Denoising Images Using Bayesian Analysis

George Kacoyanis

The goal of the project is to use Bayesian analysis and Gibbs sample to reduce noise from noisy images. Bayesian image processing based on Markov random fields (MRFs) is an important framework in the field of image processing. An MRF is an undirected graph representation of probability distribution, and many applications of MRFs exist in the image processing and computer vision fields. MRFs have also been applied to other research fields, including traffic engineering and earth science. In Bayesian image processing, the objective image can be inferred based on the posterior probability distribution.

```
load ImageFile1.mat;
D = I;
sig1 =1;
sig2 = 0.01;
[n,m] = size(I);
Niter =20;
```

First is to read in the data file and set the file to an empty matrix of size [n,m].

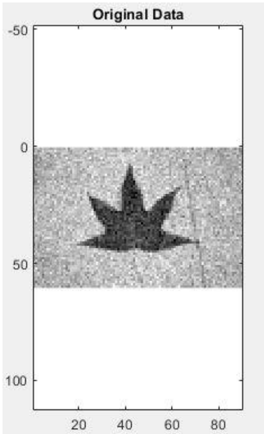
```
II(2:n+1, 2:m+1) = I;
II(1,1) = 0;
II(n+2,m+2) = 0;
II(1,2:m+1) = I(n,:);
II(n+2,2:m+1) = I(1,:);
II(2:n+1,1)= I(:,m);
II(2:n+1,m+2) = I(:,1);
for i=1:n
    for j = 1:m
        for k=1:10
```

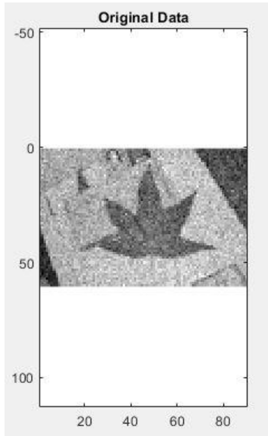
Checks for the neighbors of the observation to create the prior model to form a Markov Random Field. From that we form the mean of the data's neighbors and create the observation model Y = mu + sig1*randn to represent D = I + W to specify the likelihood function.

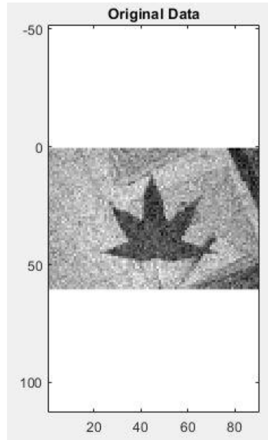
```
mu = (II(i,j+1) + II(i+2,j+1) + II(i+1,j) + II(i+1,j+2))/4;
Y = mu + sig1*randn;

%[I(i,j), Y, D(i,j)]

fact = f(Y,D(i,j),sig2)/f(I(i,j),D(i,j),sig2);
rho = min(fact,1);
if rand < rho
    I(i,j)= Y;
```

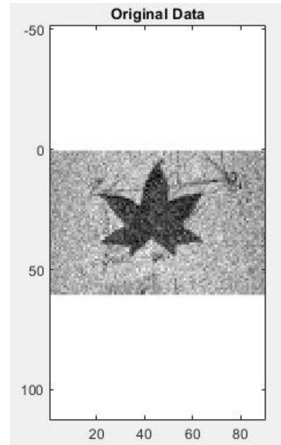
Uses the sigma 1 and 2 to create the posterior density from the observation and prior to generate samples of 20 iterations from. Using Gibbs sampler, we implement sampling from the full conditional of the prior and use Metropolis Hastings to sample from the conditionals of the posterior. We create samples with the proposal density(f(Y,D(i,j),sig2) over the target density(f(I(I,j),D(i,j),sig2) to create the posterior density.

