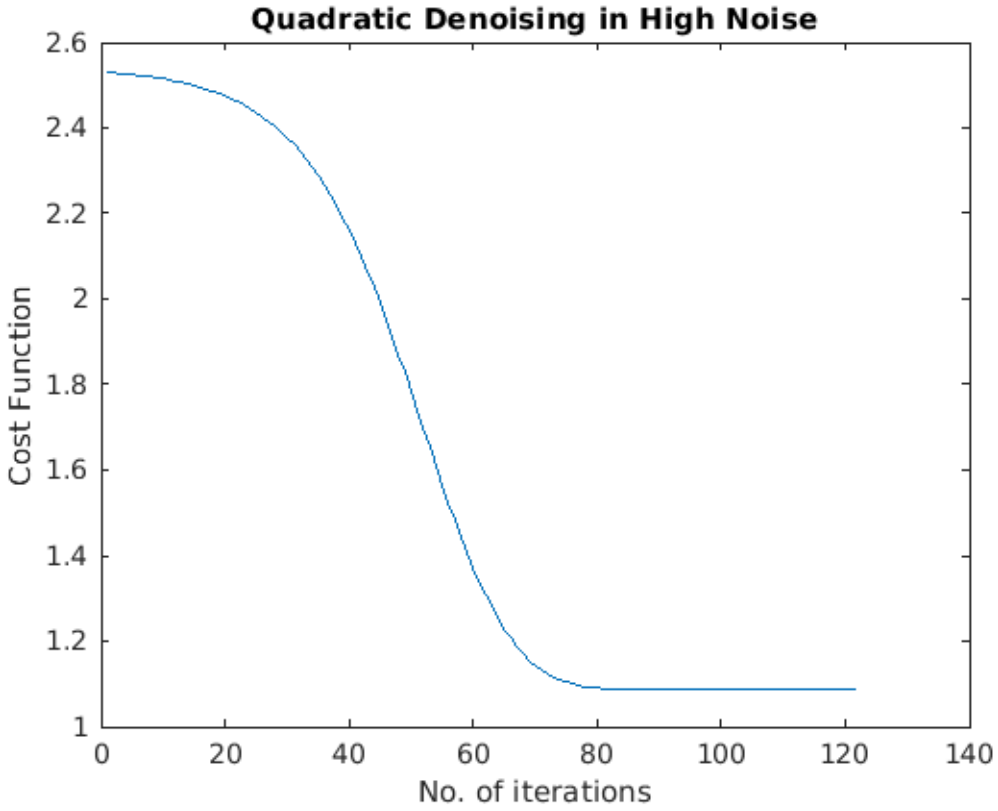
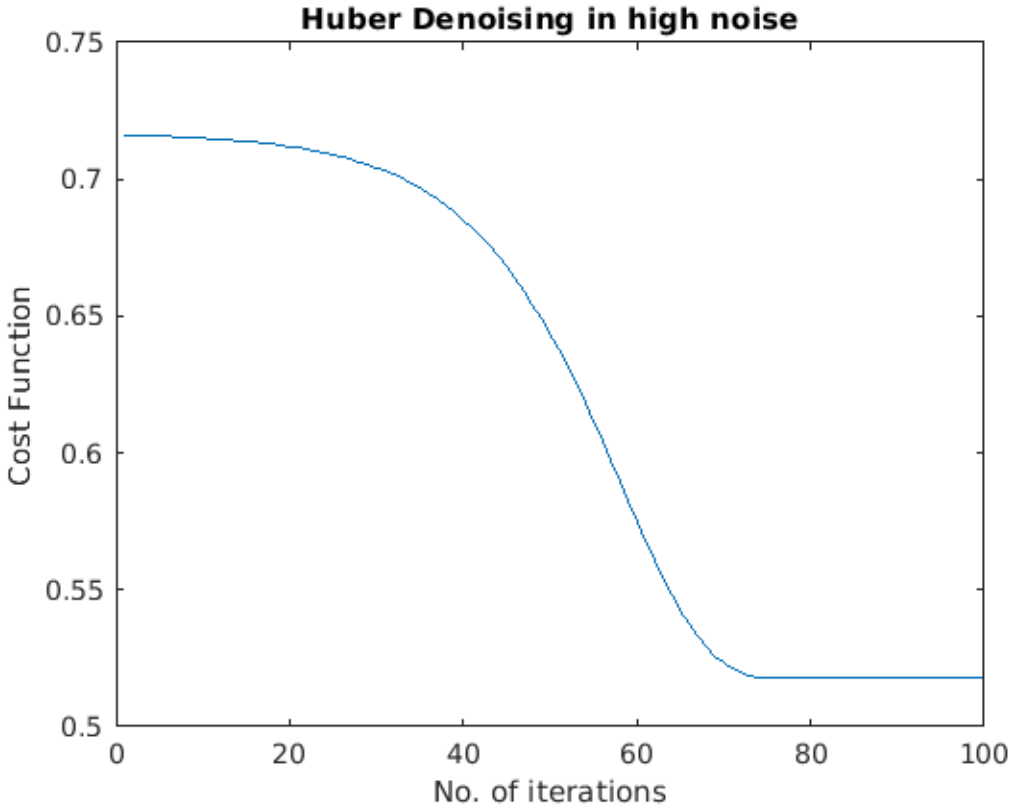
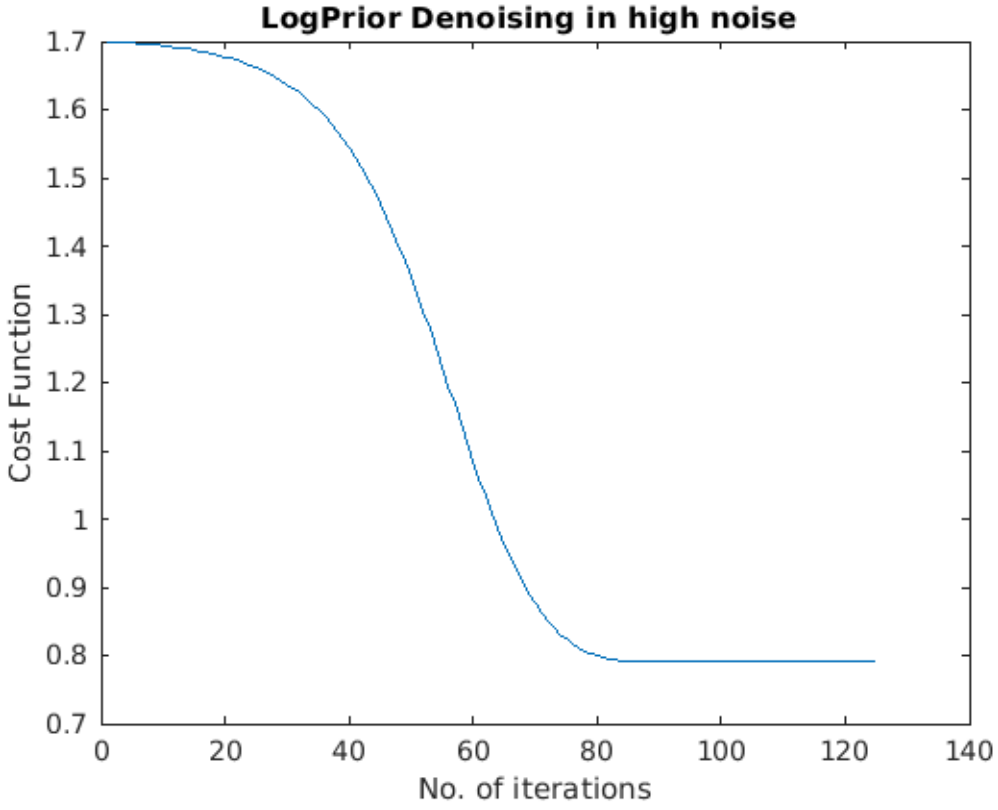
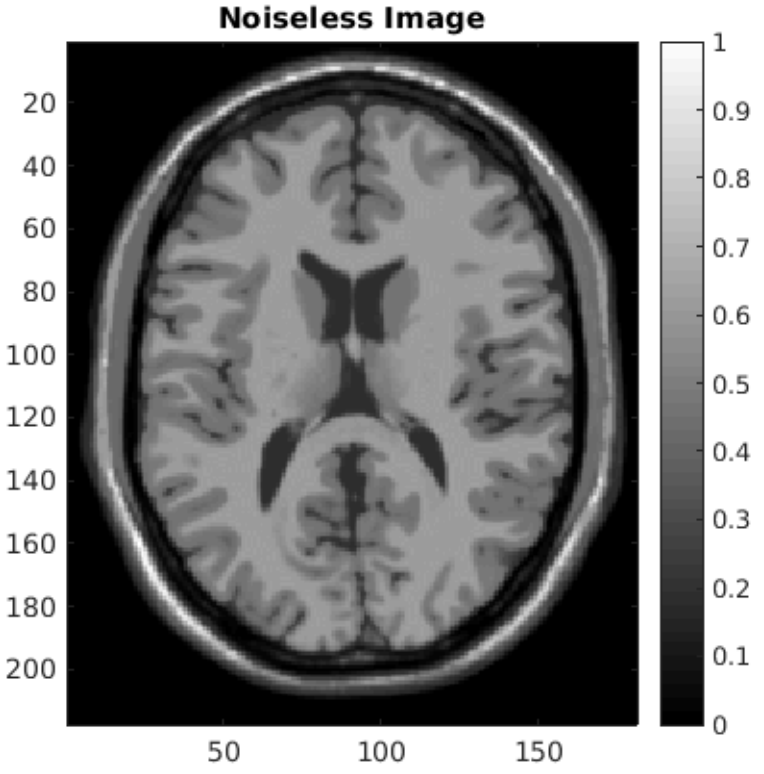
```
Denoising High noise Using Quadratic Prior
Optimal Alpha = 0.300000
RRMSE(alpha) = 0.127959
RRMSE(alpha*0.8) = 0.127077
RRMSE(alpha*1.2) = 0.130409

Denoising High noise Using Huber Prior
Optimal Alpha = 0.400000
Optimal Gamma = 0.090000
RRMSE(alpha,gamma) = 0.124974
RRMSE(alpha*0.8,gamma) = 0.128008
RRMSE(alpha*1.2,gamma) = 0.123812
RRMSE(alpha,gamma*0.8) = 0.125290
RRMSE(alpha,gamma*1.2) = 0.125045

Denoising High noise Using Disc Prior
Optimal Alpha = 0.510000
Optimal Gamma = 0.470000
RRMSE(alpha,gamma) = 0.127112
RRMSE(alpha*0.8,gamma) = 0.126269
RRMSE(alpha*1.2,gamma) = 0.131990
RRMSE(alpha,gamma*0.8) = 0.126688
RRMSE(alpha,gamma*1.2) = 0.127432

Denoising Low noise Using Quadratic Prior
Optimal Alpha = 0.070000
RRMSE(alpha) = 0.046899
RRMSE(alpha*0.8) = 0.047298
RRMSE(alpha*1.2) = 0.046739

Denoising Low noise Using Huber Prior
Optimal Alpha = 0.440000
Optimal Gamma = 0.040000
RRMSE(alpha,gamma) = 0.045851
RRMSE(alpha*0.8,gamma) = 0.044419
RRMSE(alpha*1.2,gamma) = 0.049354
RRMSE(alpha,gamma*0.8) = 0.044633
RRMSE(alpha,gamma*1.2) = 0.047168

Denoising Low noise Using Disc Prior
Optimal Alpha = 0.250000
Optimal Gamma = 0.600000
RRMSE(alpha,gamma) = 0.046724
RRMSE(alpha*0.8,gamma) = 0.046222
RRMSE(alpha*1.2,gamma) = 0.047995
RRMSE(alpha,gamma*0.8) = 0.046489
RRMSE(alpha,gamma*1.2) = 0.046903

Denoising Medium noise Using Quadratic Prior
Optimal Alpha = 0.200000
RRMSE(alpha) = 0.116242
RRMSE(alpha*0.8) = 0.116189
RRMSE(alpha*1.2) = 0.117105
```

