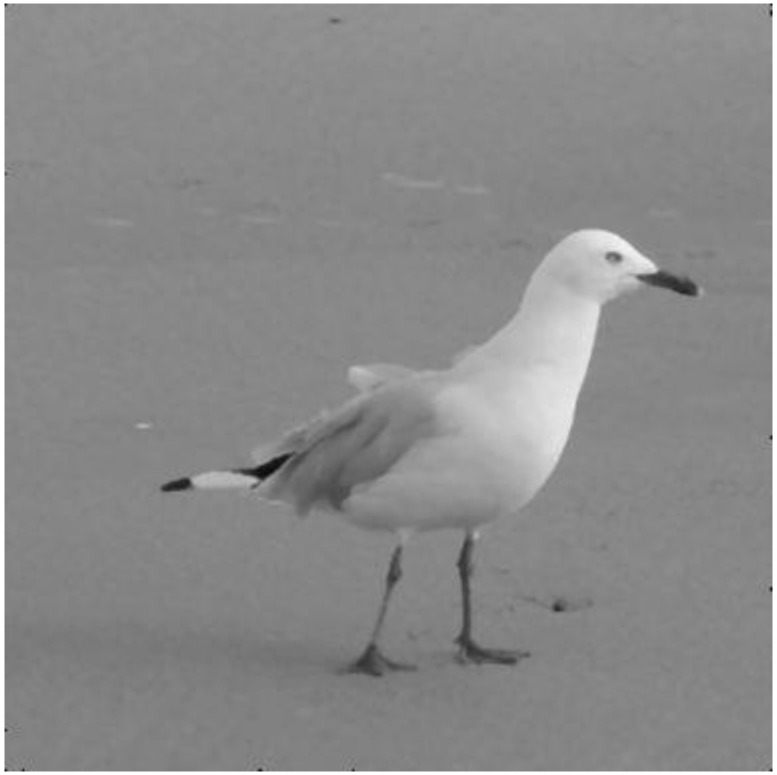
and in Python by the `median_filter` method in the `ndimage` module of `numpy`:

```
In :  gm3 = ndi.median_filter(gsp,3)
```

Python

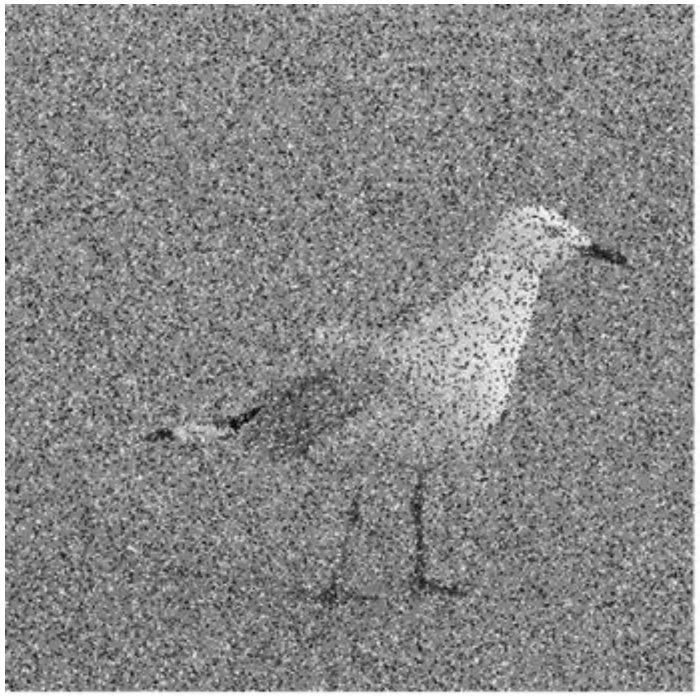
Cleaning
Salt & Pepper Noise

Figure 8.5: Cleaning salt and pepper noise with a median filter

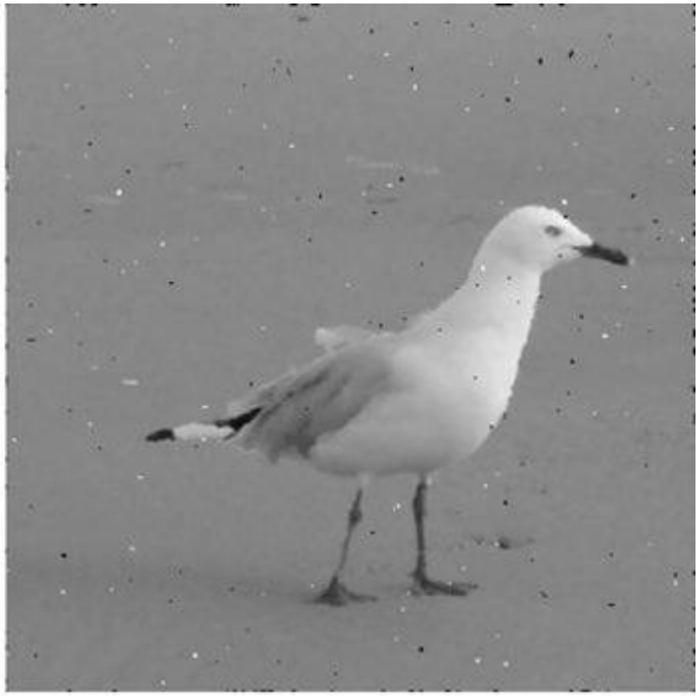


Cleaning
Salt & Pepper Noise

MORE
NOISE!!!!



