```
%%
A = double(imread("histology_noiseless.png"));
Y1 = double(imread("histology_noisy.png"));

g1 = @(x,gamma) ((x.^2)/gamma);
g2 = @(x,gamma) (0.5*(x.^2).*(abs(x)<=gamma) + (gamma*abs(x)-0.5*gamma^2).*(abs(x)>gamma));
g3 = @(x,gamma) (gamma*abs(x) - (gamma^2)*log(1+(abs(x)/gamma)));

data=zeros([3,3,2]);
data(1,:,:)=([[1,0.74];[1,0.98];[1.1,0.98]]);
data(2,:,:)=([[1.1,0.64];[0.8,0.98];[0.9,0.98]]);
data(3,:,:)=([[1,0.78];[1,0.98];[1.2,0.98]]);
name=["R","G","B"];
dRGB = [];
 for i=1:3
 %     Y = double(imread("mri_image_noise_level_"+name(i)+".png"));
    Y = Y1(:,:,i);
    disp("Optimizing color channel "+name(i)+" of the Image");
    disp("RRMSE b/w noisy and noiseless "+name(i)+" channels = " + RRMSE(Y,A(:,:,i)));
    X=2*Y;
    for j=1:3
        partB(X,Y,data,i,j,g1,g2,g3,A(:,:,i));
    end
 end

%%

 for i=1:3
%     Y = double(imread("mri_image_noise_level_"+name(i)+".png"));
    Y = Y1(:,:,i);
    disp("Denoising color channel "+name(i)+" of the Image");
    disp(" ");
    X=2*Y;
   [f,Cg1,Cg2,Cg3] = partCnD(X,Y,data,i,g1,g2,g3,A(:,:,i));
   dRGB = cat(3,dRGB,Cg1,Cg2,Cg3);
%     save("results_img"+i+".mat",f);
 end

 img_g1 = uint8(cat(3,dRGB(:,:,1),dRGB(:,:,4),dRGB(:,:,7)));
 img_g2 = uint8(cat(3,dRGB(:,:,2),dRGB(:,:,5),dRGB(:,:,8)));
 img_g3 = uint8(cat(3,dRGB(:,:,3),dRGB(:,:,6),dRGB(:,:,9)));

 OK(3) = figure(25);
 imshow(uint8(A));
 title("Noiseless RGB");

 OK(3) = figure(26);
 imshow(uint8(Y1));
 title("Noisy RGB");
```

```
OK(3) = figure(27);
imshow(img_g1);
title("RGB Image denoised using g1 prior");

OK(4) = figure(28);
imshow(img_g2);
title("RGB Image denoised using g2 prior");

OK(5) = figure(29);
imshow(img_g3);
title("RGB Image denoised using g3 prior");

%  save("results_RGB.mat","OK");

%%

% X1=2*Y1;
%
%  R1 = X1(:,:,1);
%  G1 = X1(:,:,2);
%  B1 = X1(:,:,3);
%
%  R2 = Y1(:,:,1);
%  G2 = Y1(:,:,2);
%  B2 = Y1(:,:,3);
% code to find optimal gamma given alpha
% N=30;
% obj_vec=zeros([N,1]);
% error_vec=zeros([N,1]);
% for i = 1:N
%     i
%     gamma = 0.1*i;
%     [X_opt, obj] = denoising(R1, R2, g3, gamma, 0.1, 0.82);
%     error = RRMSE(X_opt,A(:,:,1))
%     obj_vec(i)=obj;
%     error_vec(i)=error;
% end


% Code to find the optimal alpha
% N=50;
% obj_vec=zeros([N,1]);
% error_vec=zeros([N,1]);
% for i = 1:N
%     i
%     alpha = (i-1)/N;
%     [X_opt, obj] = denoising(R1, R2, adaptive, 1, 0.1, alpha);
%     obj
%     error = RRMSE(X_opt,A(:,:,1))
%     obj_vec(i)=obj;
```

```
%     error_vec(i)=error;
% end
```

Results for RRMSE (Part B):

Optimizing color channel R of the Image

RRMSE b/w noisy and noiseless R channels = 0.18484

Optimal Values using g1 prior:

alpha = 0.74

gamma = 1

RRMSE(alpha,gamma) = 0.048927

RRMSE(0.8*alpha,gamma) = 0.053497

RRMSE(1.2*alpha,gamma) = 0.057554

Optimal Values using g2 prior:

alpha = 0.98

gamma = 1

RRMSE(alpha,gamma) = 0.049012

RRMSE(0.8*alpha,gamma) = 0.16301

RRMSE(1.2*alpha,gamma) = 0.067239

RRMSE(alpha,0.8*gamma) = 0.050588

RRMSE(alpha,1.2*gamma) = 0.050689

Optimal Values using g3 prior:

alpha = 0.98

gamma = 1.1

RRMSE(alpha,gamma) = 0.047537

RRMSE(0.8*alpha,gamma) = 0.16164

RRMSE(1.2*alpha,gamma) = 0.061013

RRMSE(alpha,0.8*gamma) = 0.049526

RRMSE(alpha,1.2*gamma) = 0.048173

Optimizing color channel G of the Image

RRMSE b/w noisy and noiseless G channels = 0.22491

Optimal Values using g1 prior:

alpha = 0.64

gamma = 1.1

RRMSE(alpha,gamma) = 0.072818

RRMSE(0.8*alpha,gamma) = 0.079635

RRMSE(1.2*alpha,gamma) = 0.08015

Optimal Values using g2 prior:

alpha = 0.98

gamma = 0.8

RRMSE(alpha,gamma) = 0.075009

RRMSE(0.8*alpha,gamma) = 0.20031

RRMSE(1.2*alpha,gamma) = 0.098711

RRMSE(alpha,0.8*gamma) = 0.077181

RRMSE(alpha,1.2*gamma) = 0.07733

Optimal Values using g3 prior:

alpha = 0.98

gamma = 0.9

RRMSE(alpha,gamma) = 0.072519

RRMSE(0.8*alpha,gamma) = 0.19817

RRMSE(1.2*alpha,gamma) = 0.09472

RRMSE(alpha,0.8*gamma) = 0.074204

RRMSE(alpha,1.2*gamma) = 0.074274

Optimizing color channel B of the Image

RRMSE b/w noisy and noiseless B channels = 0.1914

Optimal Values using g1 prior:

alpha = 0.78

gamma = 1

RRMSE(alpha,gamma) = 0.045687

RRMSE(0.8*alpha,gamma) = 0.051095

RRMSE(1.2*alpha,gamma) = 0.057351

Optimal Values using g2 prior:

alpha = 0.98

gamma = 1

RRMSE(alpha,gamma) = 0.045049

RRMSE(0.8*alpha,gamma) = 0.16852

RRMSE(1.2*alpha,gamma) = 0.060571

RRMSE(alpha,0.8*gamma) = 0.047452

RRMSE(alpha,1.2*gamma) = 0.046557

Optimal Values using g3 prior:

alpha = 0.98

gamma = 1.2

RRMSE(alpha,gamma) = 0.044048

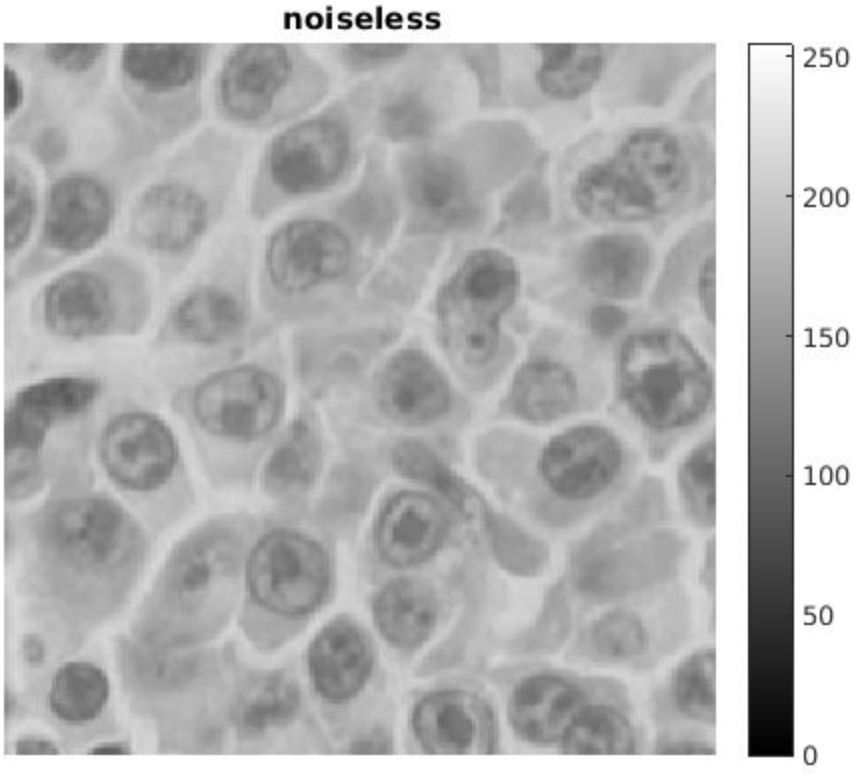
RRMSE(0.8*alpha,gamma) = 0.16505

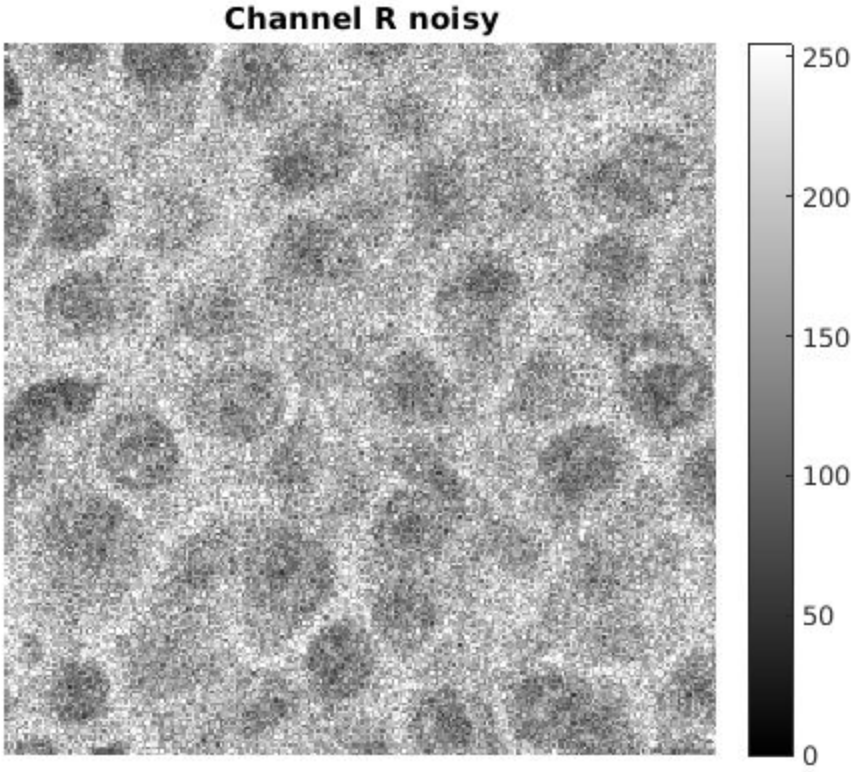
RRMSE(1.2*alpha,gamma) = 0.056376

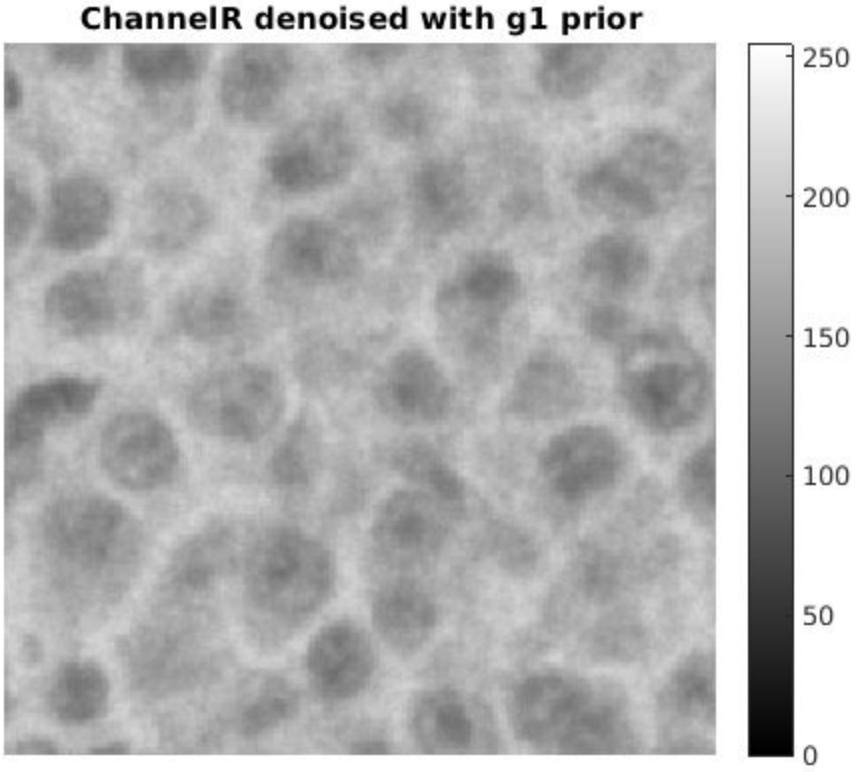
RRMSE(alpha,0.8*gamma) = 0.045332

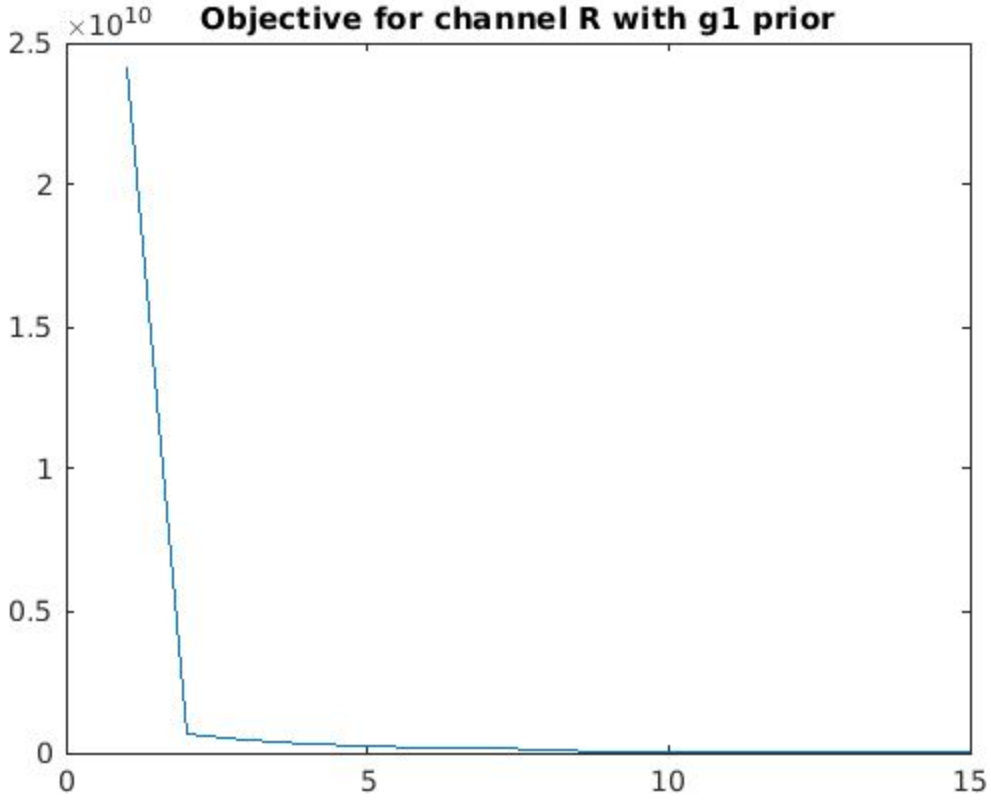