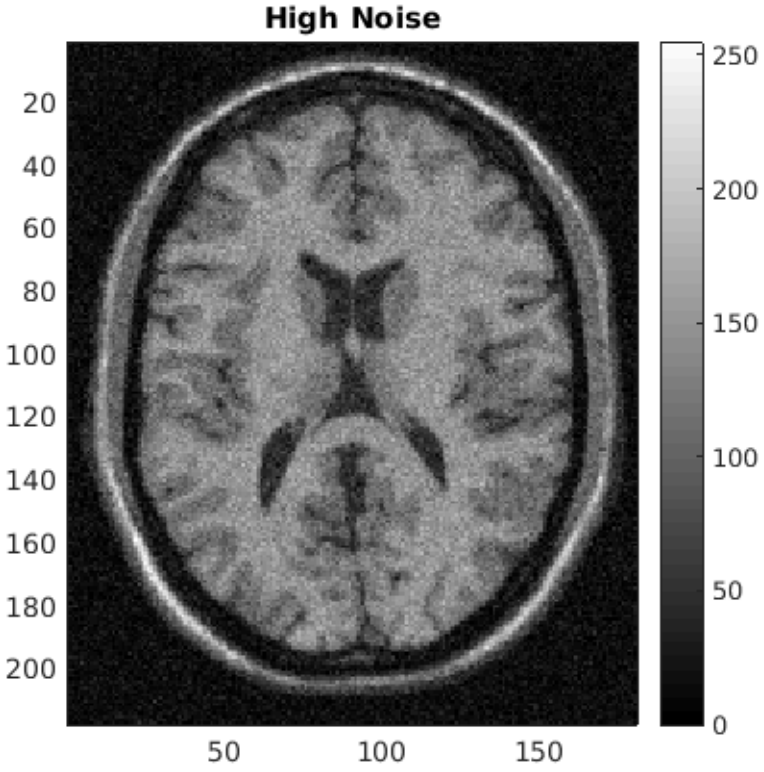
```
Denoising Medium noise Using Huber Prior
Optimal Alpha = 0.590000
Optimal Gamma = 0.040000
RRMSE(alpha,gamma) = 0.114694
RRMSE(alpha*0.8,gamma) = 0.112858
RRMSE(alpha*1.2,gamma) = 0.122527
RRMSE(alpha,gamma*0.8) = 0.113188
RRMSE(alpha,gamma*1.2) = 0.116337

Denoising Medium noise Using Disc Prior
Optimal Alpha = 0.180000
Optimal Gamma = 0.470000
RRMSE(alpha,gamma) = 0.046724
RRMSE(alpha*0.8,gamma) = 0.046222
RRMSE(alpha*1.2,gamma) = 0.047995
RRMSE(alpha,gamma*0.8) = 0.046489
RRMSE(alpha,gamma*1.2) = 0.046903
```

