Table of Contents

PART A: RMSSE between Noiseless and Noisy image 1
PART B: 1. Quadratic prior 1
PART B: 2. Huber prior 2
PART B: 3. Discontinuity adaptive prior 3
PART C: Image plotting 4
PART D: Objective function vs iteration 8

PART A: RMSSE between Noiseless and Noisy image

```
load('../data/assignmentImageDenoisingPhantom.mat');
disp('RRMSE between Noiseless and noisy image');
disp(rrmse(imageNoiseless,imageNoisy));
```

```
RRMSE between Noiseless and noisy image
    0.4584
```

PART B: 1. Quadratic prior

```
y=imageNoisy;
x=y;
alpha=1;
step=0.001;
prev_rrmse=100;
obj_fun_quadratic=[];
while(abs(rrmse(imageNoiseless,x)-prev_rrmse)>0.000001)

    prev_rrmse=rrmse(imageNoiseless,x);
    grad_prior=2*((x-circshift(x,1,1))+(x-circshift(x,-1,1))+(x-
circshift(x,1,2))+(x-circshift(x,-1,2)));
    grad=2*(1-alpha)*(x-y)+alpha*grad_prior;
    x=x-step*grad;
    %disp(rrmse(imageNoiseless,x));
    obj_fun_quadratic=[obj_fun_quadratic prev_rrmse];

end
denoised_quadratic_image=x;
disp('RRMSE for alpha=1:');
disp(rrmse(imageNoiseless,x));
disp('Since alpha=1, 1.2*1 will be outside the limits');
disp('Hence RRMSE(1.2*alpha*)=');
disp(rrmse(imageNoiseless,x));
```

```
RRMSE for alpha=1:
    0.2279
```

```
Since alpha=1, 1.2*1 will be outside the limits
Hence RRMSE(1.2*alpha*)=
    0.2279
```

PART B: 2. Huber prior

```
alpha=1;
gamma=0.007;
disp('alpha_optimum:')
disp(alpha)
disp('gamma_optimum:')
disp(gamma)
x=imageNoiseless;
y=imageNoisy;

[denoised_huber_image,rrmse,obj_fun_huber]=huber(x,y,alpha,gamma);
disp('For alpha=1 and gamma=0.007, RRMSE(alpha*,gamma*)=');
disp(rrmse);

disp('Since alpha=1, 1.2*1 will be outside the limits');
disp('Hence RRMSE(1.2*alpha*,gamma*)=');
disp(rrmse);

disp('For alpha=0.8*1=0.8 and gamma=0.007,
 RRMSE(0.8*alpha*,gamma*)=0.2776');

disp('For alpha=1 and gamma=1.2*0.007=0.0084,
 RRMSE(alpha*,1.2*gamma*)=0.0705');

disp('For alpha=1 and gamma=0.8*0.007=0.0056,
 RRMSE(alpha,0.8*gamma*)=0.0651');

disp('Thus values of alpha and gamma are optimum at 1 and 0.007
 respectively');
```

```
alpha_optimum:
     1

gamma_optimum:
    0.0070

For alpha=1 and gamma=0.007, RRMSE(alpha*,gamma*)=
    0.0688

Since alpha=1, 1.2*1 will be outside the limits
Hence RRMSE(1.2*alpha*,gamma*)=
    0.0688

For alpha=0.8*1=0.8 and gamma=0.007, RRMSE(0.8*alpha*,gamma*)=0.2776
For alpha=1 and gamma=1.2*0.007=0.0084,
 RRMSE(alpha*,1.2*gamma*)=0.0705
```

```
For alpha=1 and gamma=0.8*0.007=0.0056, RRMSE(alpha,0.8*gamma*)=0.0651
Thus values of alpha and gamma are optimum at 1 and 0.007 respectively
```