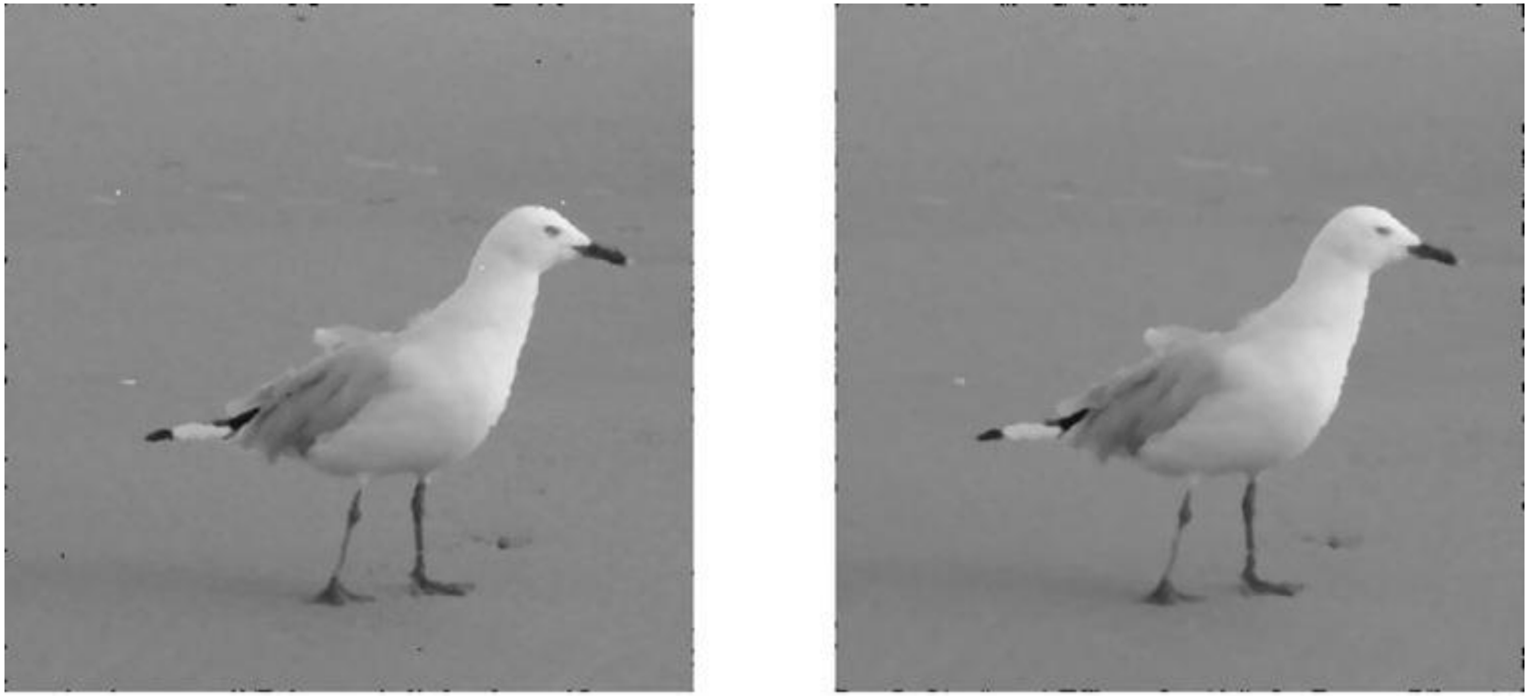
(a) 20% salt and pepper noise



(b) After median filtering

Cleaning Salt & Pepper Noise

Figure 8.7: Cleaning 20% salt and pepper noise with median filtering



(a) Using `medfilt2` twice

(b) Using a 5×5 median filter

Cleaning
Salt & Pepper Noise

Rank-Order Filtering

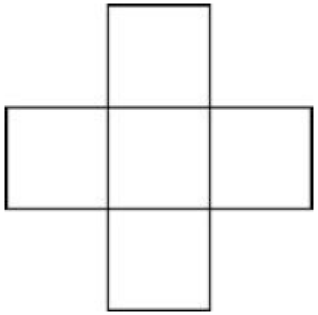
- Nth value of ordered list is chosen, for some predetermined value of n
- When to use Rank-Order filtering instead of median filtering?
    - When using a median filter over non-rectangular masks

Cleaning Salt & Pepper Noise

MIDWESTERN STATE UNIVERSITY

Rank-Order Filtering

For example, if we decided to use as a mask a 3 × 3 cross shape:



then the median would be the third of these values when sorted. The MATLAB/Octave command to do this is

```
>> co = ordfilt2(gsp,3,[0 1 0;1 1 1;0 1 0]);
```

MATLAB/Octave

In general, the second argument of `ordfilt2` gives the value of the ordered set to take, and the third element gives the *domain*; the non-zero values of which specify the mask. If we wish to use a cross with size and width 5 (so containing nine elements), we can use:

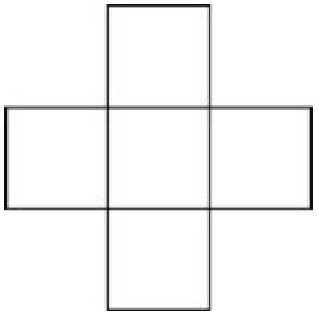
```
>> ordfilt2(gsp,5,[0 0 1 0 0;0 0 1 0 0;1 1 1 1 1;0 0 1 0 0;0 0 1 0 0])
```

MATLAB/Octave

Cleaning Salt & Pepper Noise

Rank-Order Filtering

For example, if we decided to use as a mask a 3 × 3 cross shape:



In Python non-rectangular masks are handled with the `median_filter` method itself, by entering the mask (known in this context as the *footprint*) to be used:

```
In :  cross = array([[0,1,0],[1,1,1],[0,1,0]])
In :  co = ndi.median_filter(gsp,footprint=cross)
```

Python

When using `median_filter` there is no need to indicate which number to choose as output, as the output will be the median value of the elements underneath the 1's of the footprint.