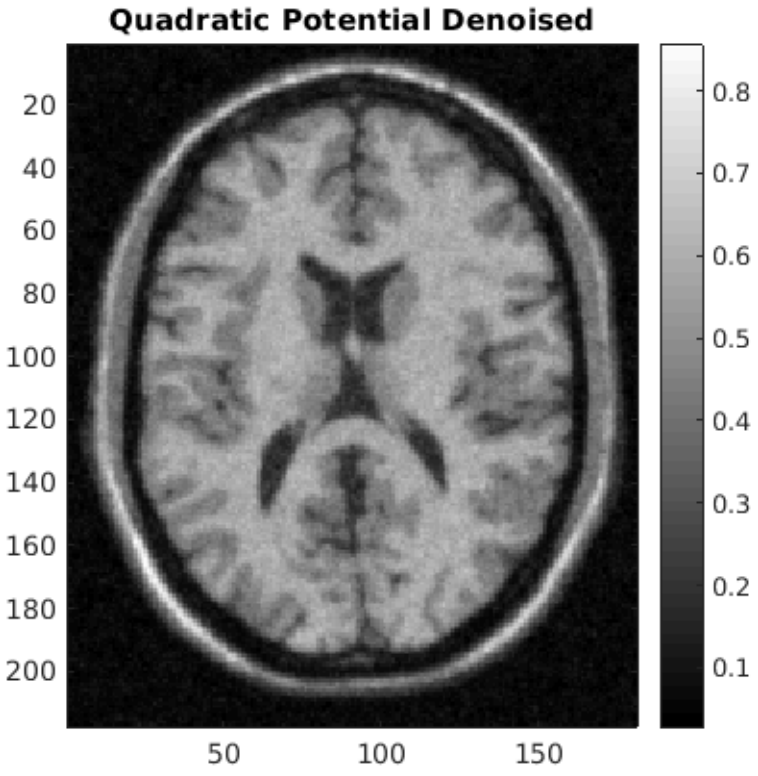


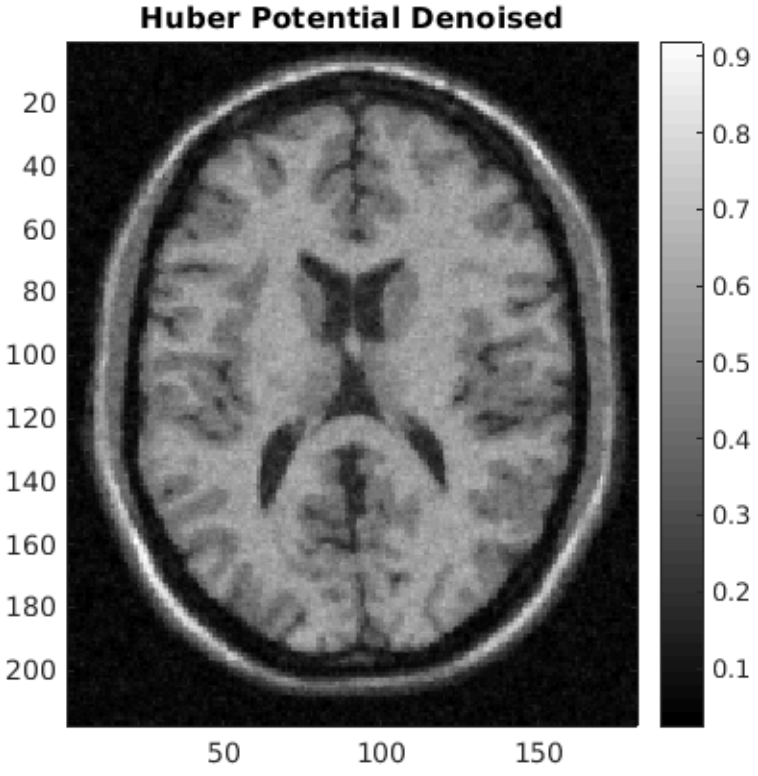

Huber Denoising in high noise
Cost Function
No. of iterations
0.75
0.7
0.65
0.6
0.55
0.5
0
20
40
60
80
100

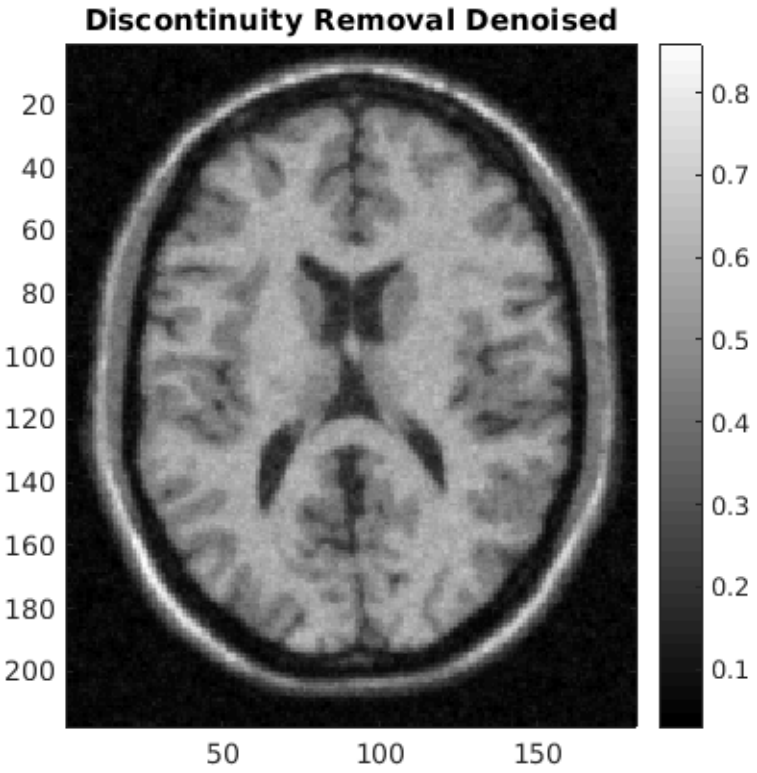

LogPrior Denoising in high noise
Cost Function
No. of iterations
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0
20
40
60
80
100
120
140