PART B: 3. Discontinuity adaptive prior

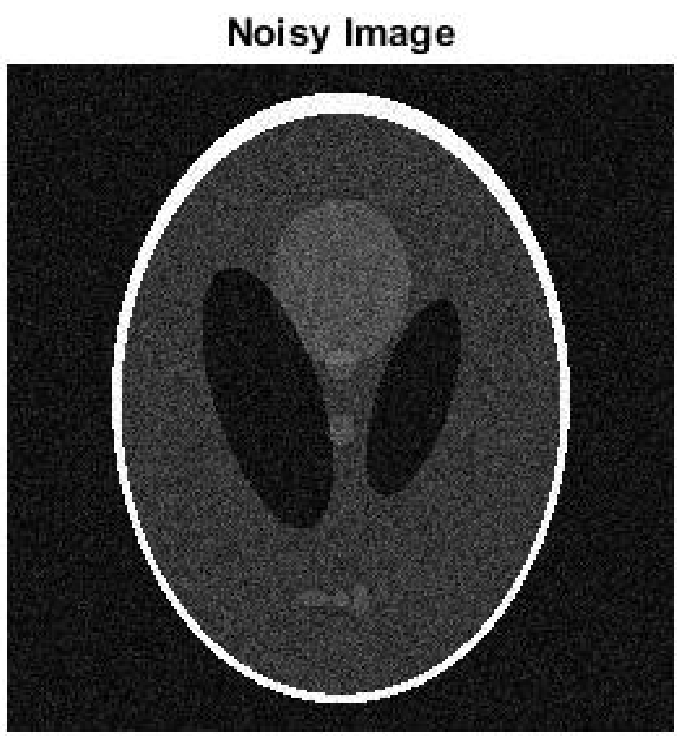
```
alpha=1;
gamma=0.0001;
disp('alpha:')
disp(alpha)
disp('gamma:')
disp(gamma)
x=imageNoiseless;
y=imageNoisy;

[denoised_discontinuity_adaptive_image,rrmse,obj_fun_discontinuity_adaptive]=disco

disp('For alpha=1 and gamma=0.0001, RRMSE(alpha*,gamma*)=');
disp(rrmse);

disp('Since alpha=1, 1.2*1 will be outside the limits');
disp('Hence RRMSE(1.2*alpha*,gamma*)=');
disp(rrmse);

disp('For alpha=0.8*1=0.8 and gamma=0.0001,
 RRMSE(0.8*alpha*,gamma*)=0.4555');

disp('For alpha=1 and gamma=1.2*0.0001=0.0012,
 RRMSE(alpha*,1.2*gamma*)=0.0545');

disp('For alpha=1 and gamma=0.8*0.0001=0.00008,
 RRMSE(alpha,0.8*gamma*)=0.0542');

disp('Thus values of alpha and gamma are optimum at 1 and 0.0001
 respectively');
```

```
alpha:
     1

gamma:
   1.0000e-04

For alpha=1 and gamma=0.0001, RRMSE(alpha*,gamma*)=
    0.0542

Since alpha=1, 1.2*1 will be outside the limits
Hence RRMSE(1.2*alpha*,gamma*)=
    0.0542

For alpha=0.8*1=0.8 and gamma=0.0001, RRMSE(0.8*alpha*,gamma*)=0.4555
For alpha=1 and gamma=1.2*0.0001=0.0012,
 RRMSE(alpha*,1.2*gamma*)=0.0545
For alpha=1 and gamma=0.8*0.0001=0.00008,
 RRMSE(alpha,0.8*gamma*)=0.0542
```

Thus values of alpha and gamma are optimum at 1 and 0.0001 respectively

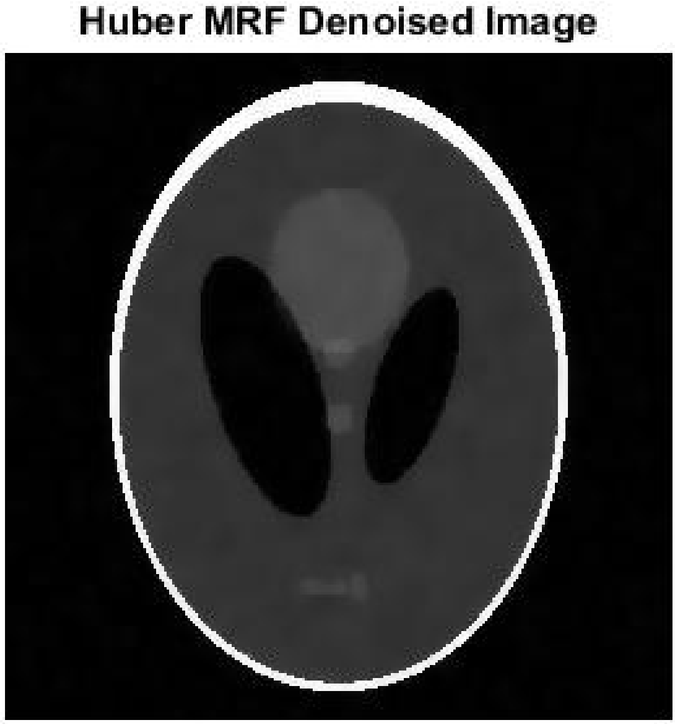
PART C: Image plotting

```
figure; imshow(abs(imageNoisy)); title('Noisy Image');

figure; imshow(imageNoiseless); title('Noiseless Image');

figure;imshow(abs(denoised_quadratic_image)); title('Quadratic MRF
 Denoised Image');

figure;imshow(abs(denoised_huber_image)); title('Huber MRF Denoised
 Image');

figure;imshow(abs(denoised_discontinuity_adaptive_image));
 title('Discontinuity Adaptive MRF Denoised Image');
```