Cleaning Salt & Pepper Noise

Outlier Method

- treat noisy pixels as *outliers*
    - Outliers are pixels whose gray values are significantly different from those of their neighbors
- Approach:
    1. Choose a threshold value *D*
    2. For a given pixel, compare its value *p* with the mean *m* of the values of its eight neighbors
    3. If $|p - m| > D$, then classify the pixel as noisy
    4. If the pixel is noisy, replace its value with *m*
- *Choosing the right* D *value is essential!*

Cleaning
Salt & Pepper Noise

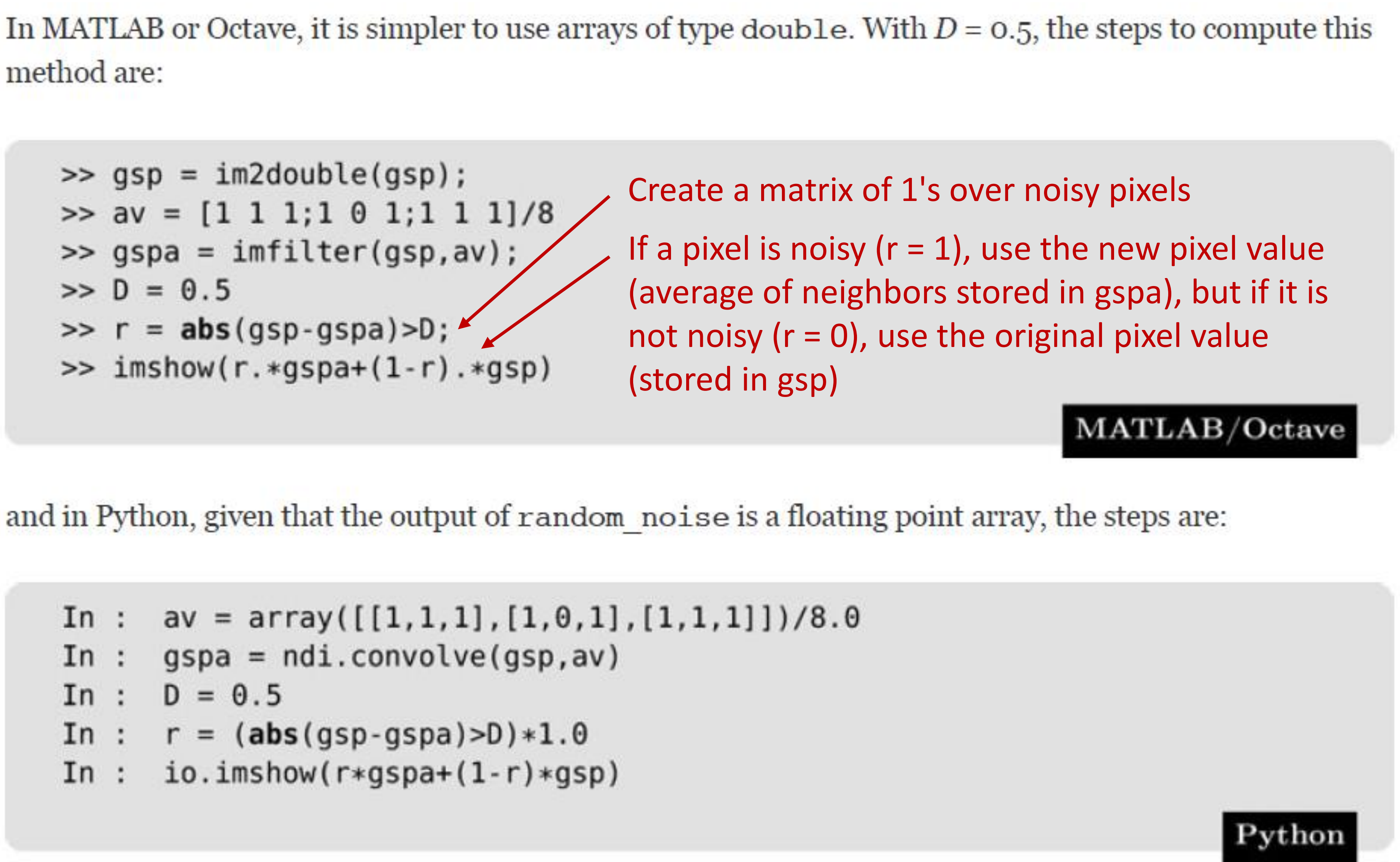
Outlier Method

There is no MATLAB function for doing this, but it is very easy to write one. First, we can calculate the average of a pixel's eight neighbors by convolving with the linear filter

$$\frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.125 & 0.125 & 0.125 \\ 0.125 & 0 & 0.125 \\ 0.125 & 0.125 & 0.125 \end{bmatrix}$$

We can then produce a matrix r consisting of 1's at only those places where the difference of the original and the filter are greater than D; that is, where pixels are noisy. Then $1 - r$ will consist of ones at only those places where pixels are not noisy. Multiplying r by the filter replaces noisy values with averages; multiplying $1 - r$ with original values gives the rest of the output.

Cleaning Salt & Pepper Noise



In MATLAB or Octave, it is simpler to use arrays of type `double`. With $D = 0.5$, the steps to compute this method are:

```
>> gsp = im2double(gsp);
>> av = [1 1 1;1 0 1;1 1 1]/8
>> gspa = imfilter(gsp,av);
>> D = 0.5
>> r = abs(gsp-gspa)>D;
>> imshow(r.*gspa+(1-r).*gsp)
```

Create a matrix of 1's over noisy pixels

If a pixel is noisy (r = 1), use the new pixel value (average of neighbors stored in gspa), but if it is not noisy (r = 0), use the original pixel value (stored in gsp)

MATLAB/Octave

and in Python, given that the output of `random_noise` is a floating point array, the steps are:

```
In :  av = array([[1,1,1],[1,0,1],[1,1,1]])/8.0
In :  gspa = ndi.convolve(gsp,av)
In :  D = 0.5
In :  r = (abs(gsp-gspa)>D)*1.0
In :  io.imshow(r*gspa+(1-r)*gsp)
```