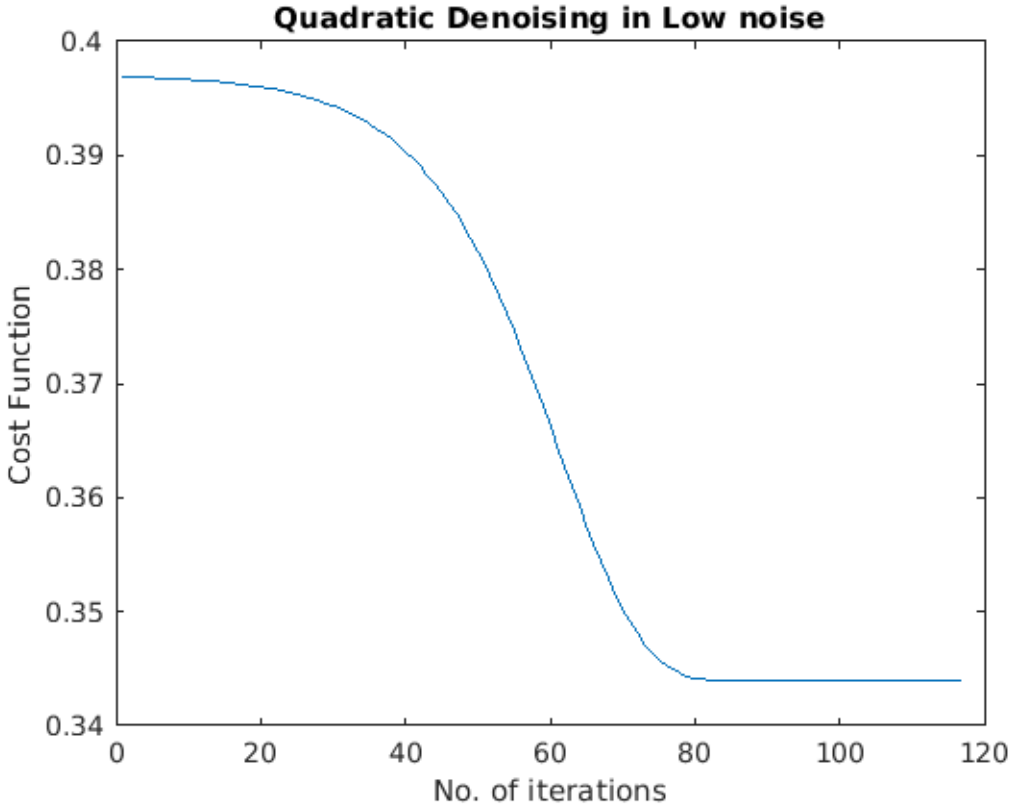

Noiseless Image
20
40
60
80
100
120
140
160
180
200
50
100
150
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

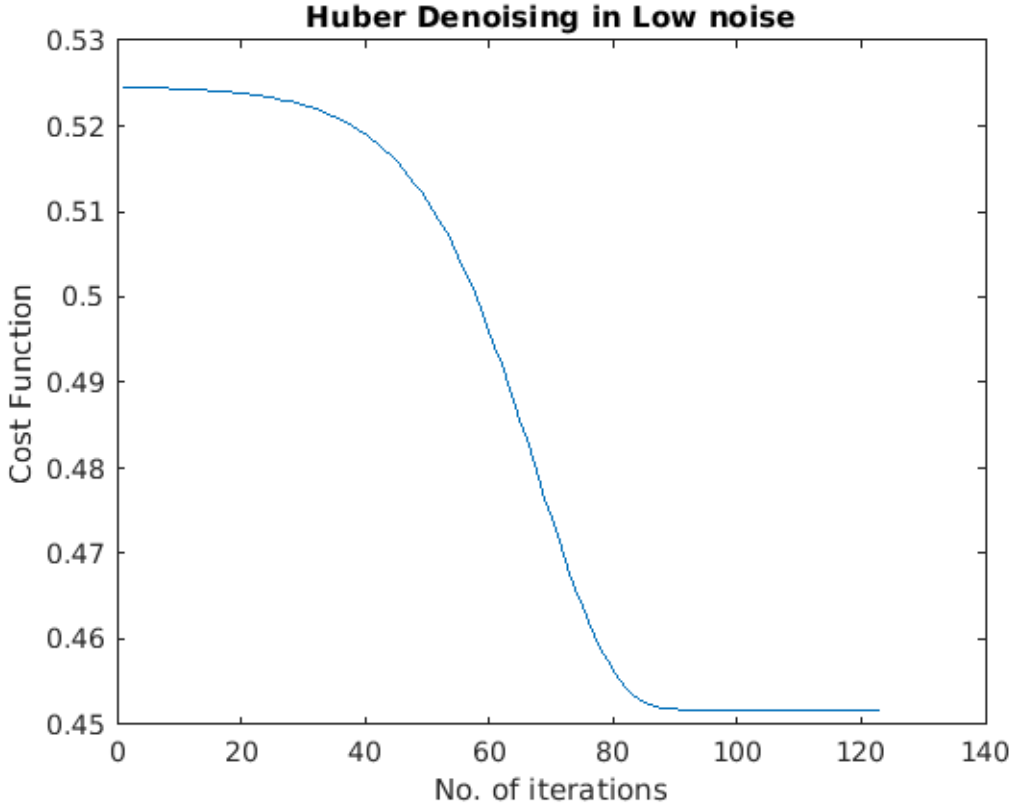

High Noise
20
40
60
80
100
120
140
160
180
200
50
100
150
250
200
150
100
50
0

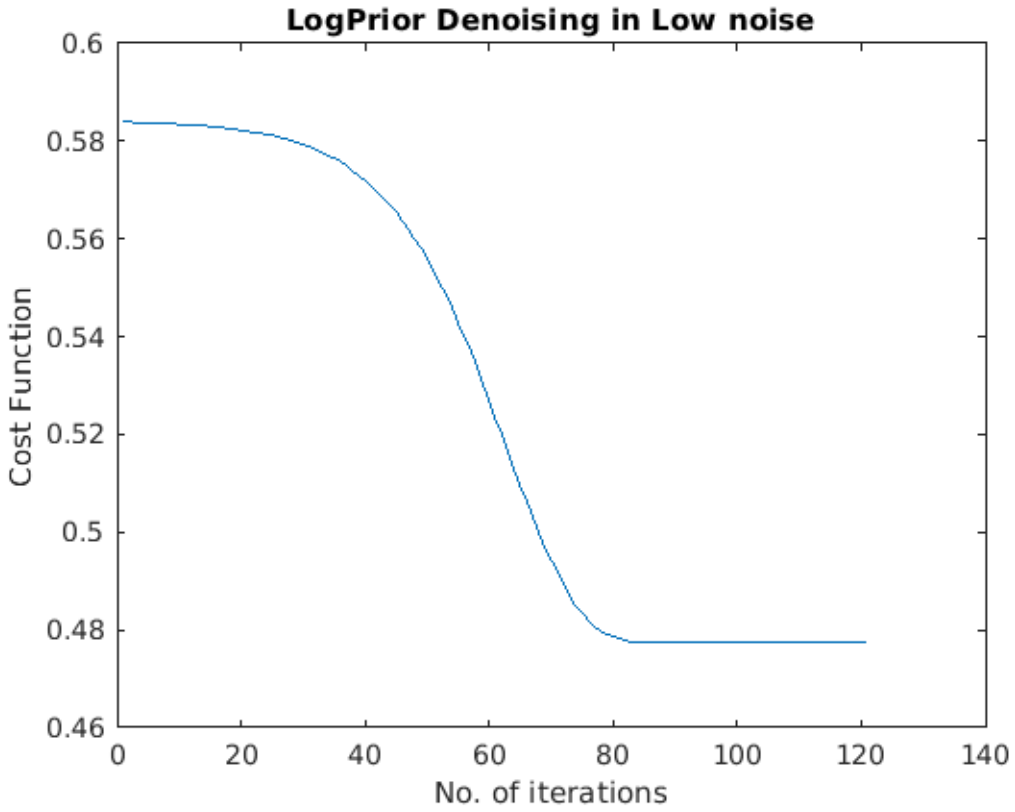

Quadratic Potential Denoised
20
40
60
80
100
120
140
160
180
200
50
100
150
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