RRMSE(alpha,1.2*gamma) = 0.04484

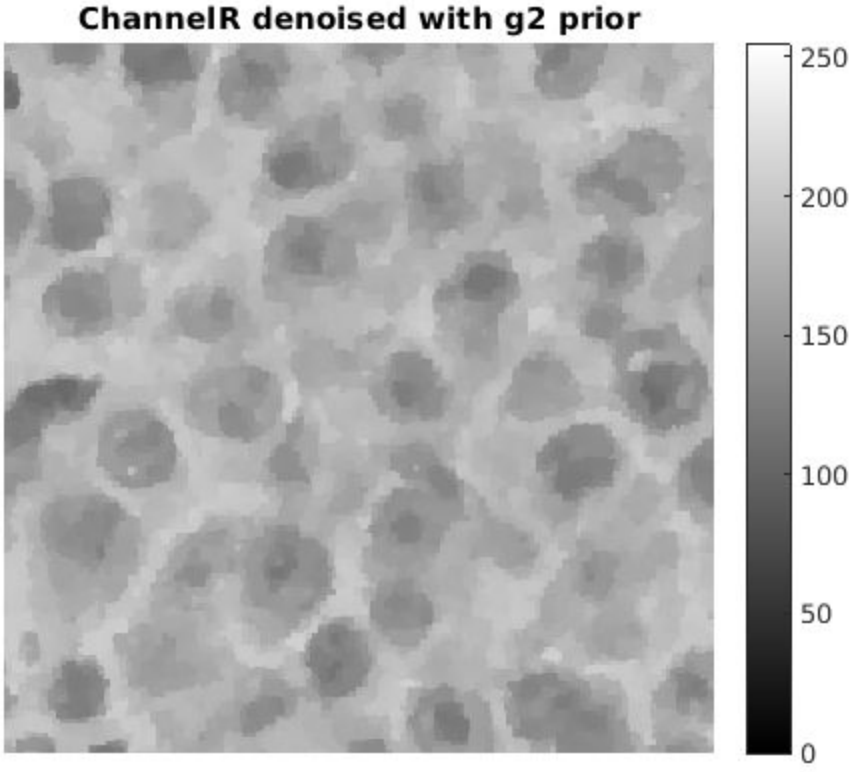
Denoising color channel R of the Image

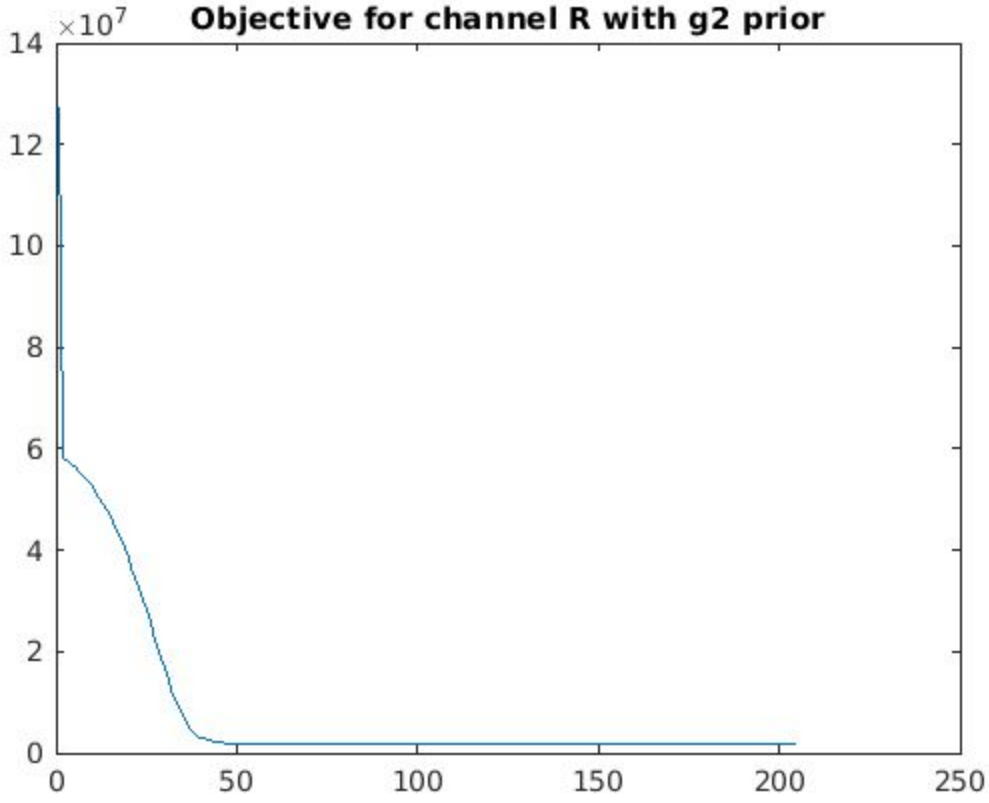


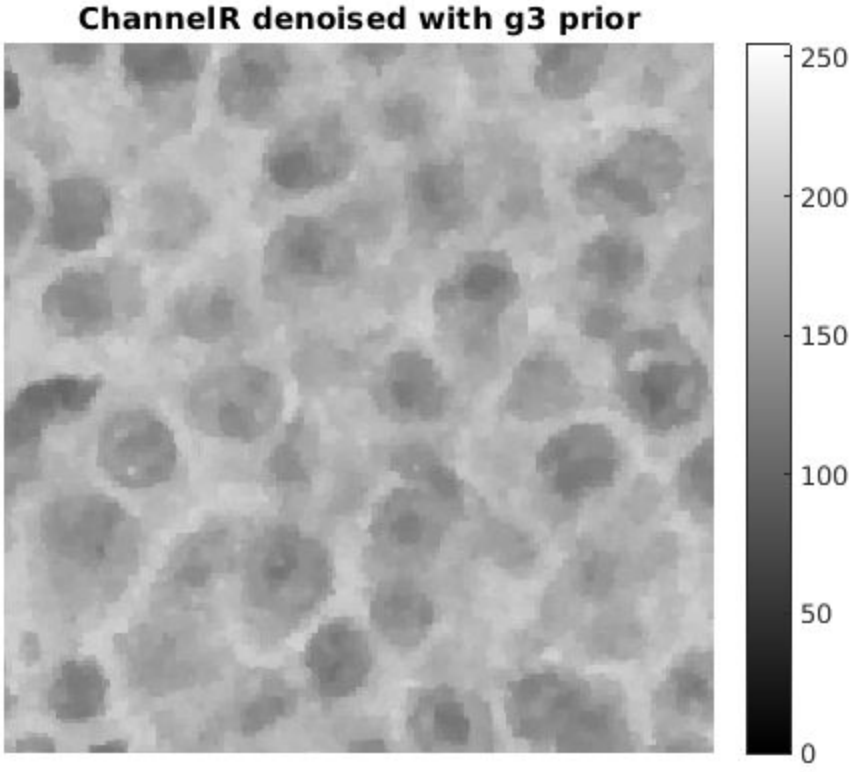

Channel R noisy
250
200
150
100
50
0

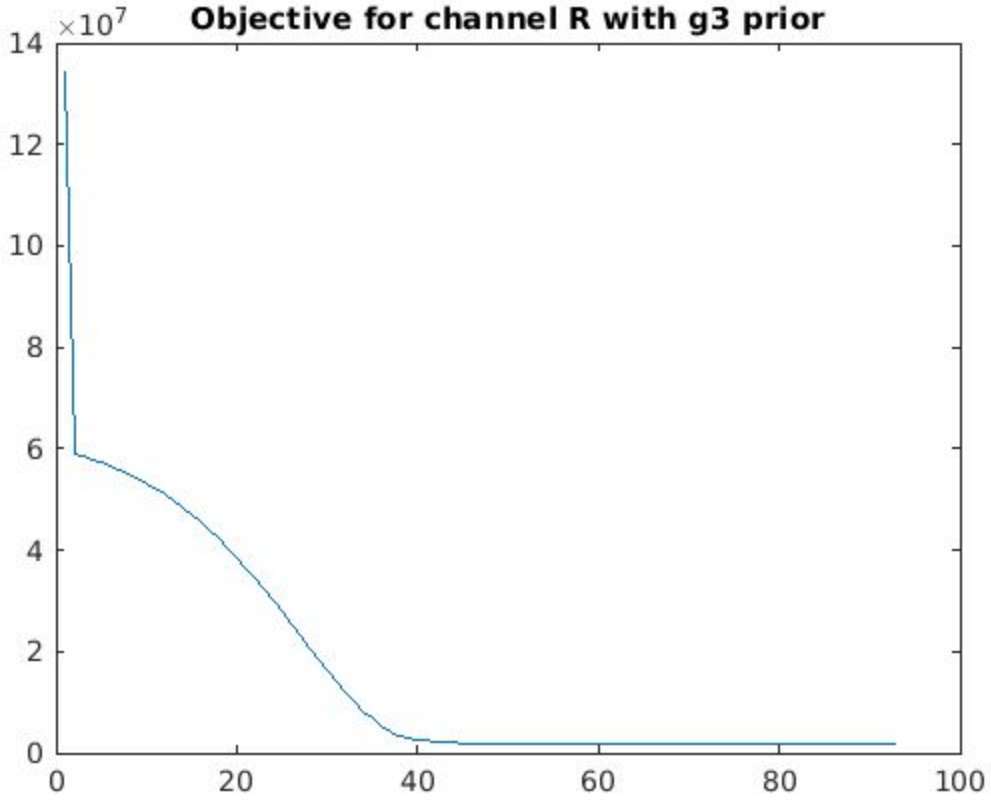

ChannelR denoised with g1 prior