Python

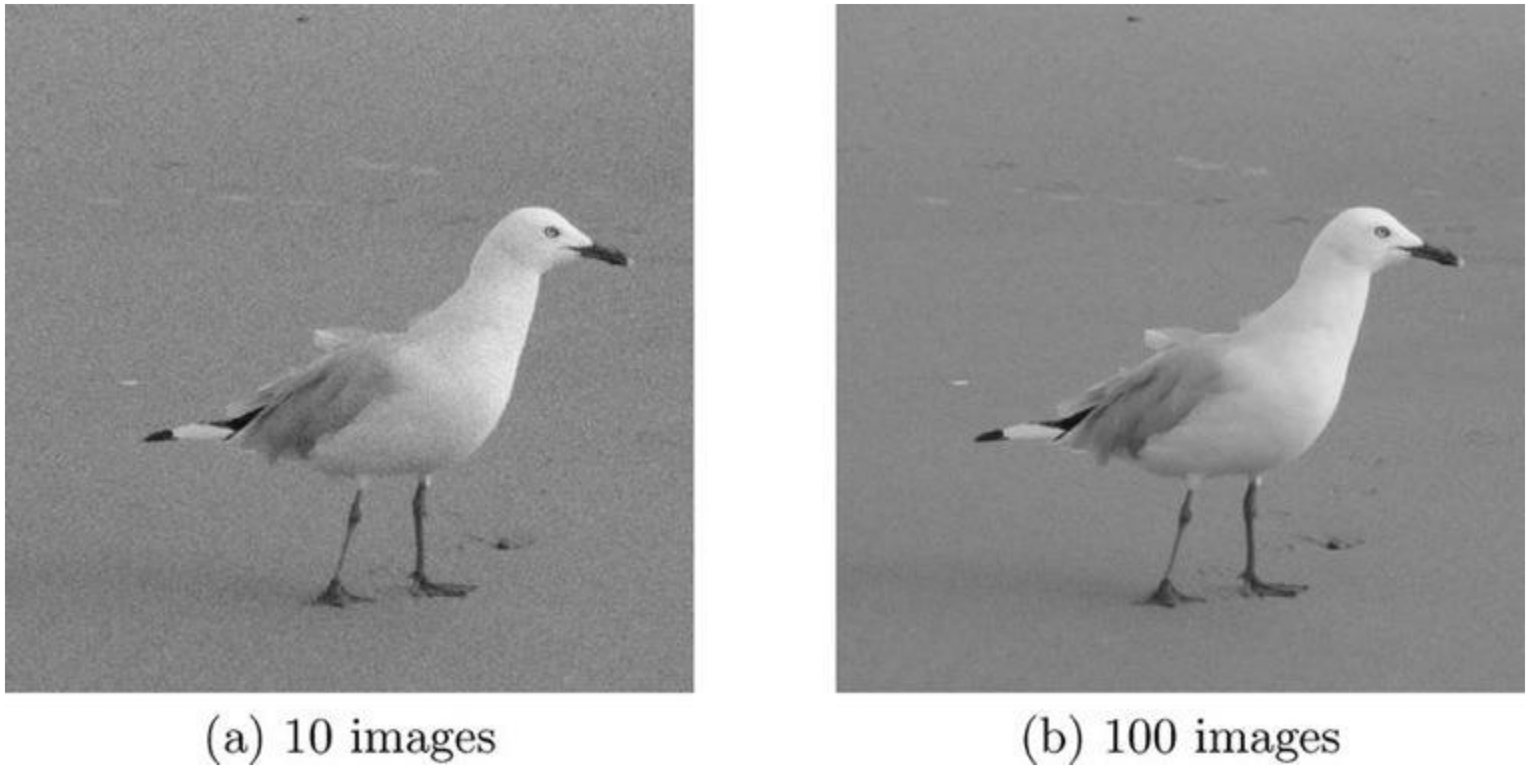
Cleaning Gaussian Noise

Image Averaging

- Sometimes we have different copies of an image corrupted with Gaussian noise
- Example:
  - Satellite imaging
    - Satellite passes over the same spot many times, taking pictures, many different images of the same place
  - Microscopy
    - Many different images of the same object are taken
- When these types of problems occur, its very easy to take the average of all the images