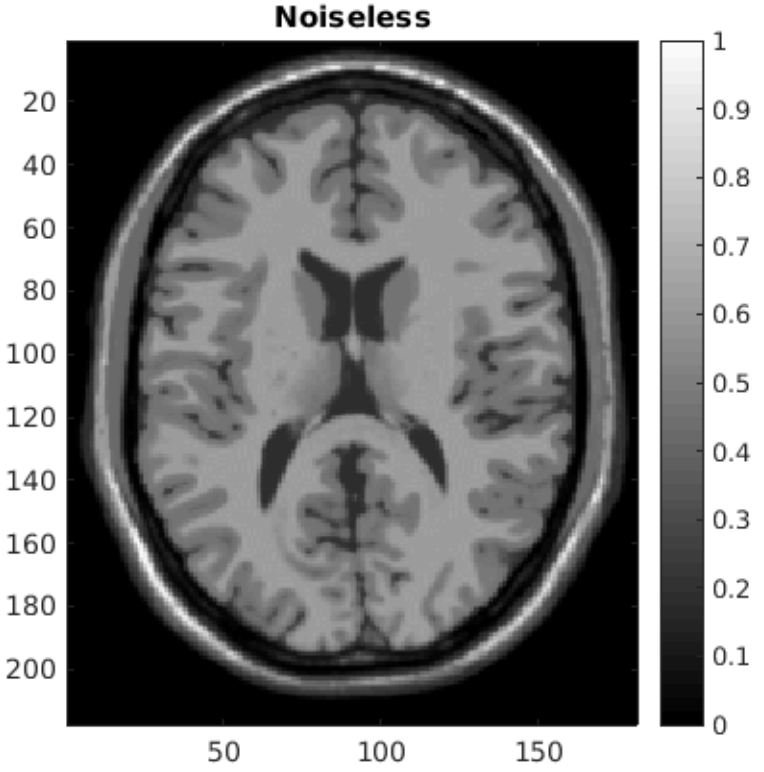

Huber Potential Denoised
20
40
60
80
100
120
140
160
180
200
50
100
150
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

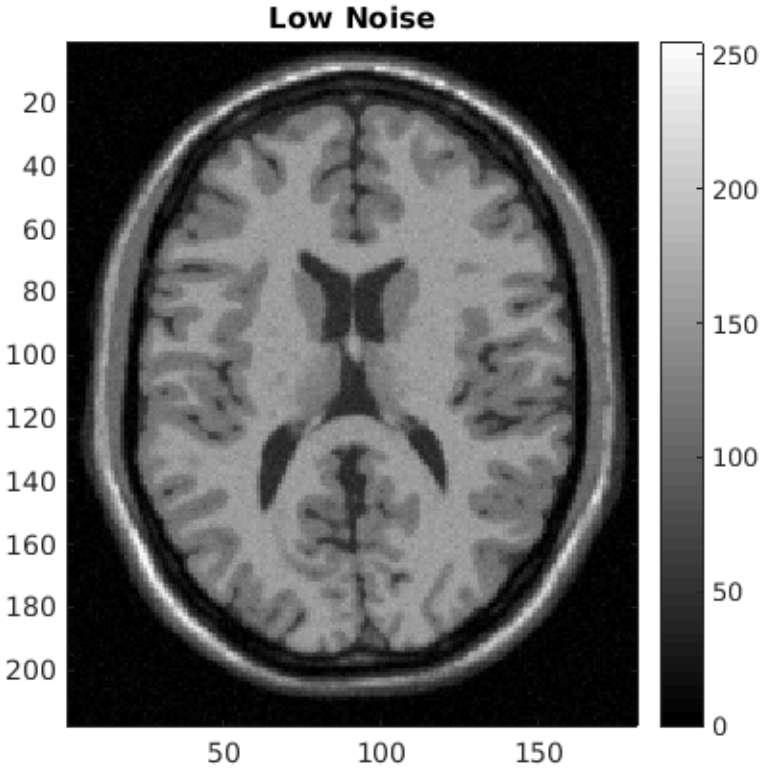

Discontinuity Removal Denoised
20
40
60
80
100
120
140
160
180
200
50
100
150
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

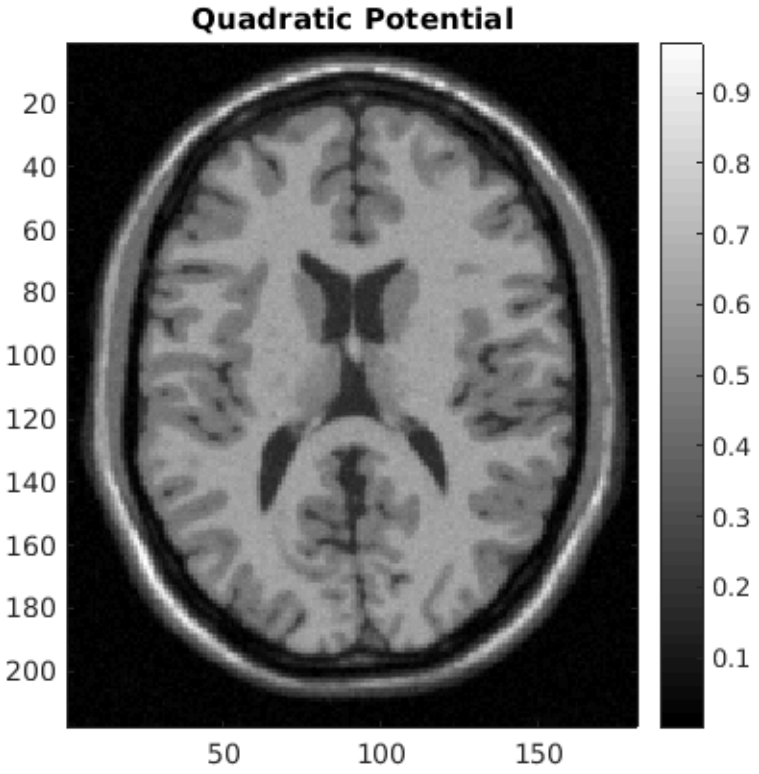

Quadratic Denoising in Low noise
Cost Function
No. of iterations
0.4
0.39
0.38
0.37
0.36
0.35
0.34
0
20
40
60
80
100
120