Orignial Data







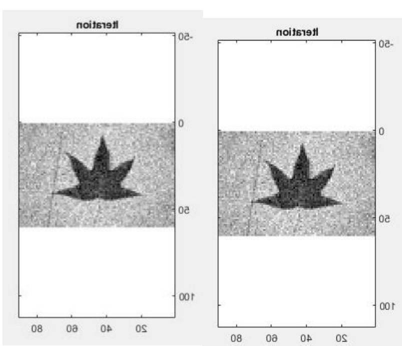


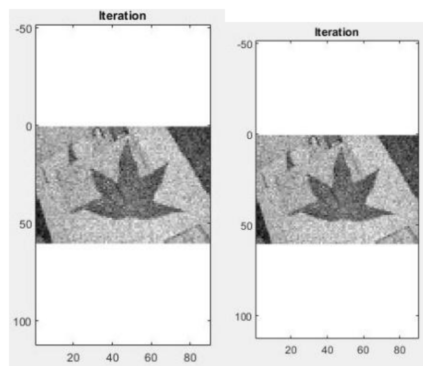
1 2 3 4

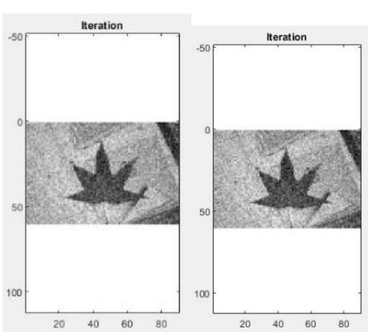
Sigma1=0.01

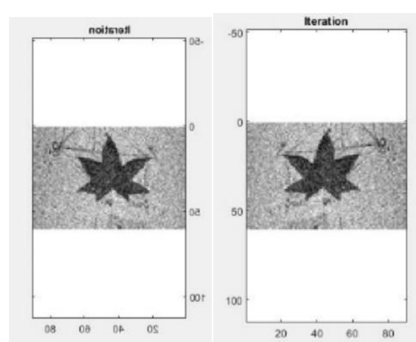
Sigma2=0.01

1 iteration 20 iterations



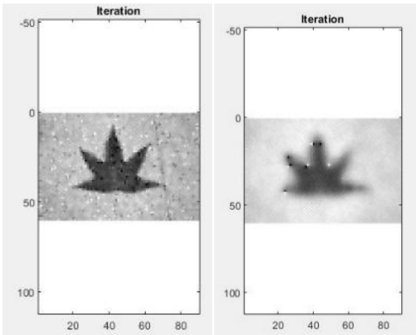


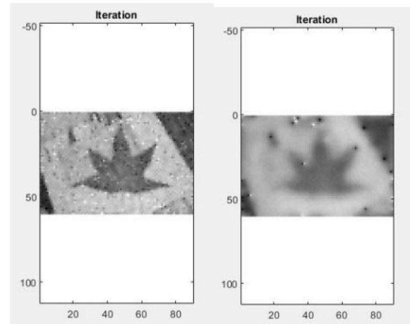


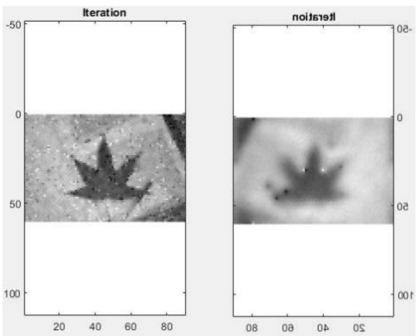


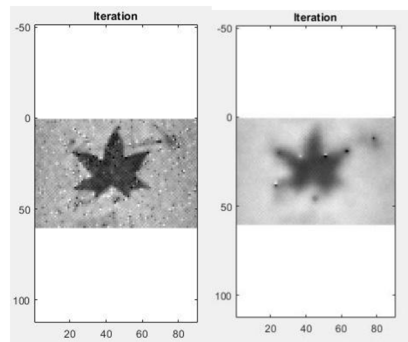
Sigma1=0.01

Sigma2=0.1



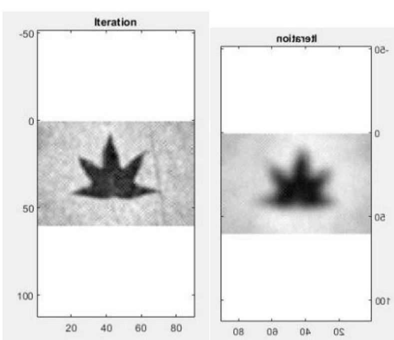


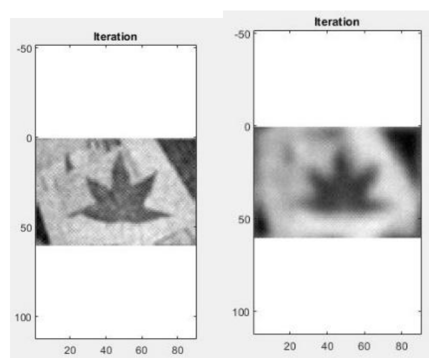


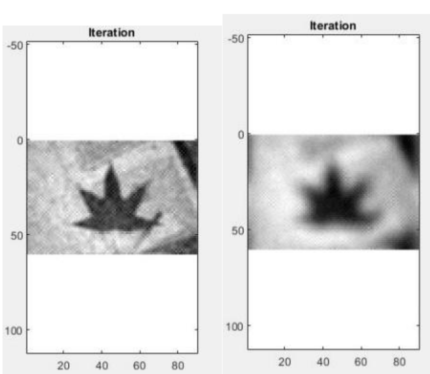