Cleaning Gaussian Noise

Figure 8.9: Image averaging to remove Gaussian noise



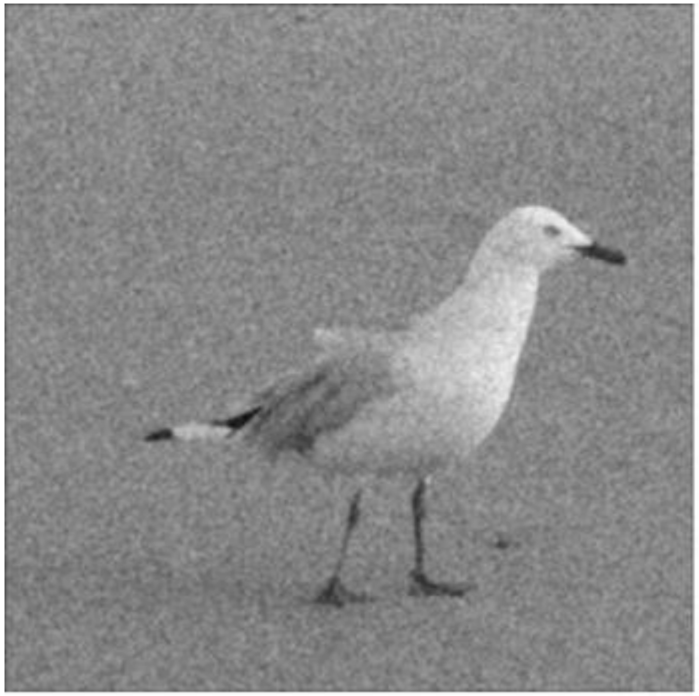
Cleaning Gaussian Noise

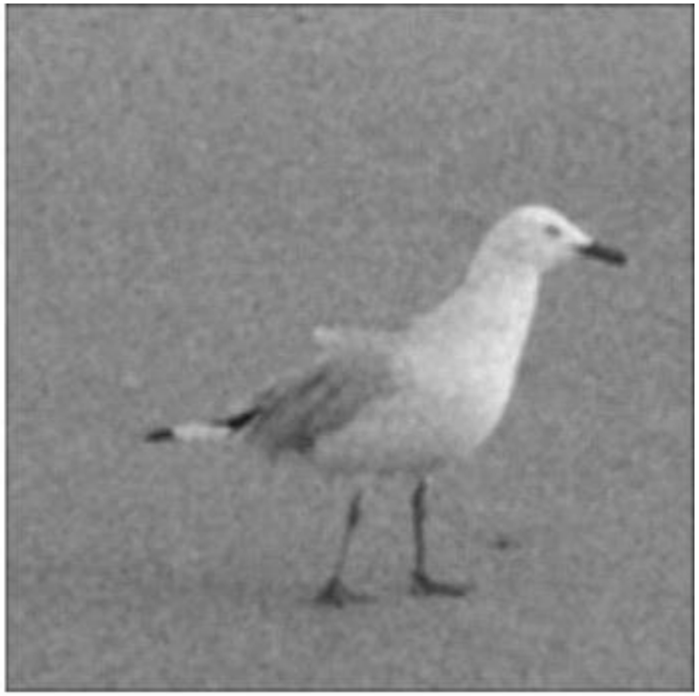
Average Filtering

- If Gaussian noise has mean 0, expect average filter to average noise to 0
  - Larger the size of filter mask, the closer to 0
- Unfortunately this tends to blur an image
- Sometimes the tradeoff of blurring for noise reduction is worthwhile

Cleaning Gaussian Noise

Figure 8.10: Using averaging filtering to remove Gaussian noise


(a) 4×3 averaging


(b) 5×5 averaging

Cleaning Gaussian Noise

Adaptive Filtering

- Class of filters that change their characteristics according to the values of the grayscales under the mask
- Cleans Guassian noise by using local statistical properties of the values under the mask
- *Minimum mean-square error filter*
  - Non-linear spatial filter implemented by applying a function to the gray values under the mask
  - Name reflects that this filter attempts to minimize the square of the difference between the input and output images

Cleaning Gaussian Noise

MIDWESTERN STATE UNIVERSITY

Adaptive Filtering

We will use the wiener filter (developed by Norbert Wiener in 1942), which can take an optional parameter indicating the size of the mask to be used. The default size is 3 × 3. We shall create four images:

```
>> gw1 = wiener2(gg);
>> gw2 = wiener2(gg,[5,5]);
>> gw3 = wiener2(gg,[7,7]);
>> gw4 = wiener2(gg,[9,9]);
```

MATLAB/Octave

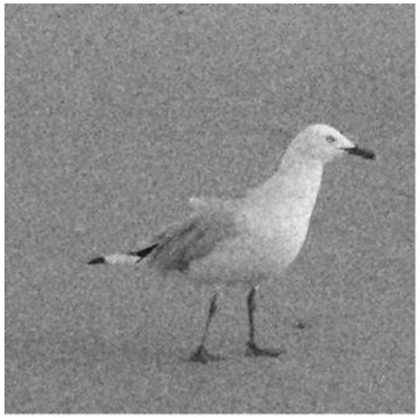
In Python the Wiener filter is implemented by the `wiener` method of the `signal` module of `scipy`:

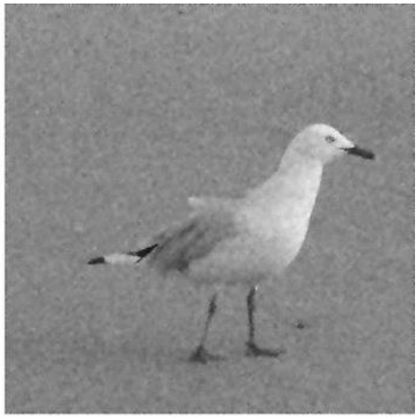
```
In :  from scipy.signal import wiener
In :  cw1 = wiener(cg,[3,3])
```