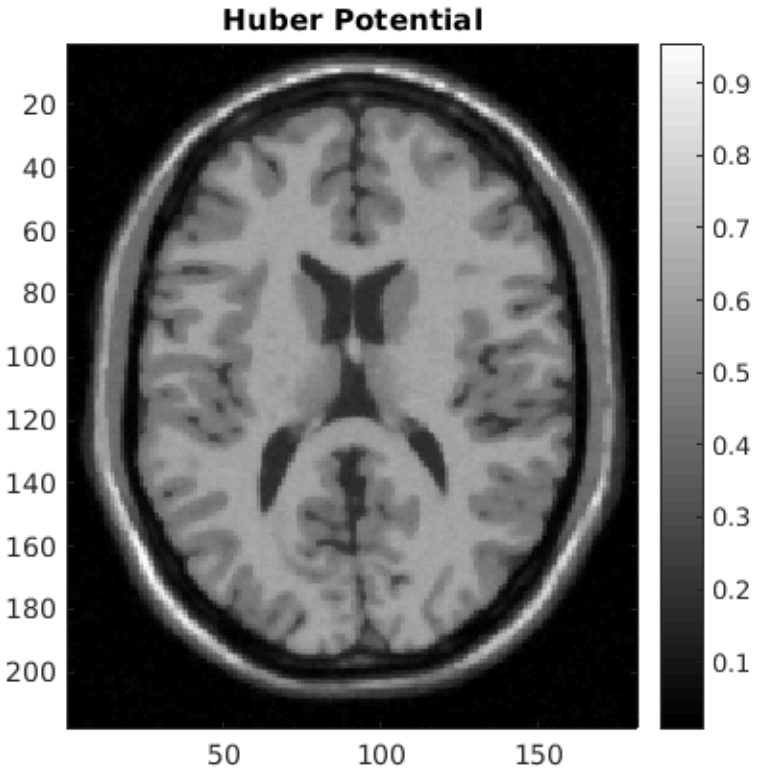

Huber Denoising in Low noise
Cost Function
No. of iterations
0.53
0.52
0.51
0.5
0.49
0.48
0.47
0.46
0.45
0
20
40
60
80
100
120
140

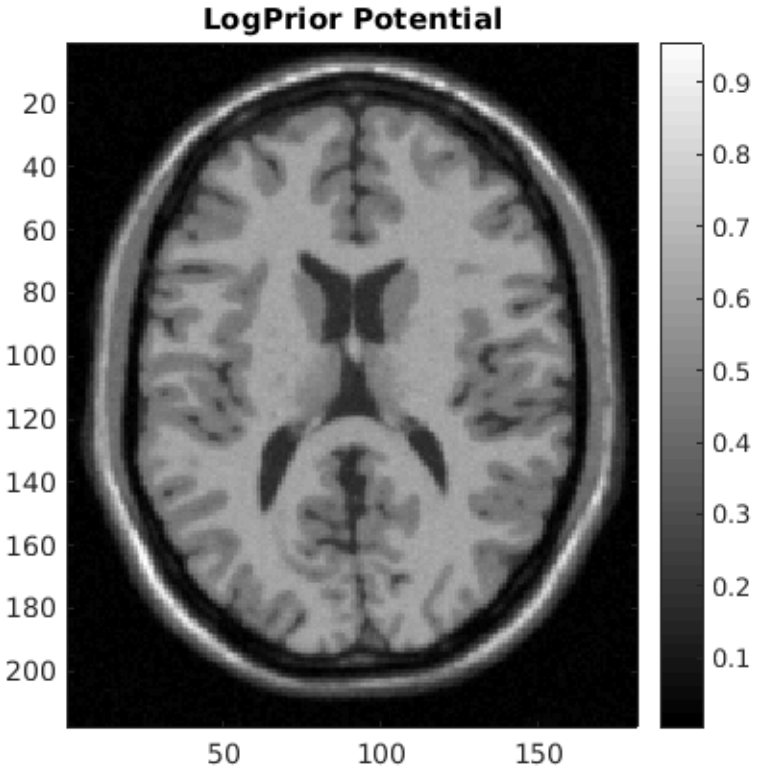

LogPrior Denoising in Low noise
Cost Function
No. of iterations
0.6
0.58
0.56
0.54
0.52
0.5
0.48
0.46
0
20
40
60
80
100
120
140

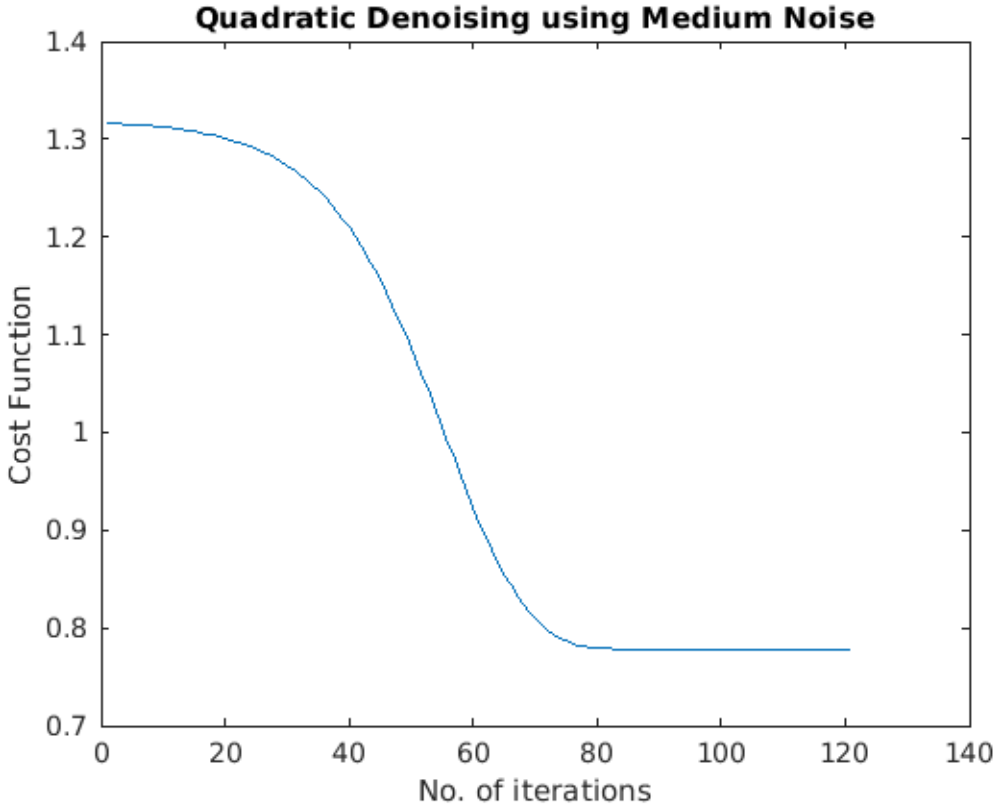

Noiseless
20
40
60
80
100
120
140
160
180
200
50
100
150
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