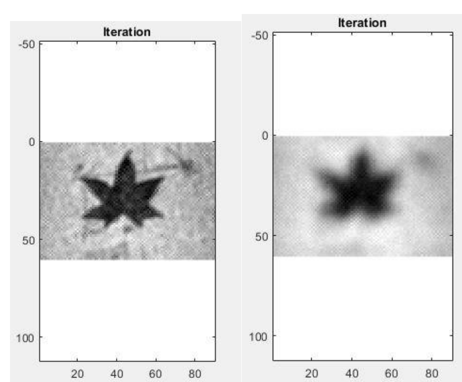
Sigma1=0.01

Sigma2=0.5



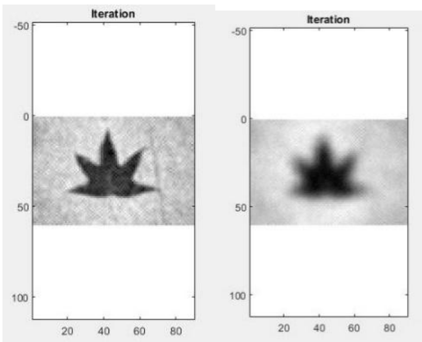


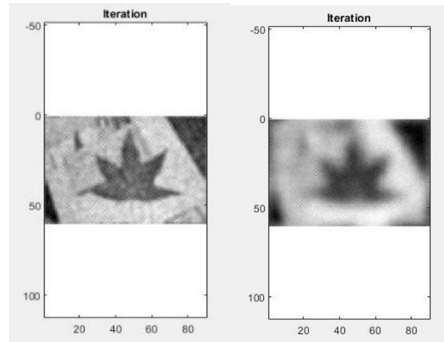


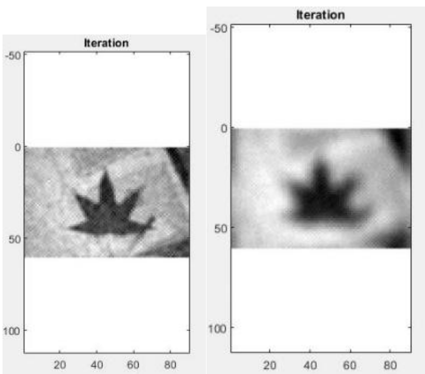


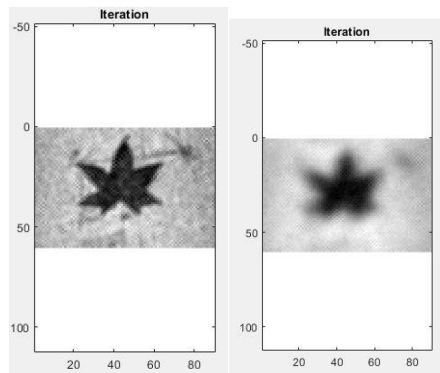
Sigma1=0.01

Sigma2=1



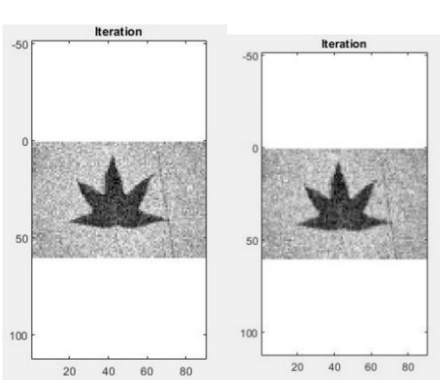


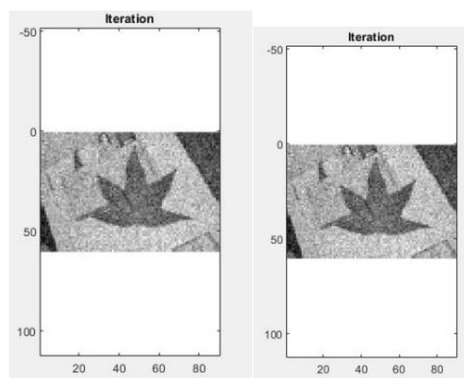


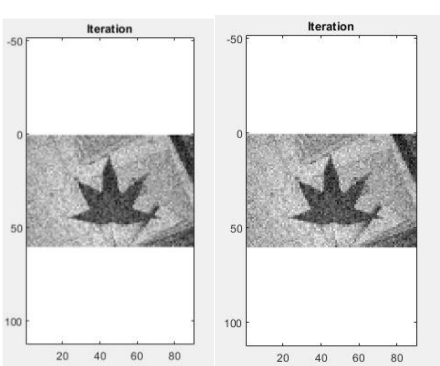


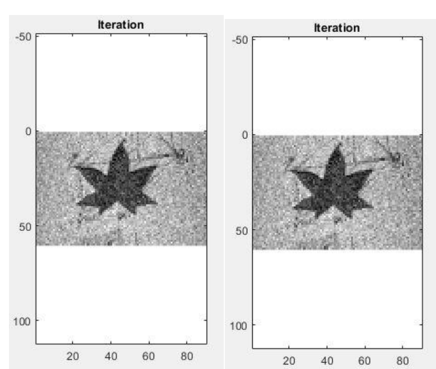
Sigma1=0.1

Sigma2=0.01



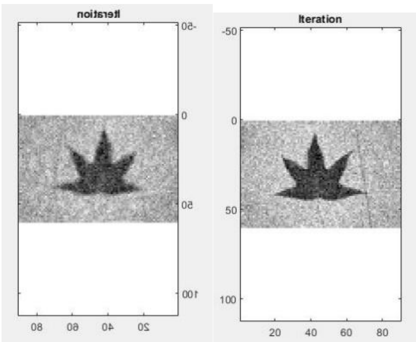


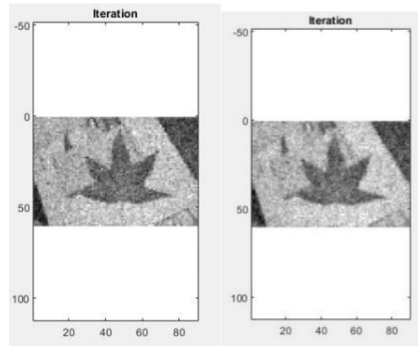