250
200
150
100
50
0


Objective for channel R with g1 prior
×10^10
2.5
2
1.5
1
0.5
0
0
5
10
15

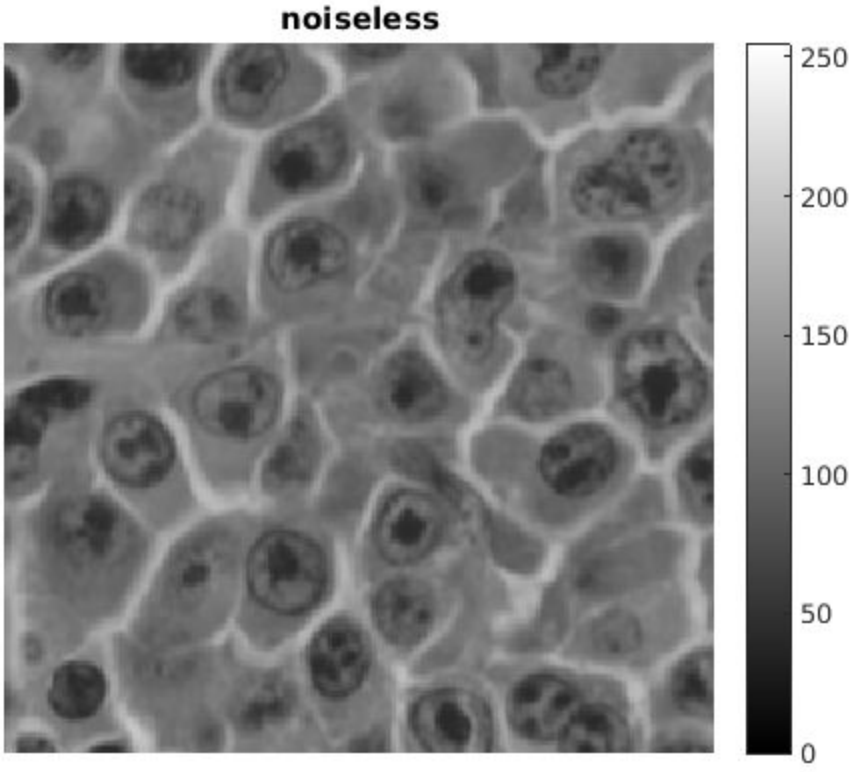

ChannelR denoised with g2 prior
250
200
150
100
50
0

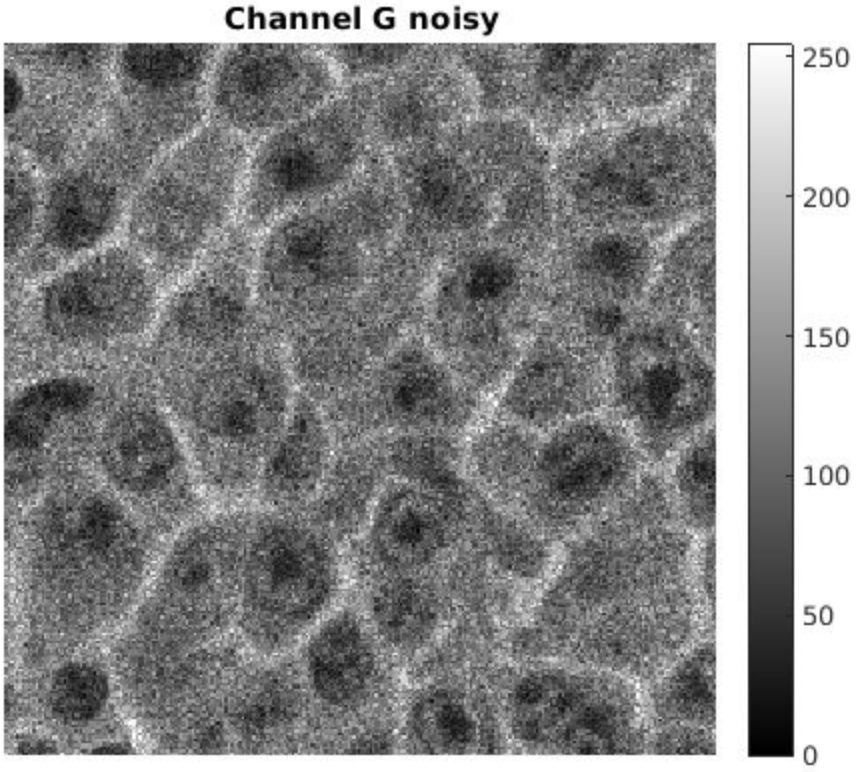

Objective for channel R with g2 prior
×10^7
14
12
10
8
6
4
2
0
0
50
100
150
200
250