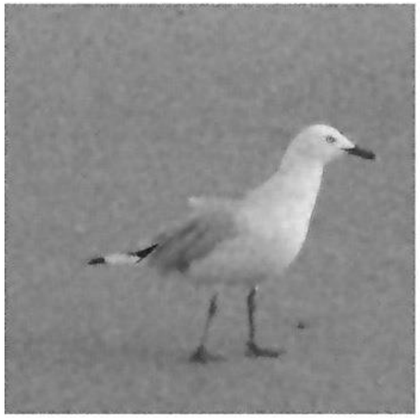
Python

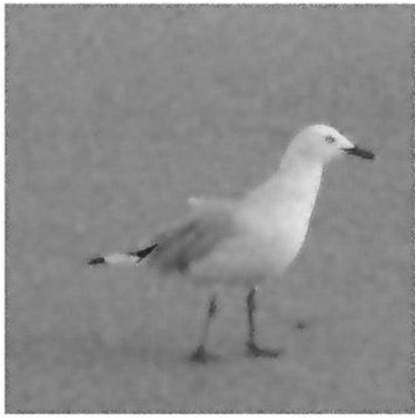
Cleaning Gaussian Noise

Figure 8.11: Examples of adaptive filtering to remove Gaussian noise


(a) 3 × 3 filtering

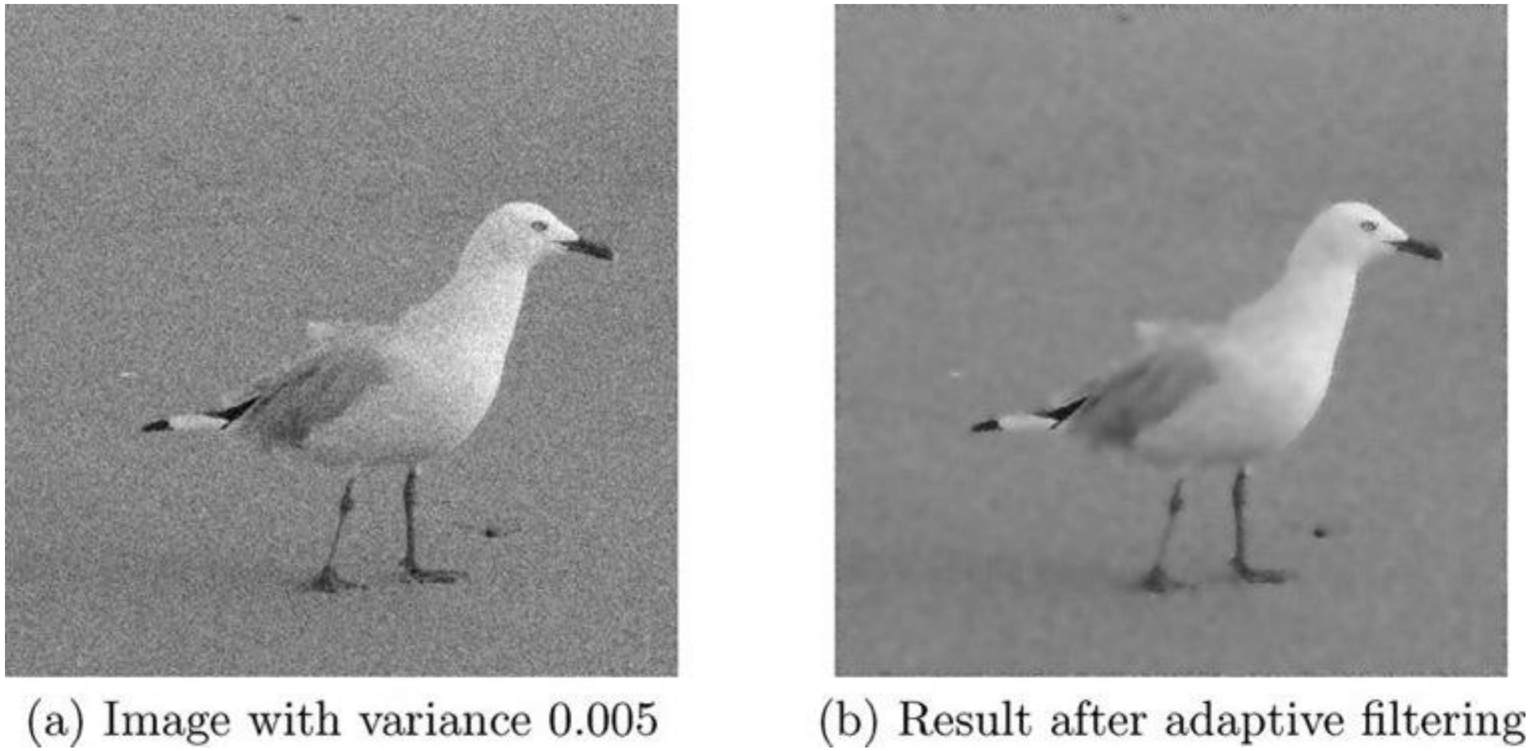

(b) 5 × 5 filtering


(c) 7 × 7 filtering


(d) 9 × 9 filtering

Cleaning Gaussian Noise

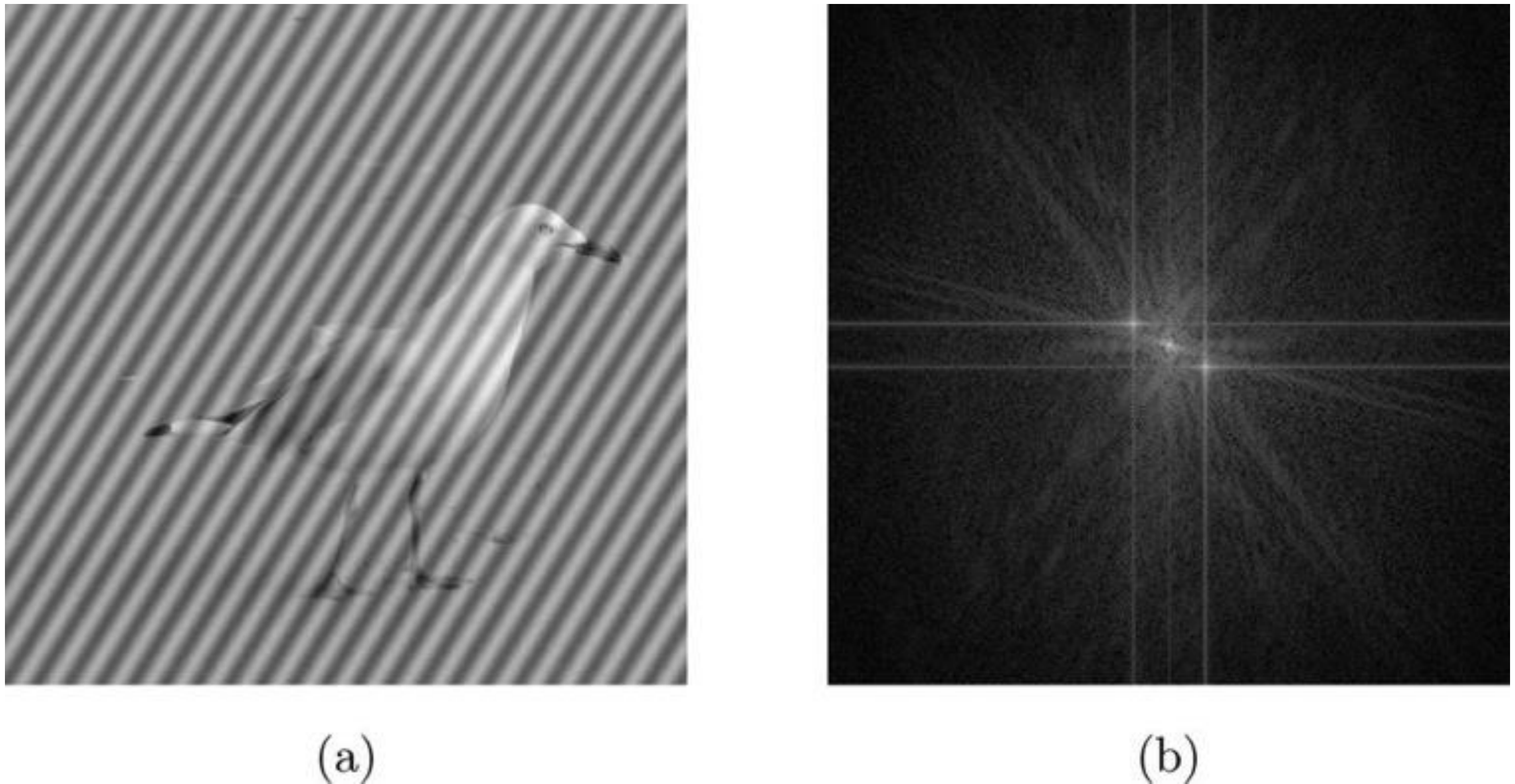
Figure 8.12: Using adaptive filtering to remove Gaussian noise with low variance