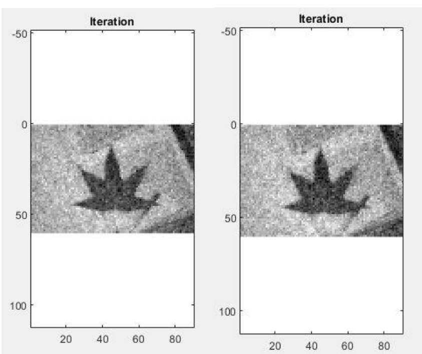


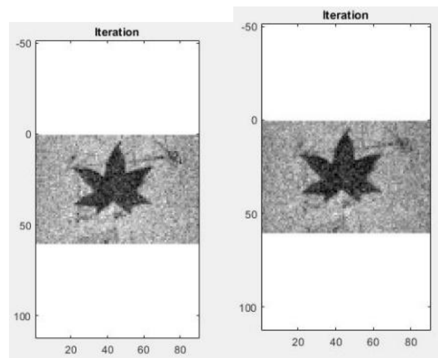
Sigma1=0.1

Sigma2=0.1



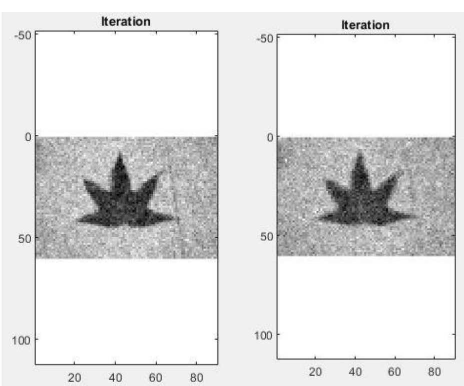


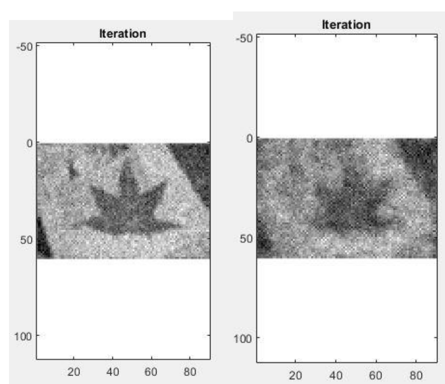


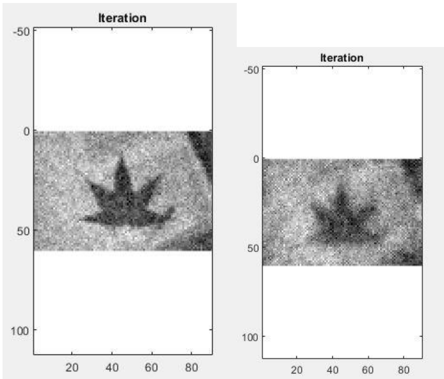


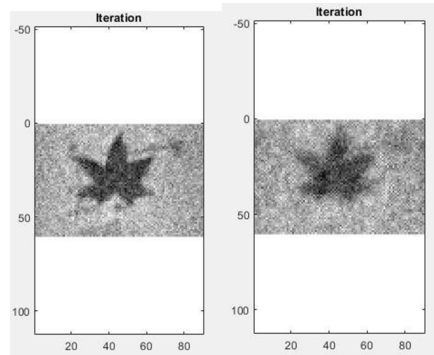
Sigma1=0.1

Sigma2=0.5



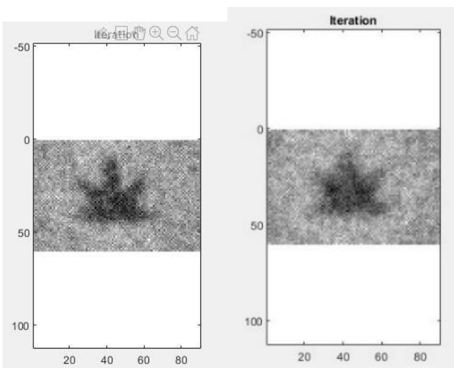


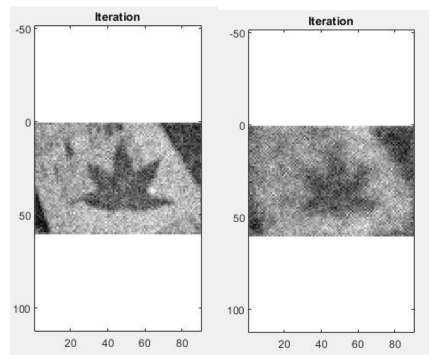


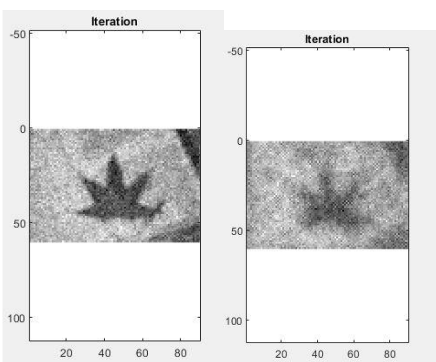


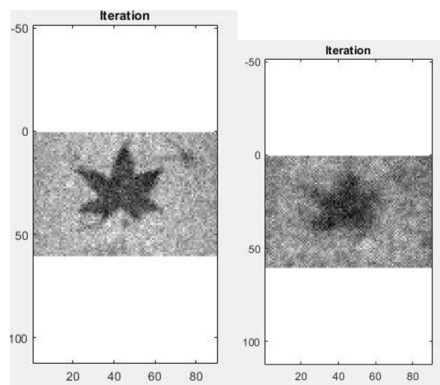
Sigma1=0.1

Sigma2=1



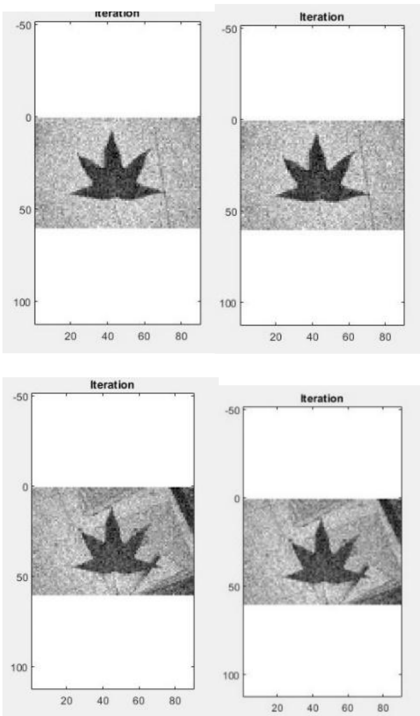


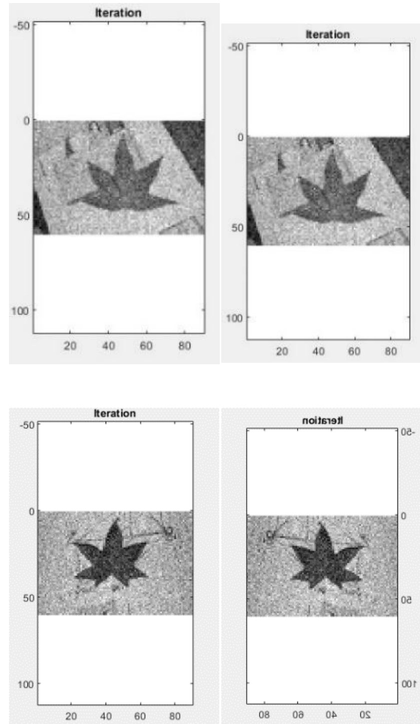