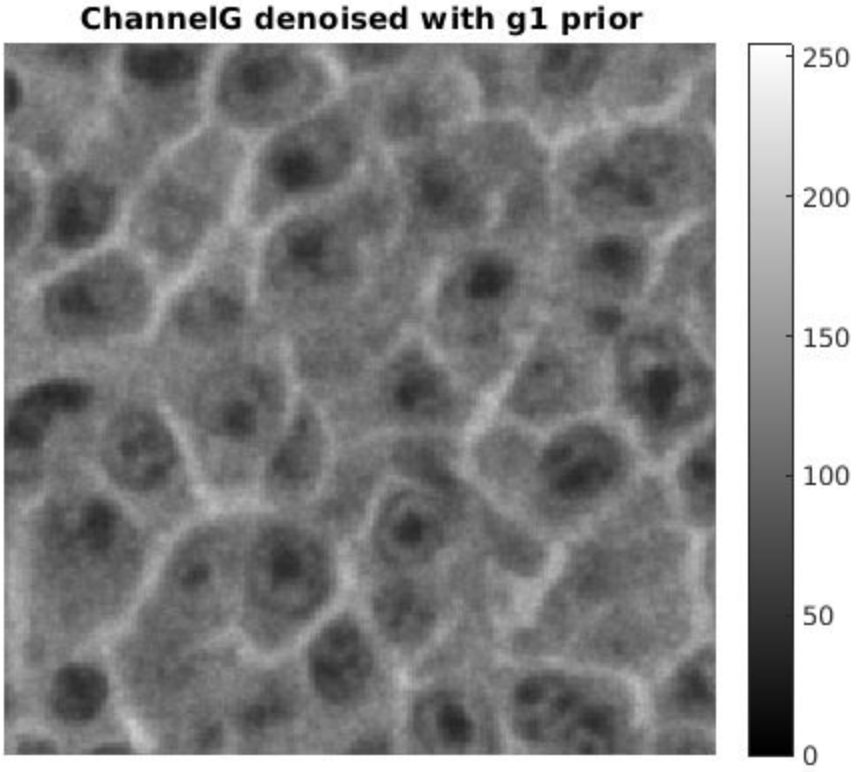

ChannelR denoised with g3 prior
250
200
150
100
50
0

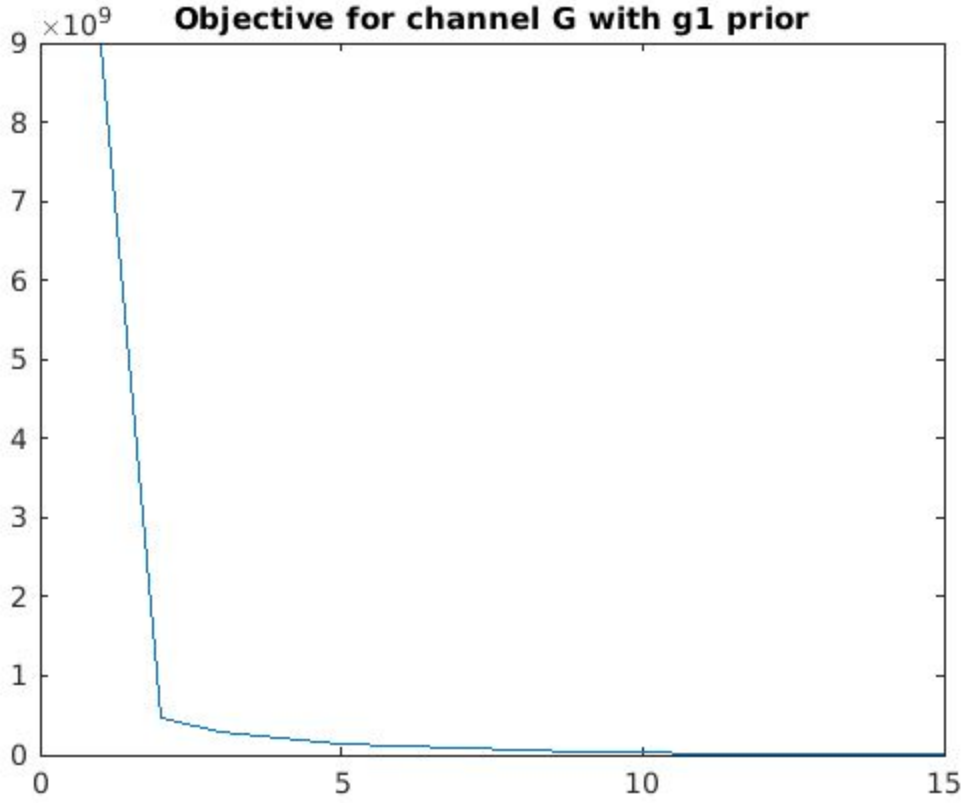


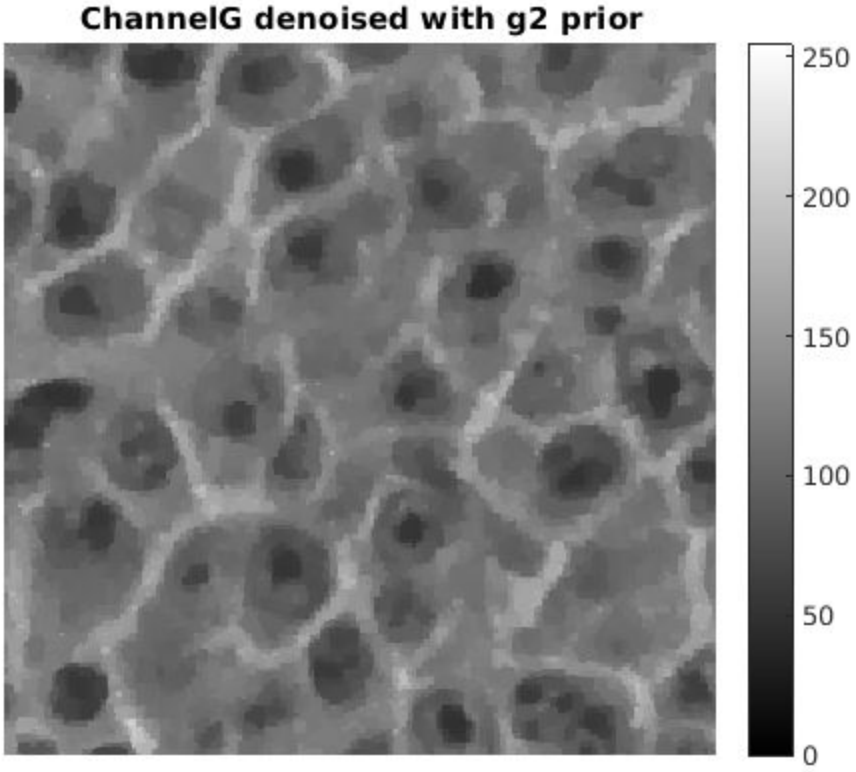
Denoising color channel G of the Image

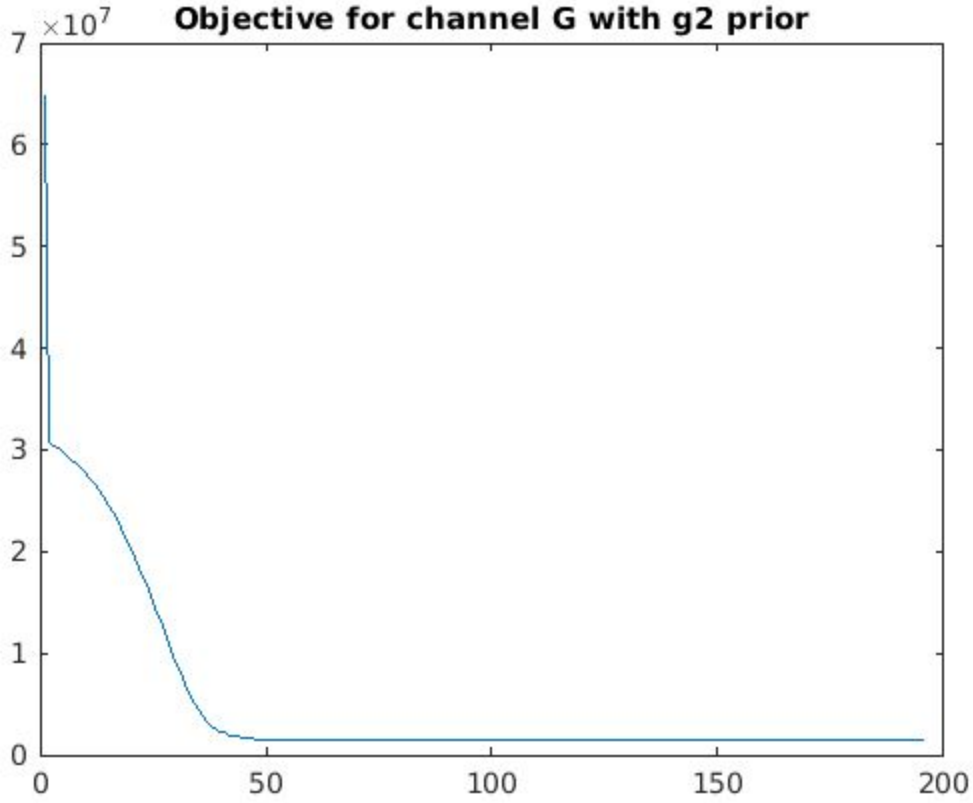

noiseless
250
200
150
100
50
0

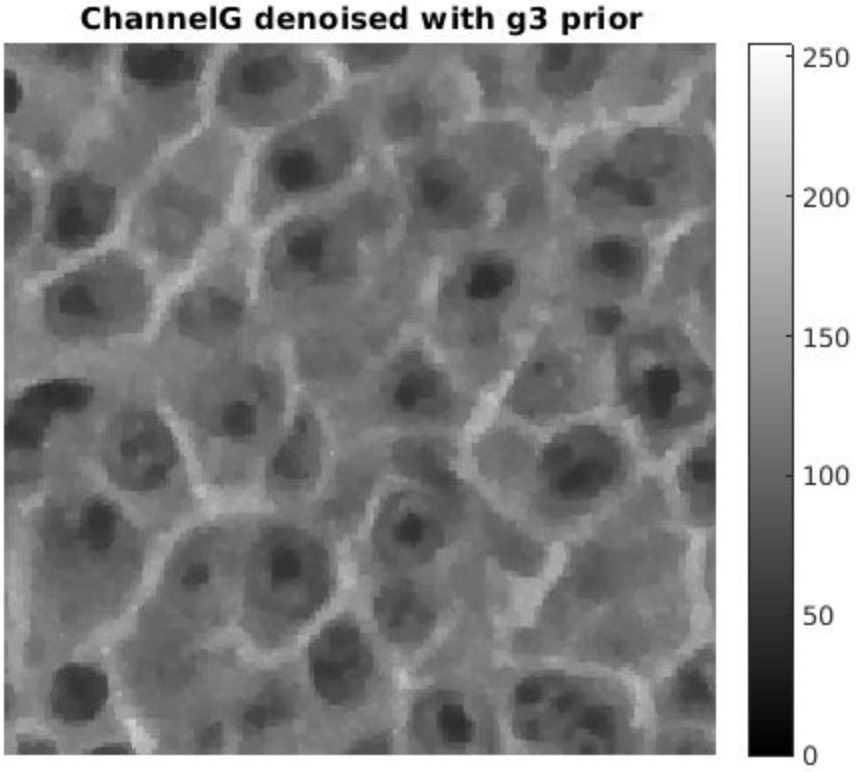

Channel G noisy
250
200
150
100
50
0

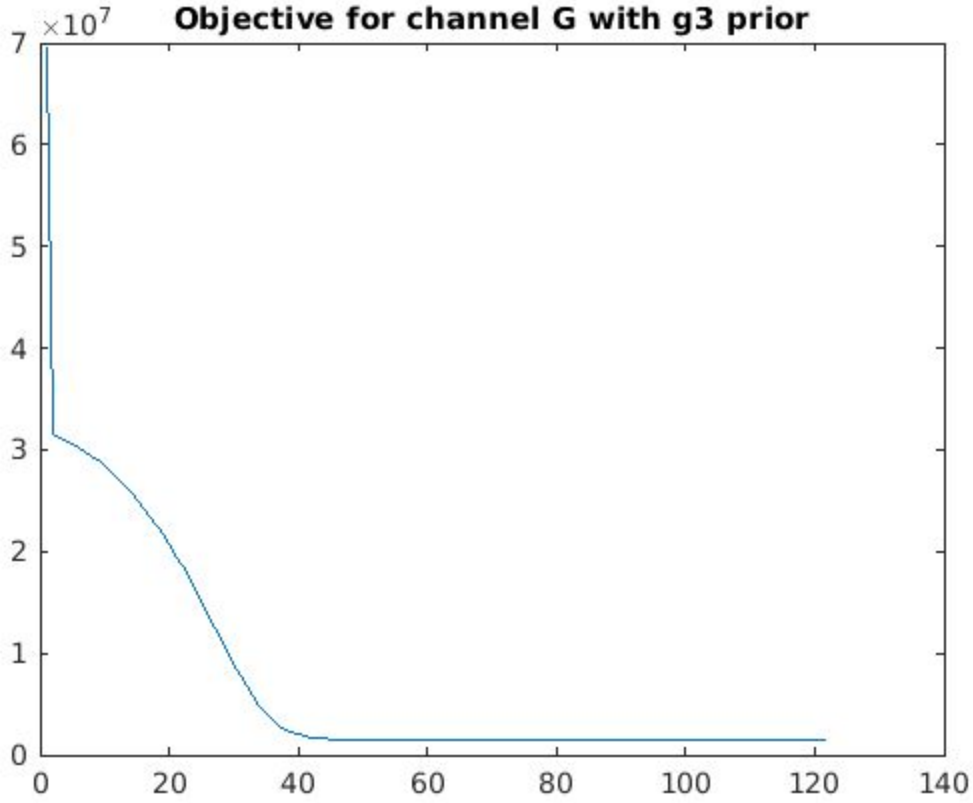

ChannelG denoised with g1 prior
250
200
150
100