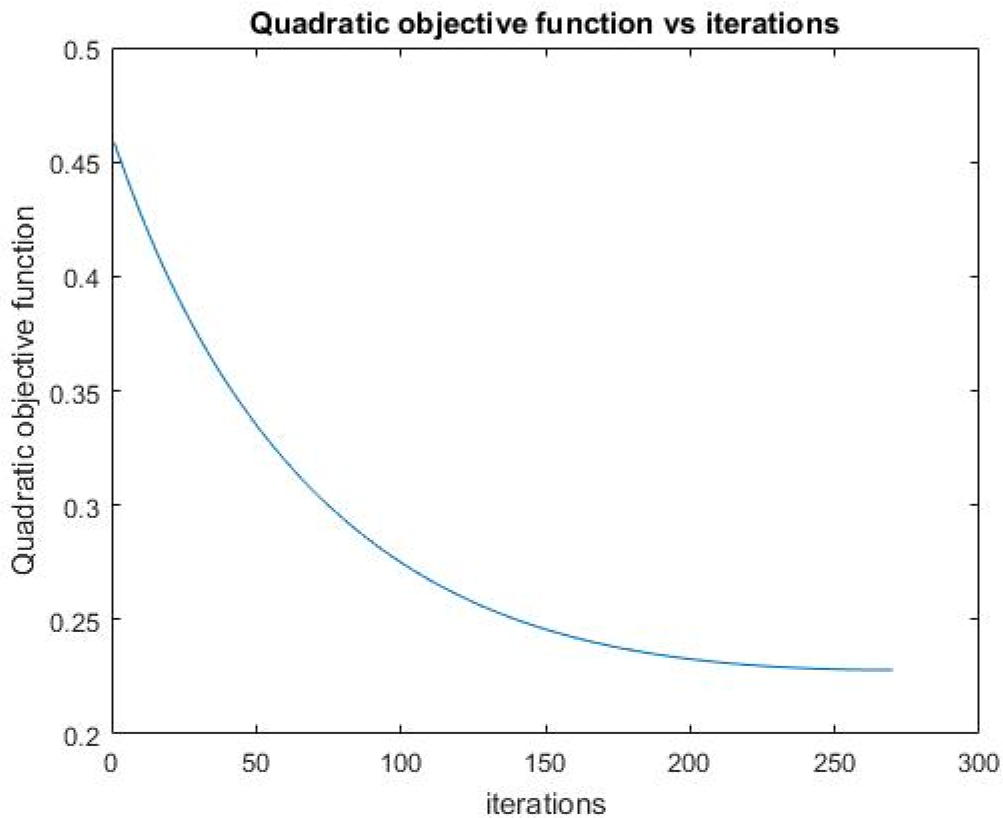
Noiseless Image

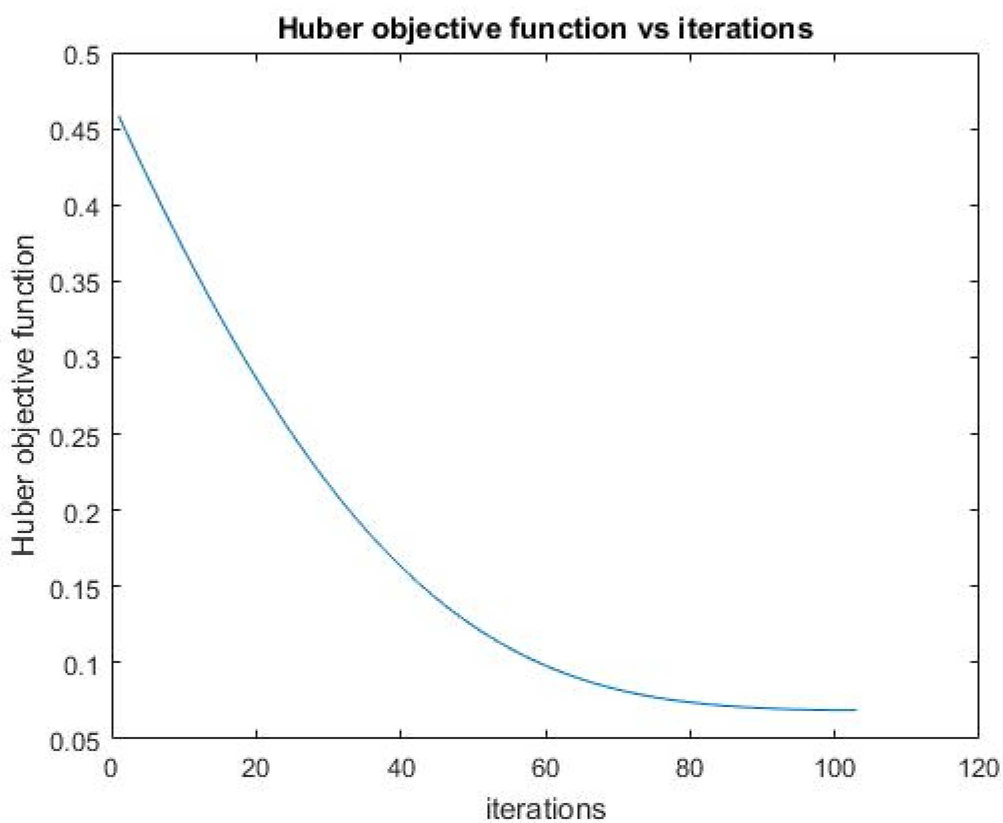
Quadratic MRF Denoised Image

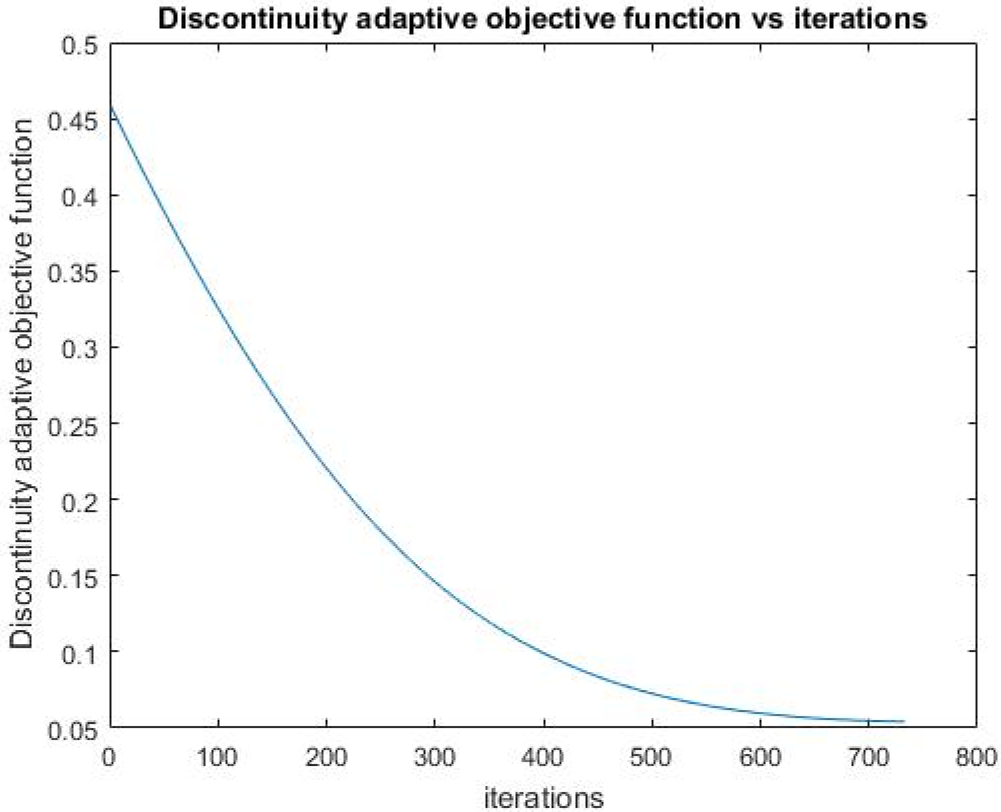

Huber MRF Denoised Image

PART D: Objective function vs iteration

```
figure; plot(obj_fun_quadratic); title('Quadratic objective function
 vs iterations'); xlabel('iterations'); ylabel('Quadratic objective
 function');

figure; plot(obj_fun_huber); title('Huber objective function
 vs iterations'); xlabel('iterations'); ylabel('Huber objective
 function');

figure; plot(obj_fun_discontinuity_adaptive); title('Discontinuity
 adaptive objective function vs iterations'); xlabel('iterations');
 ylabel('Discontinuity adaptive objective function');
```


Quadratic objective function vs iterations
Quadratic objective function
iterations
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
50
100
150
200
250
300


Huber objective function vs iterations
Huber objective function
iterations
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
20
40
60
80
100
120



Published with MATLAB® R2015b