Sigma1=0.5

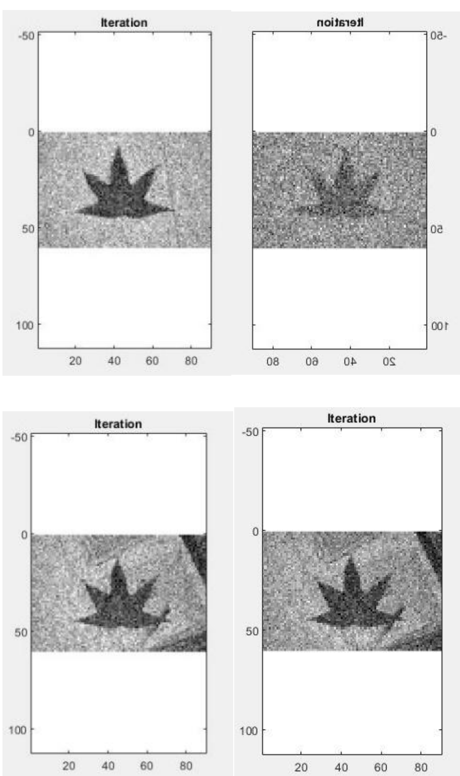
Sigma2=0.01

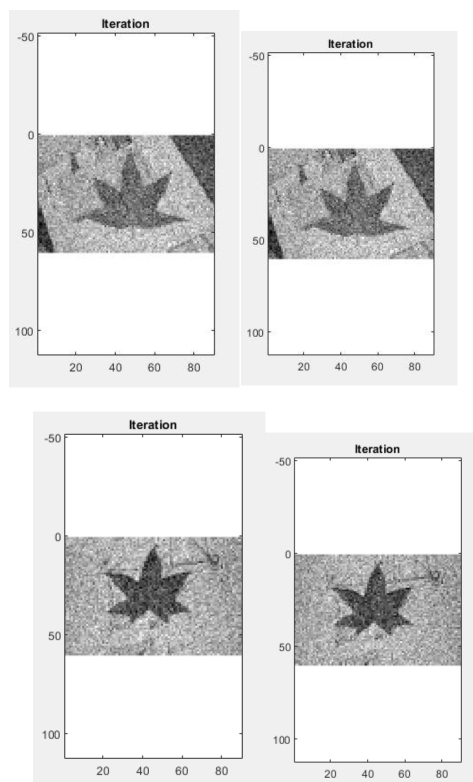




Sigma1=0.5

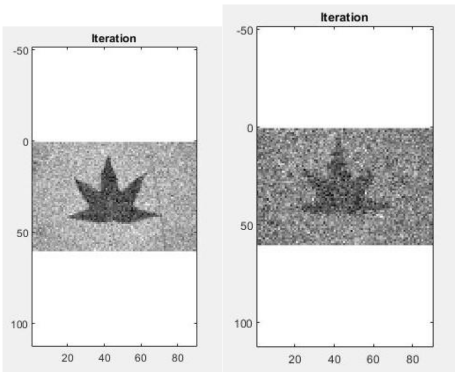
Sigma2=0.1

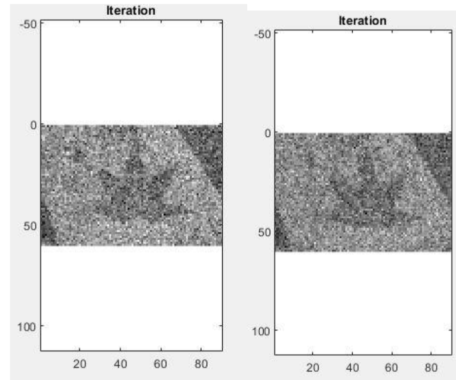


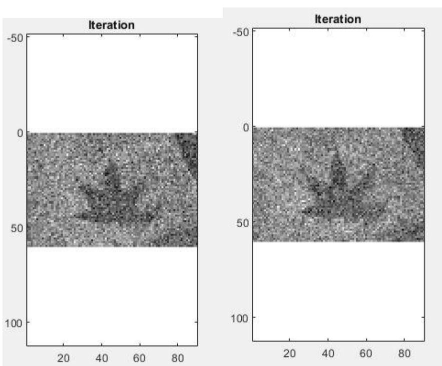


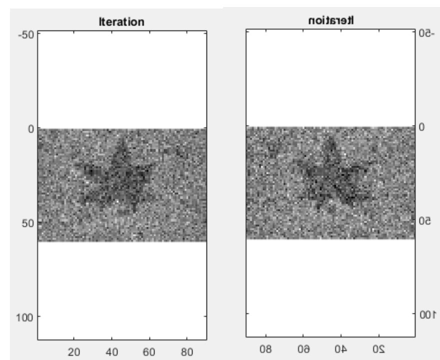
Sigma1=0.5

Sigma2=0.5



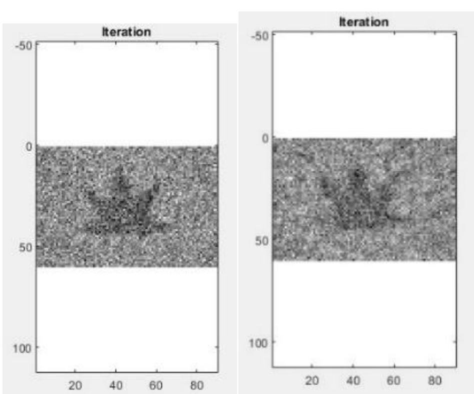


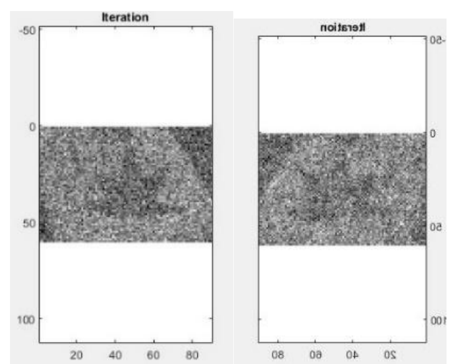


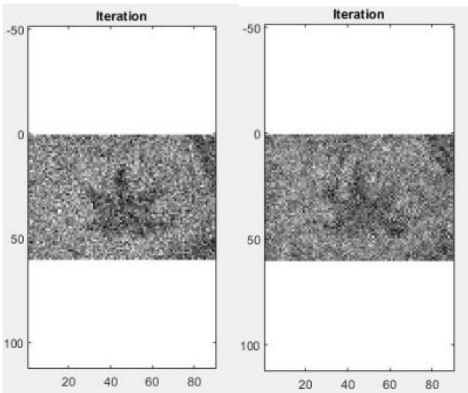


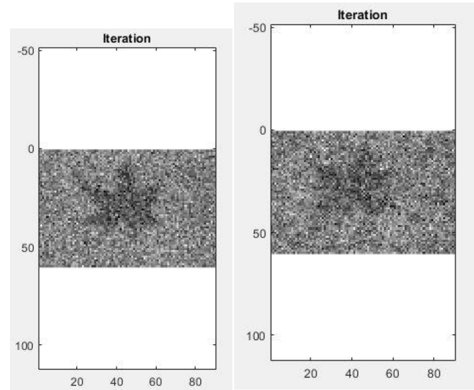
Sigma1=0.5

Sigma2=1