50
0


Objective for channel G with g1 prior
×10^9
9
8
7
6
5
4
3
2
1
0
0
5
10
15

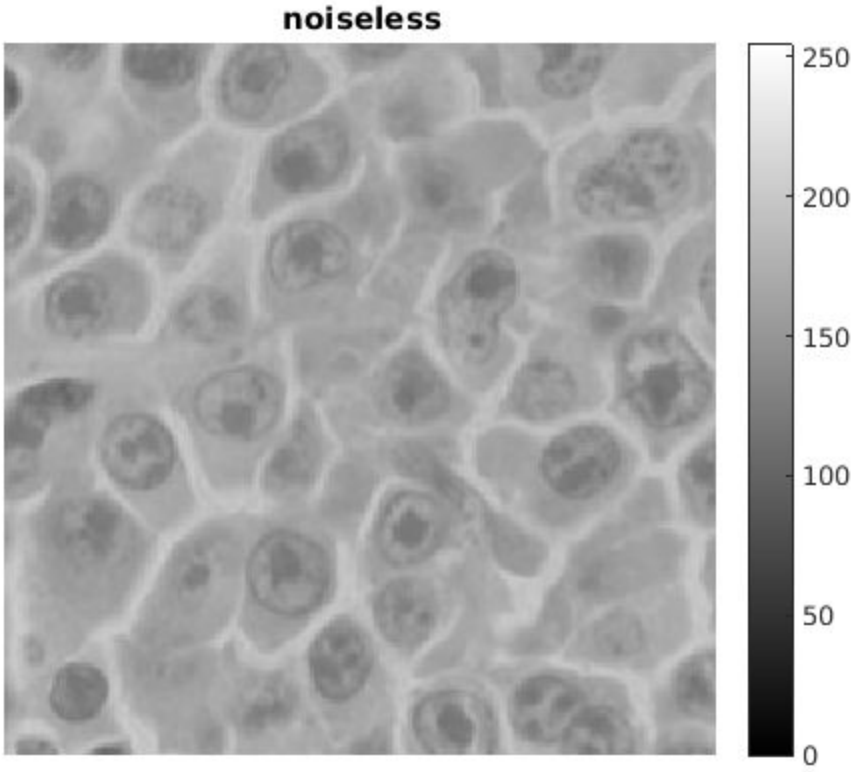

ChannelG denoised with g2 prior
250
200
150
100
50
0

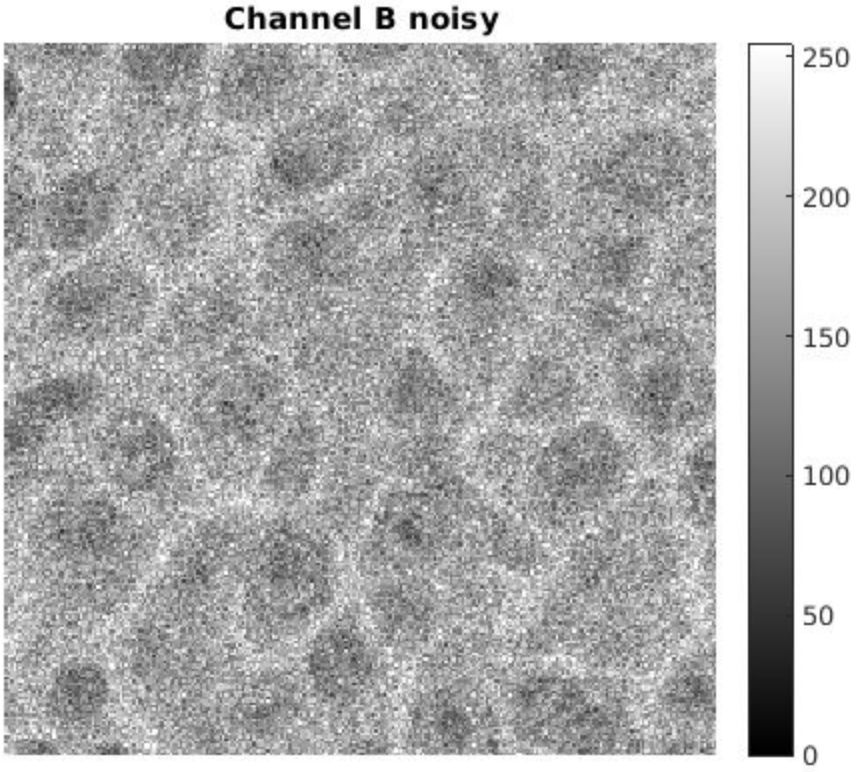

Objective for channel G with g2 prior
×10^7
7
6
5
4
3
2
1
0
0
50
100
150
200

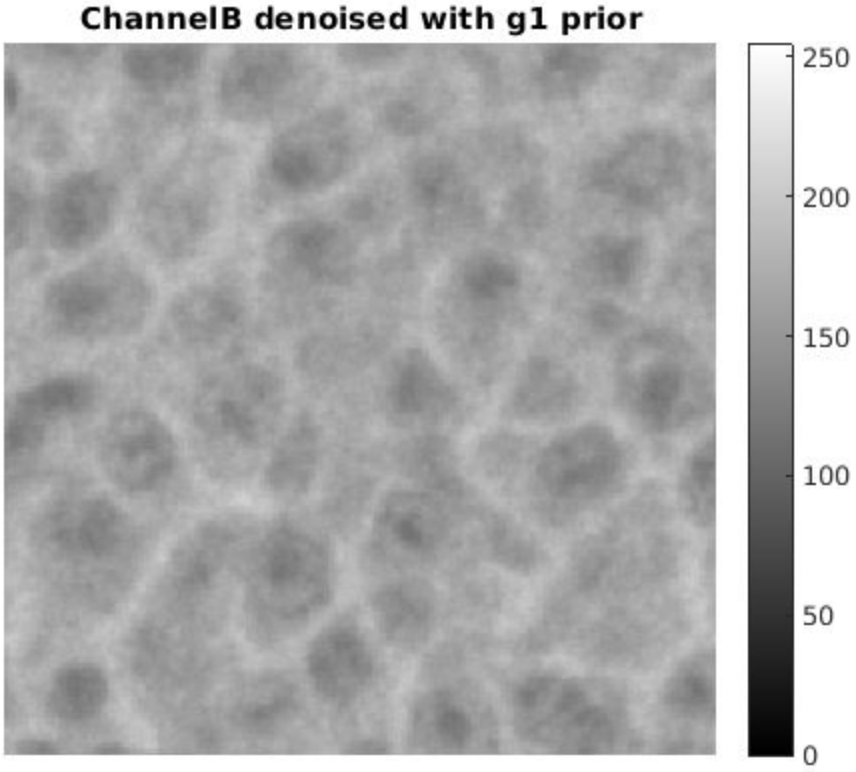

ChannelG denoised with g3 prior
250
200
150
100
50
0

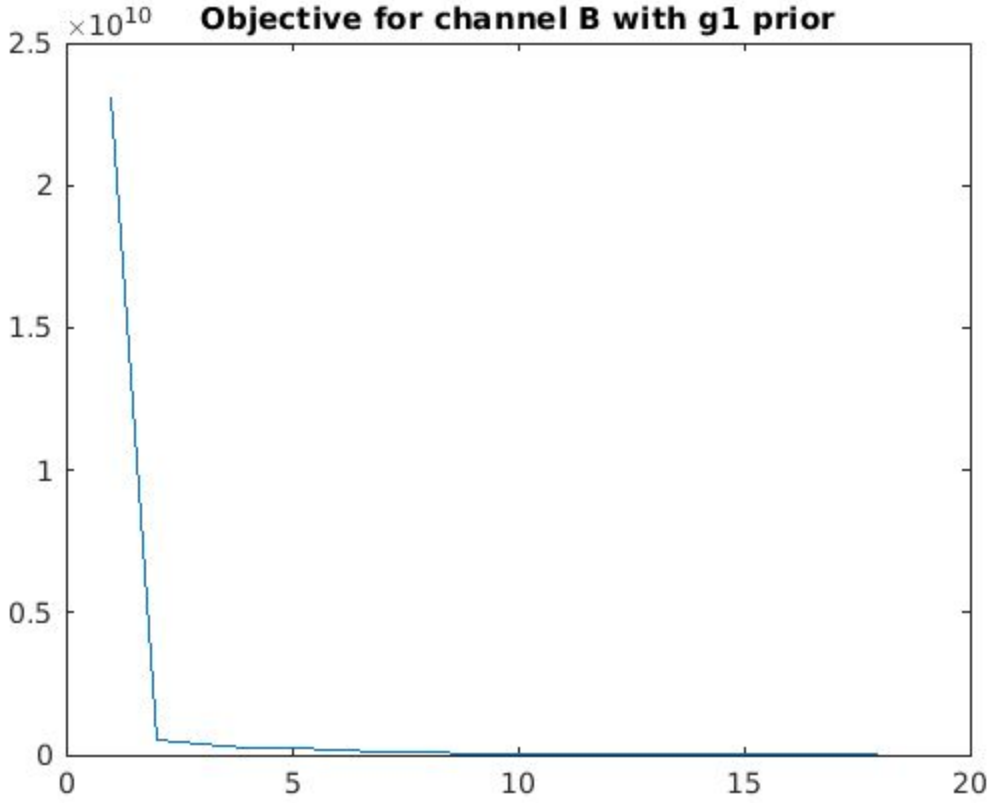


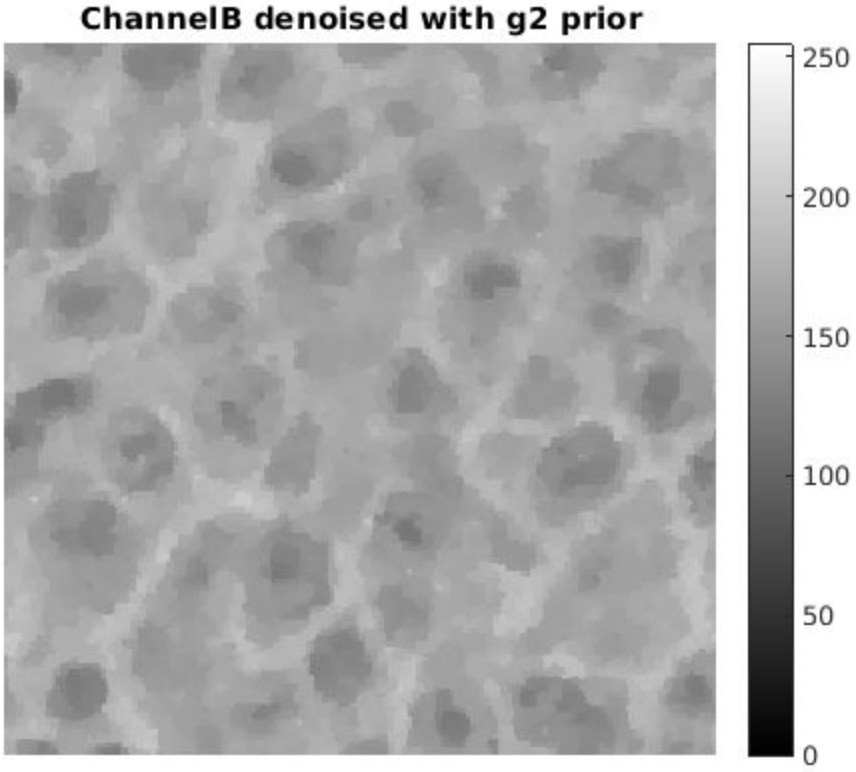