(a) Image with variance 0.005
(b) Result after adaptive filtering

Removal of Periodic Noise

Periodic noise occurs when imaging equipment is subject to electronic disturbances of a repeating nature

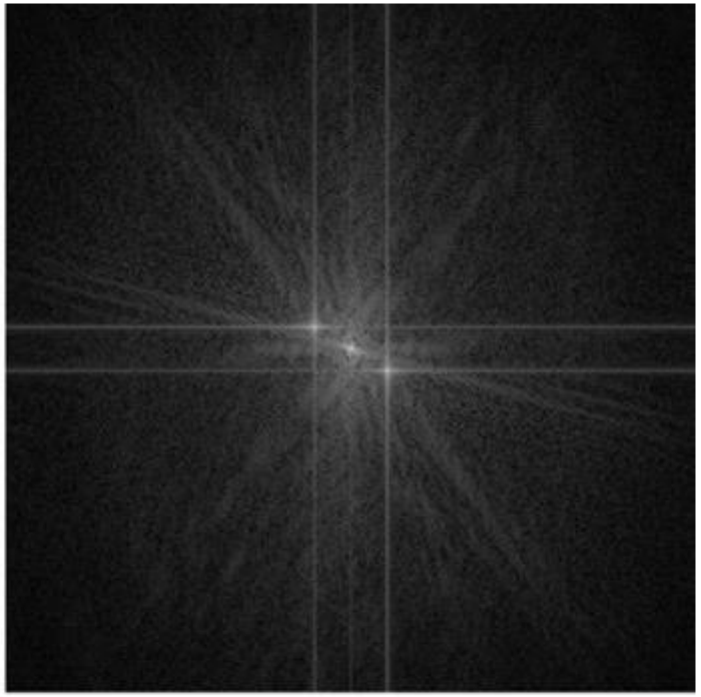
Figure 8.13: The gull image (a) with periodic noise and (b) its transform



Removal of Periodic Noise

- Extra 2 spikes in 8.3(b) correspond to the added noise
- In general, the tighter the period of the noise, the further from the center the spikes will be
  - Because small period corresponds to high frequency
- Methods to remove spikes
  - Band reject filtering
  - Criss-cross filtering

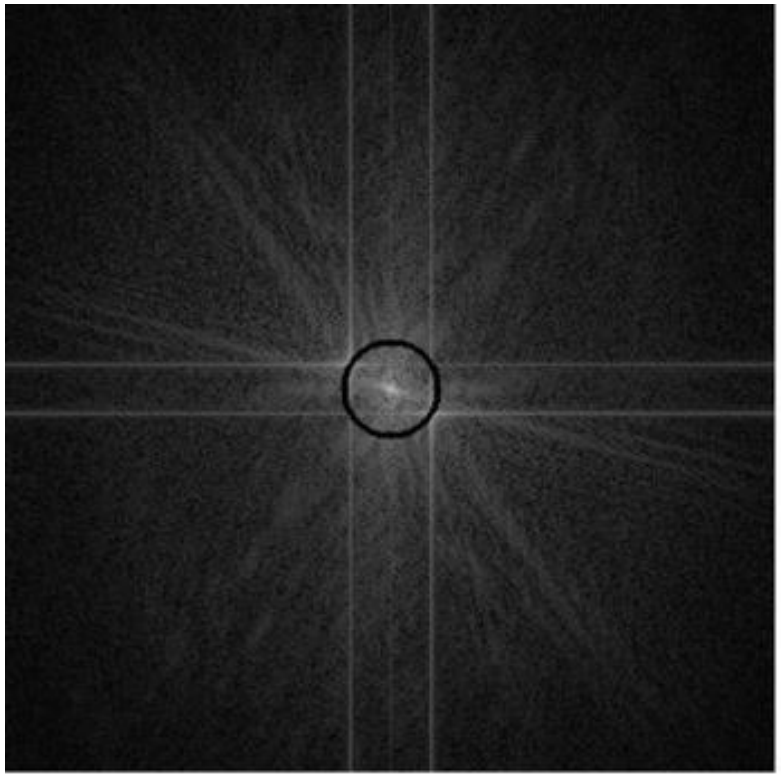
Figure 8.13: The gull image (b) its transform