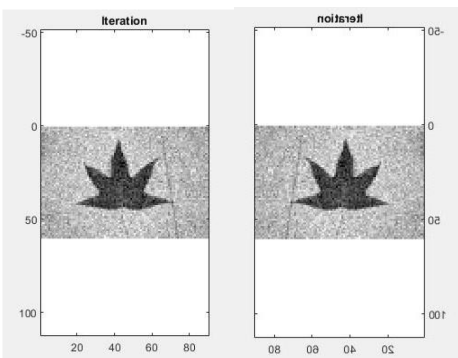


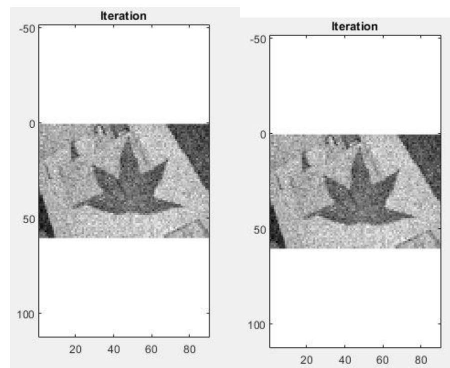


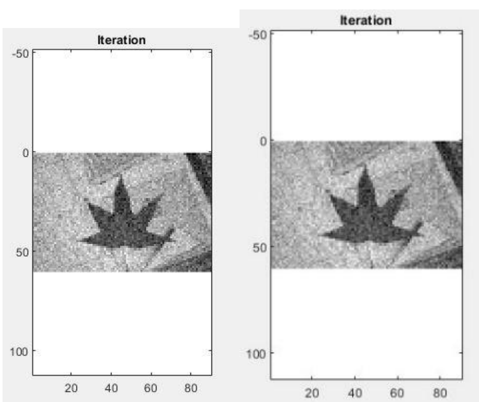


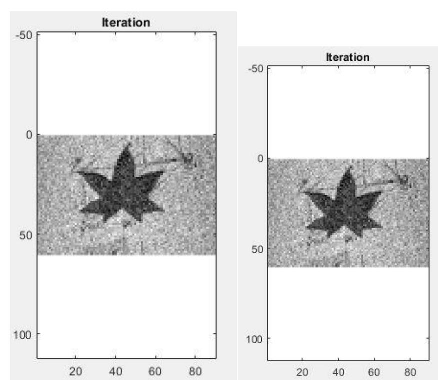
Sigma1=1

Sigma2=0.01



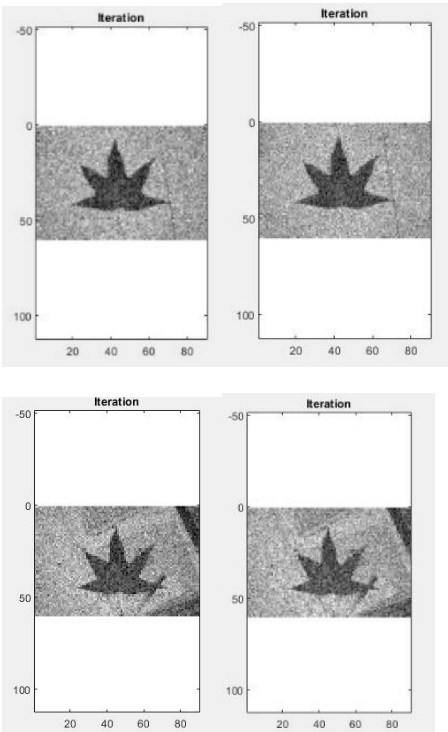


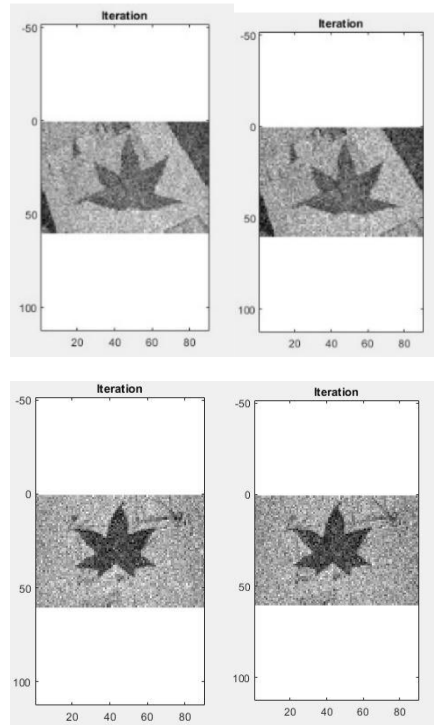




Sigma1=1

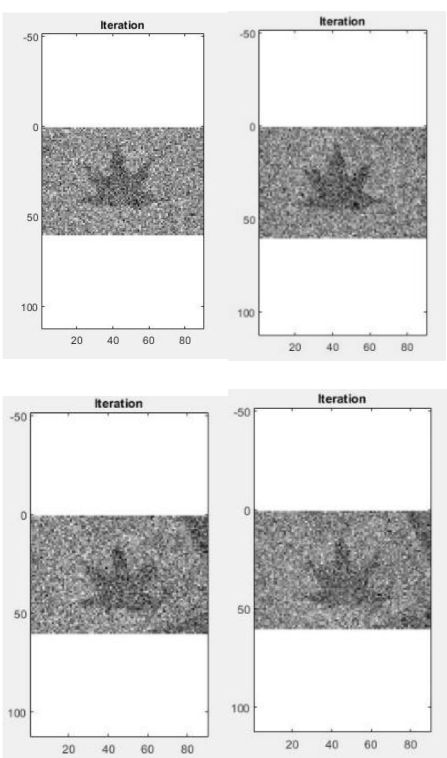
Sigma2=0.1

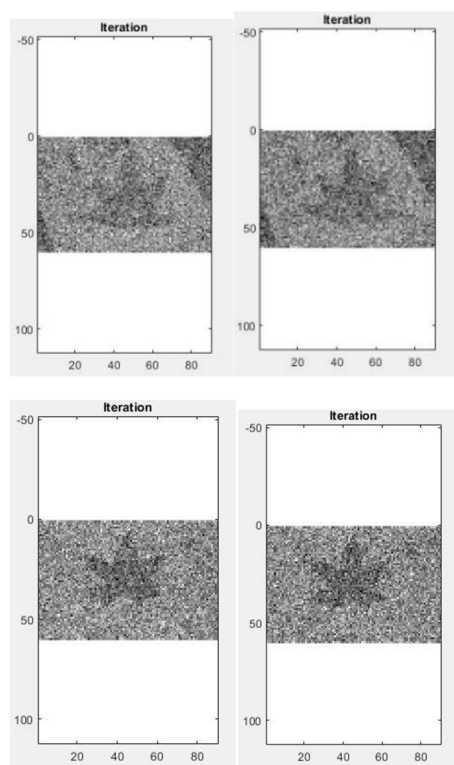




Sigma1=1

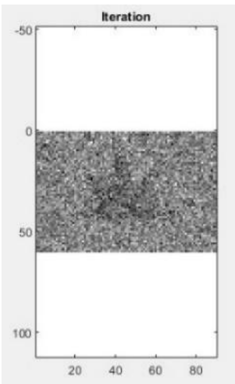
Sigma2=0.5

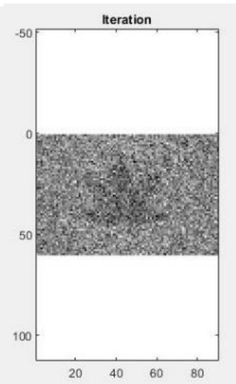


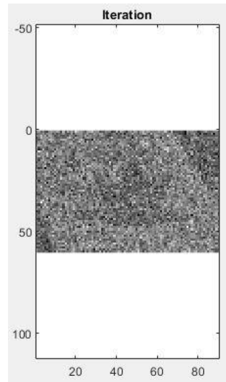


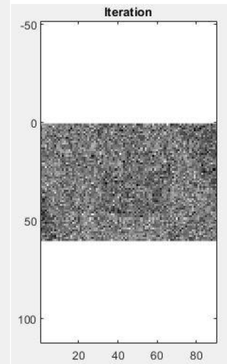