Denoising color channel B of the Image

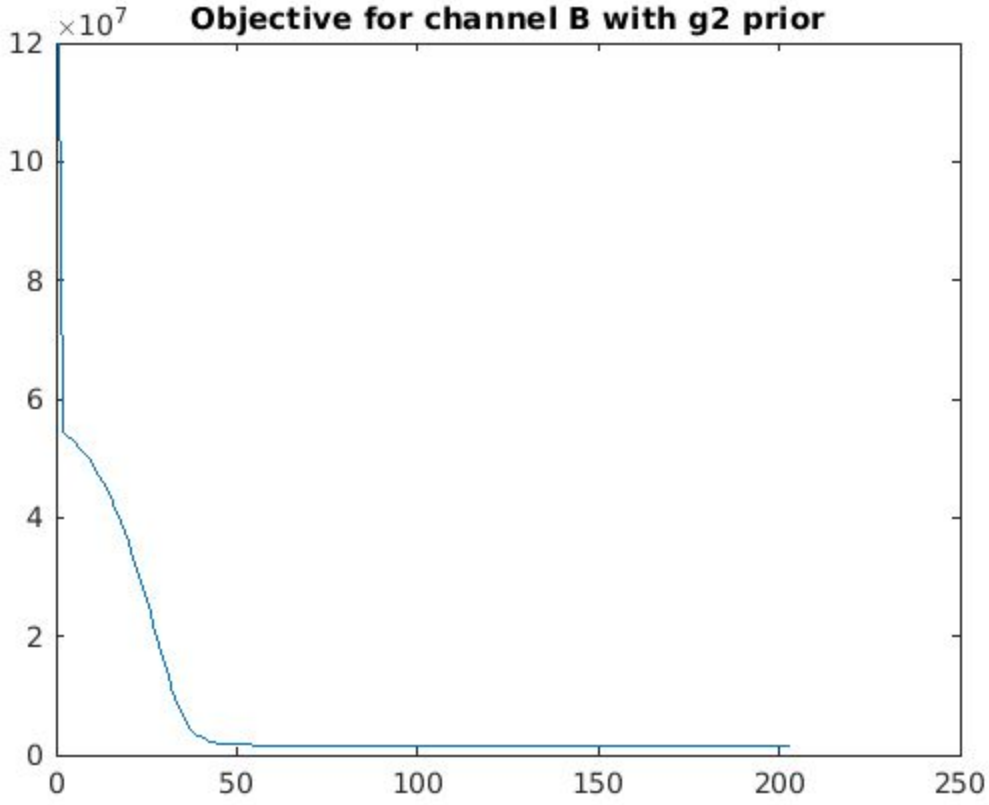

noiseless
250
200
150
100
50
0

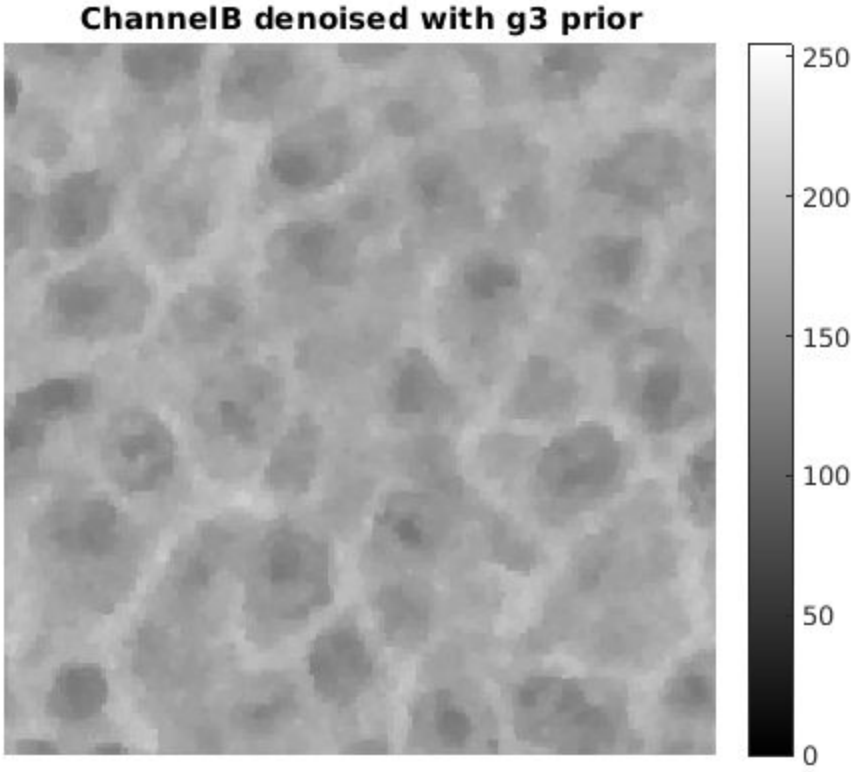

Channel B noisy
250
200
150
100
50
0

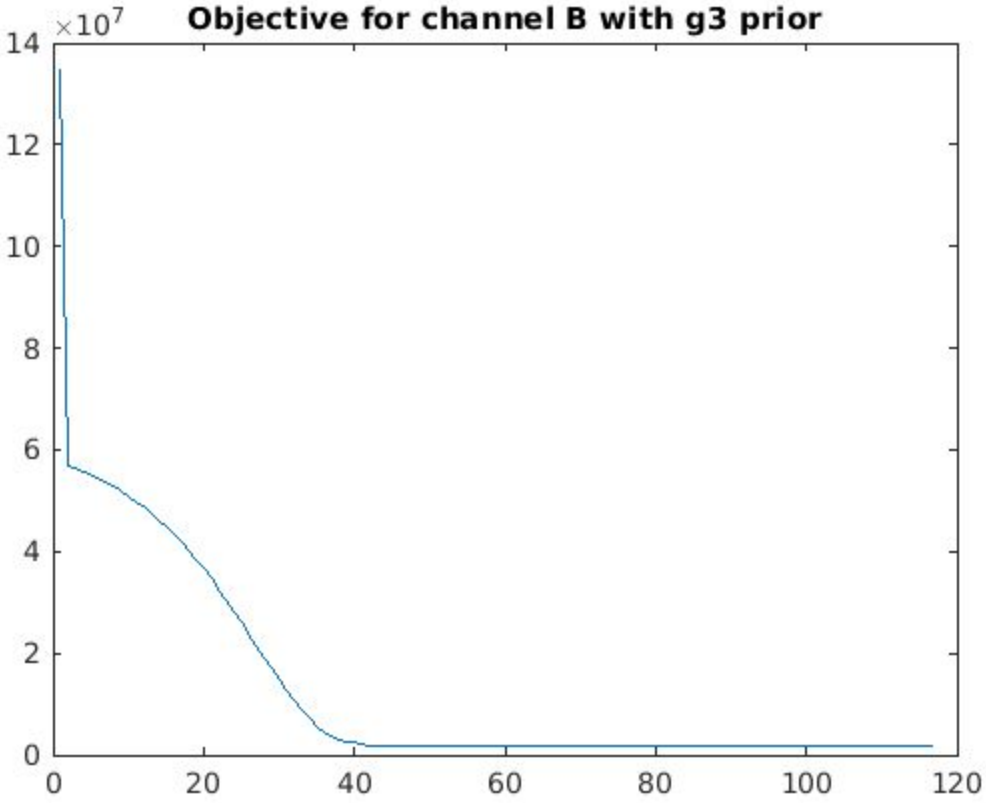

ChannelB denoised with g1 prior
250
200
150
100
50
0


Objective for channel B with g1 prior
×10^10
2.5
2
1.5
1
0.5
0
0
5
10
15
20


ChannelB denoised with g2 prior
250
200
150
100
50
0


Objective for channel B with g2 prior