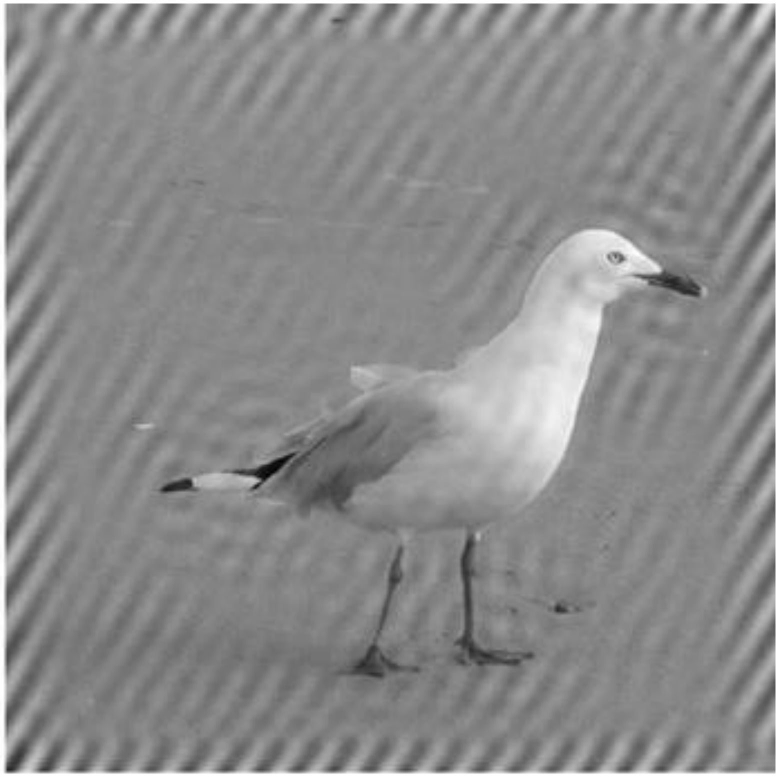


(b)

Band Reject Filtering

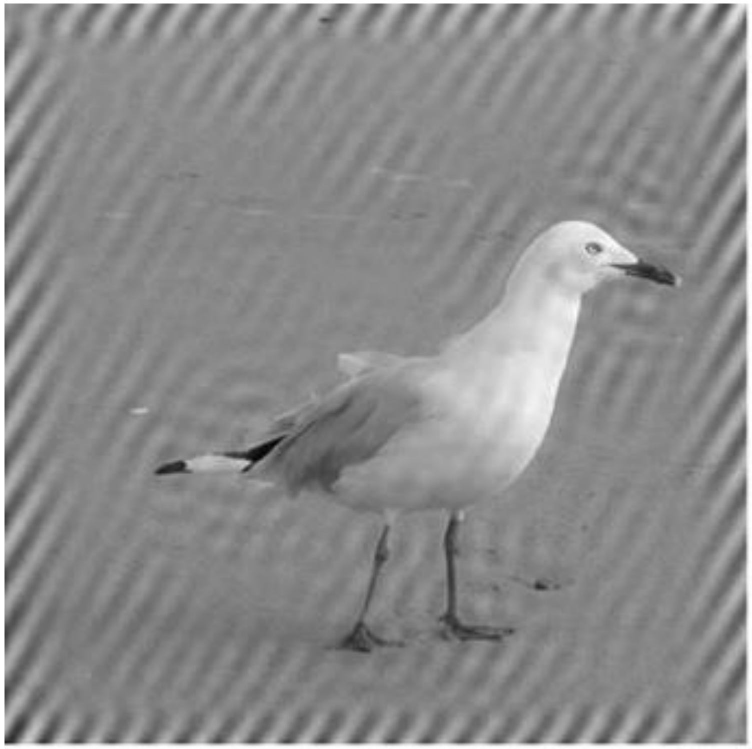
Figure 8.14: Removing periodic noise with a band-reject filter


(a) A band-reject filter

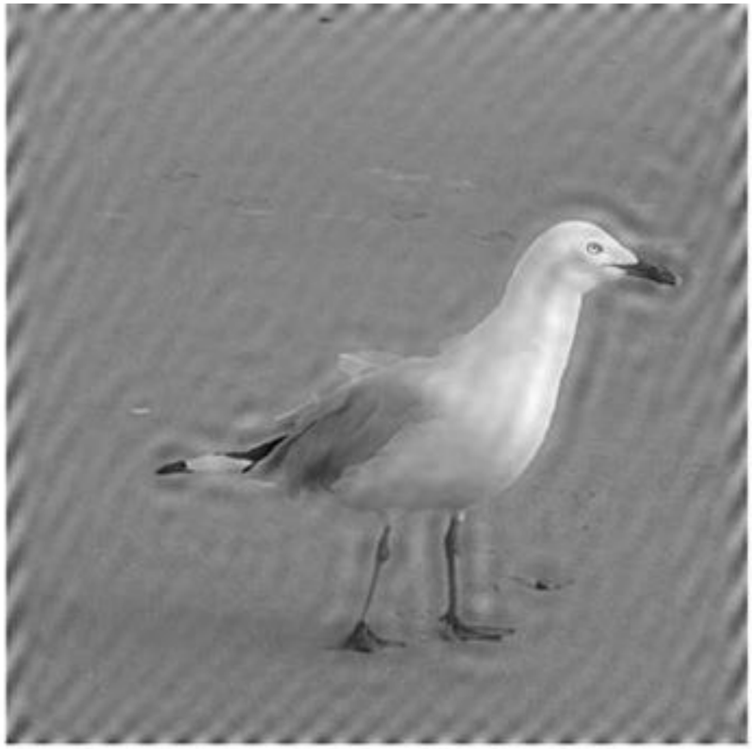

(b) After inversion

Band Reject Filtering

Figure 8.15: Removing periodic noise with wider band-reject filters



(a) Band-reject filter with $k = 2$



(b) Band-reject filter with $k = 5$

Criss-Cross Filtering

A criss-cross filter sets the rows and columns of the transform that contain the spikes to zero

- The more rows and columns of the transform that are set to zero, the better the noise reduction

Figure 8.16: Removing periodic noise with a criss-cross filter