Sigma1=1

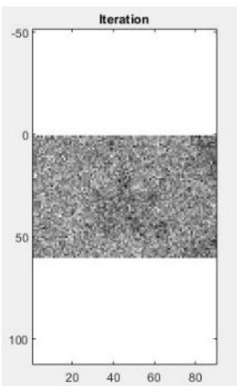
Sigma2=1

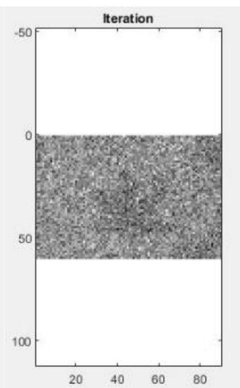


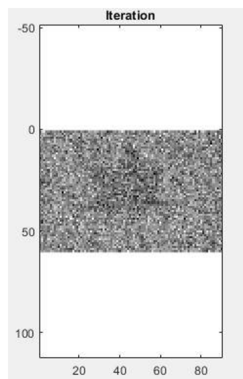


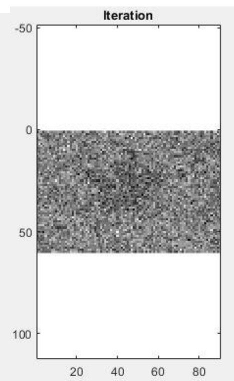












When sigma 2 was at 1 and 0.5, there were faster run times than when the sigma 2 was set significantly smaller. This is because there is less variance in the data, and the computer takes more time to generate more accurate data. When the denoising process was not as successful, the run time was faster for each iteration. The greater the differences between Sigma 1 and Sigma 2, when Sigma 1 is greater, the sharper the results of the image. According to the results, the proposed sigma 1 is sharper than the other methods. It should be noted that the difference between the methods is in whether Algorithm 1 is implemented using the MCMC. It should be noted that the difference between the methods is in whether Algorithm 1 is implemented using the MCMC. Therefore, the image denoising results of these method are all the same.

Bibliography

Givens, Geof H., and Jennifer A. Hoeting. *Computational Statistics*. Wiley, 2013.