×10^7
12
10
8
6
4
2
0
0
50
100
150
200
250

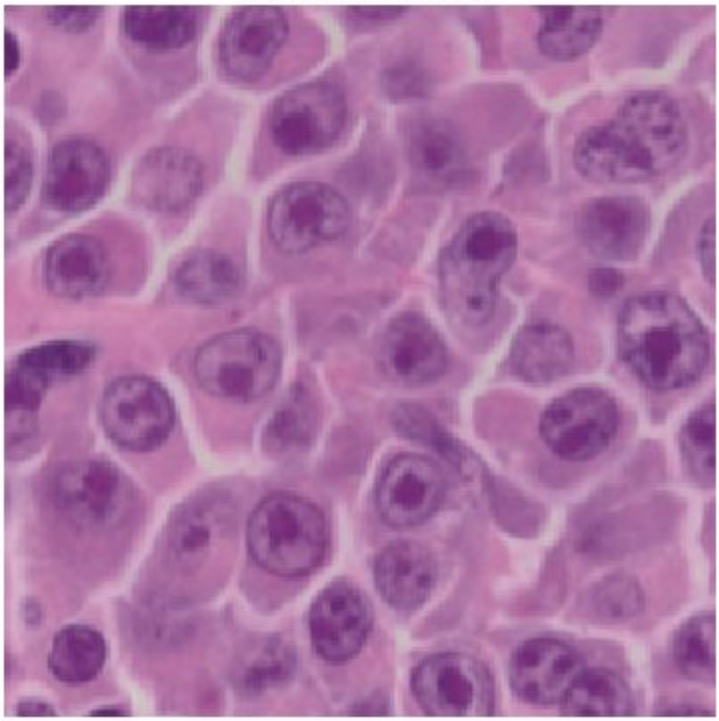

ChannelB denoised with g3 prior
250
200
150
100
50
0



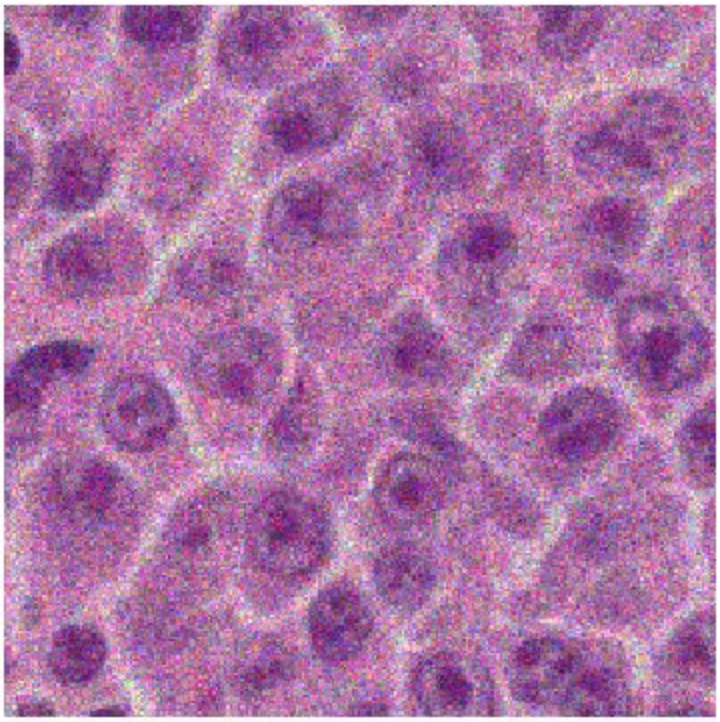
Published with MATLAB® R2019b

Overall RGB Image:

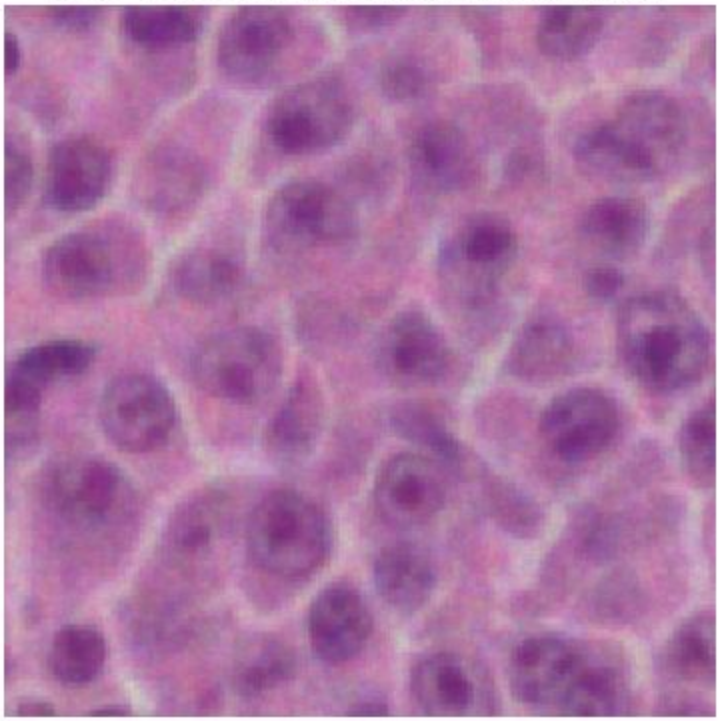
Noiseless RGB



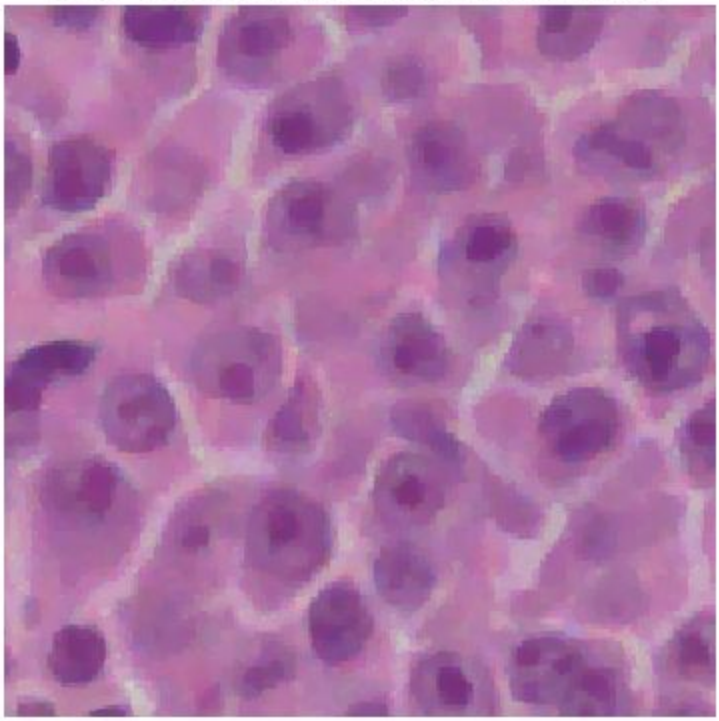
Noisy RGB



RGB Image denoised using g1 prior



RGB Image denoised using g2 prior



RGB Image denoised using g3 prior