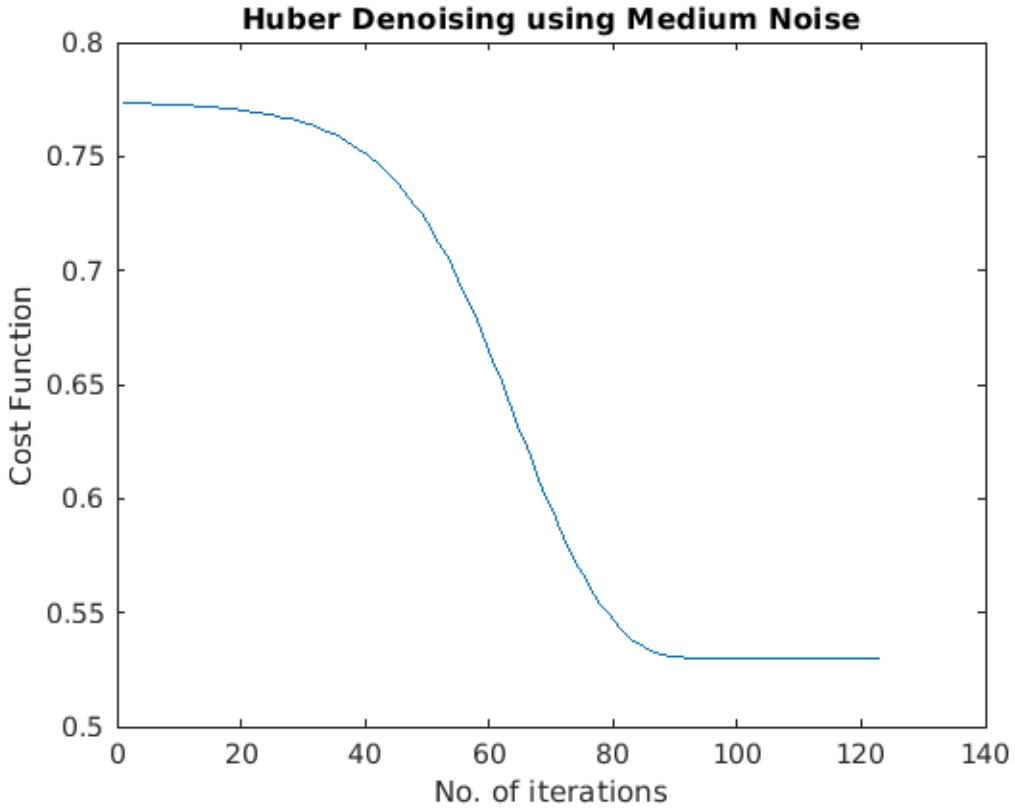

Low Noise
20
40
60
80
100
120
140
160
180
200
50
100
150
250
200
150
100
50
0

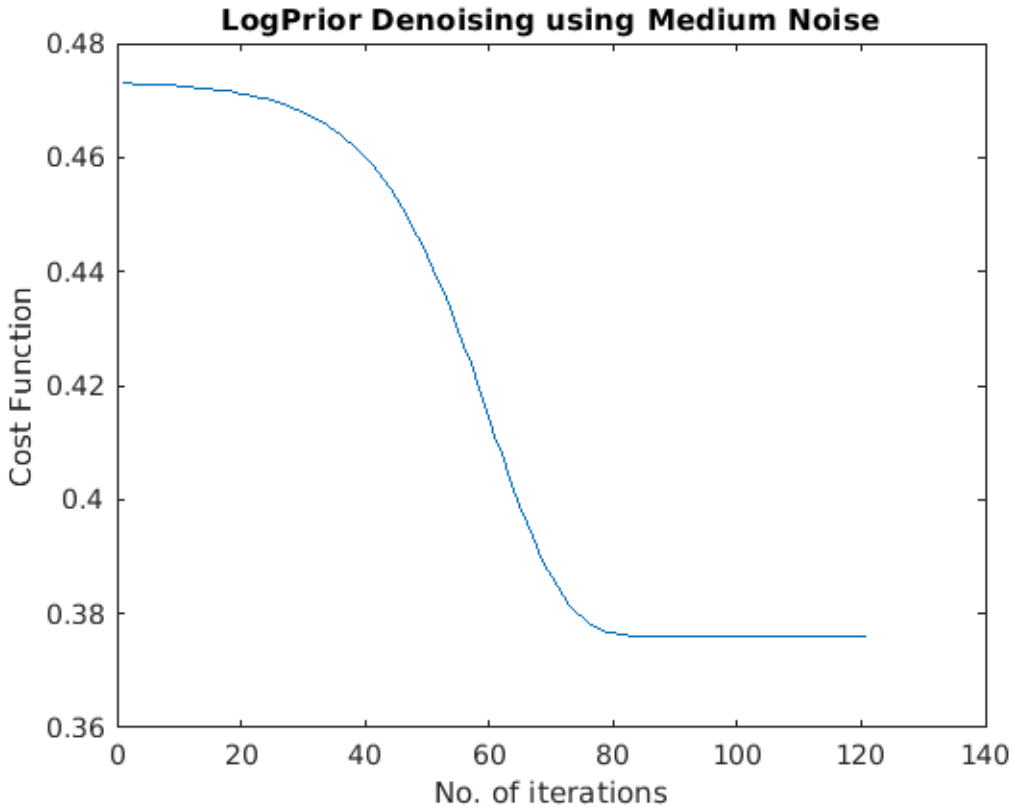

Quadratic Potential
20
40
60
80
100
120
140
160
180
200
50
100
150
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

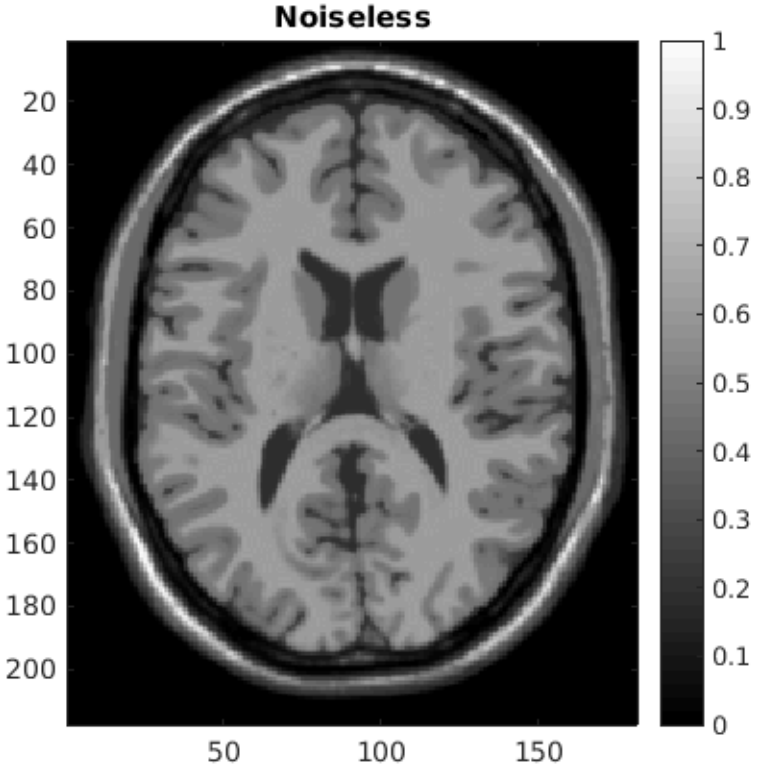

Huber Potential
20
40
60
80
100
120
140
160
180
200
50
100
150
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

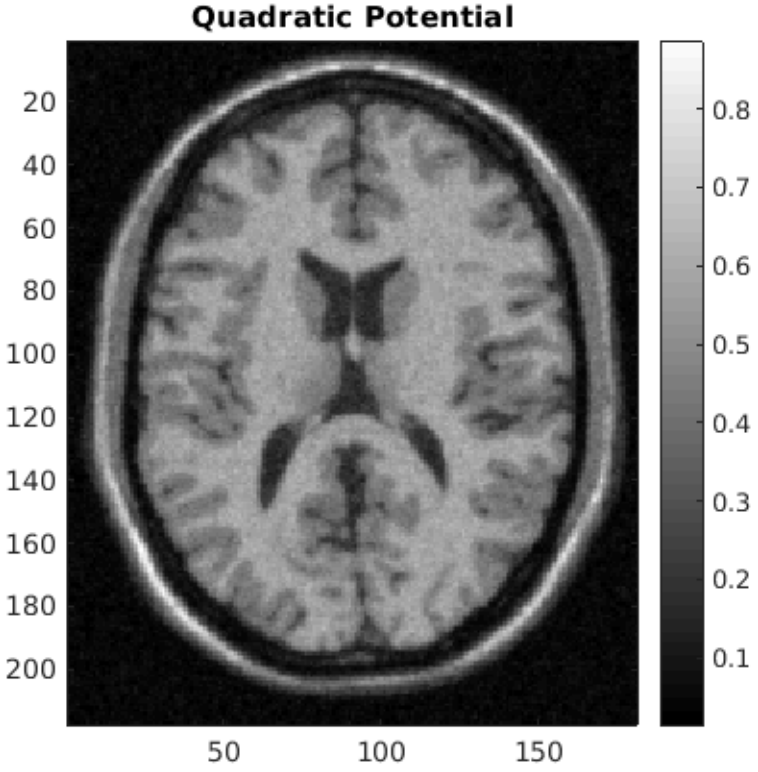

LogPrior Potential
20
40
60
80
100
120
140
160
180
200
50
100
150
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

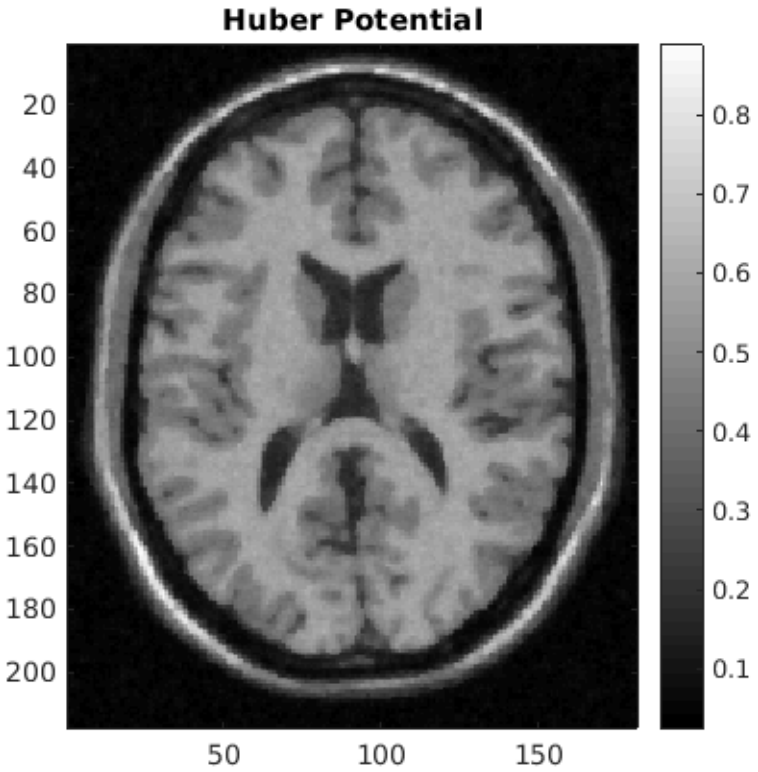

Quadratic Denoising using Medium Noise
Cost Function
No. of iterations
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0
20
40
60
80
100
120
140

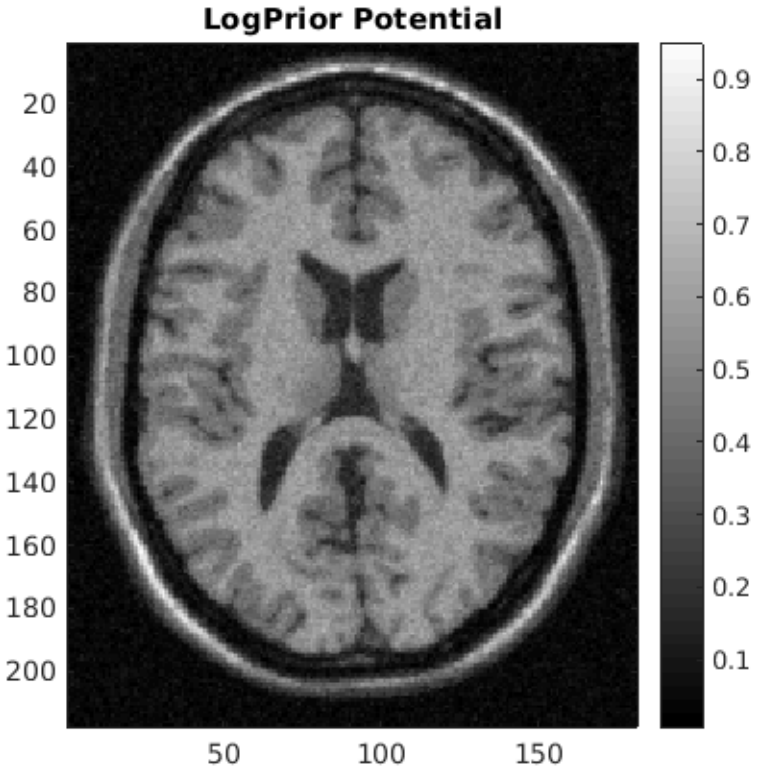

Huber Denoising using Medium Noise
Cost Function
No. of iterations
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
20
40
60
80
100
120
140


LogPrior Denoising using Medium Noise
Cost Function
No. of iterations
0.48
0.46
0.44
0.42
0.4
0.38
0.36
0
20
40
60
80
100
120
140


Noiseless
20
40
60
80
100
120
140
160
180
200
50
100
150
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Medium Noise
20
40
60
80
100
120
140
160
180
200
50
100
150
250
200
150
100
50
0


Quadratic Potential
20
40
60
80
100
120
140
160
180
200
50
100
150
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


Huber Potential
20
40
60
80
100
120
140
160
180
200
50
100
150
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1



Published with MATLAB® R2019b