MATH473 HW4

Jiasen Zhang

1 Problem 1

Suppose we are computing $w * x * y$, according to commutativity and associativity of convolution, we have:

$$w * x * y = w * y * x = (w * y) * x = w * (y * x)$$

Because

$$(w * y) * x = (M_y w) * x = M_x M_y w$$
$$w * (y * x) = M_{x*y} w$$

We have

$$w * x * y = M_x M_y w = M_{x*y} w$$

$$\Rightarrow \quad M_x M_y = M_{x*y}$$

2 Problem 2

I set the parameter of penalty term $\rho = 1$ with which the result performs best, then try different values of λ, compute SNR and MSE and list them in a table. From Table 1 we can see $\lambda = 0.0001$ corresponds to the largest SNR and smallest MSE. The result with $\lambda = 0.0001$ is the best.

λ	SNR	MSE
0.01	4.6000	6.70e-5
0.001	15.0521	6.00e-6
0.0001	28.9612	2.46e-7
0.00001	28.6852	2.63e-7
0.000001	19.4412	2.21e-6

Table 1: λ vs. SNR and MSE

Plot the result with $\lambda = 0.0001$ below. As shown in Figure 1, with a large SNR, the computed x_0 is almost the same as the ground truth X.

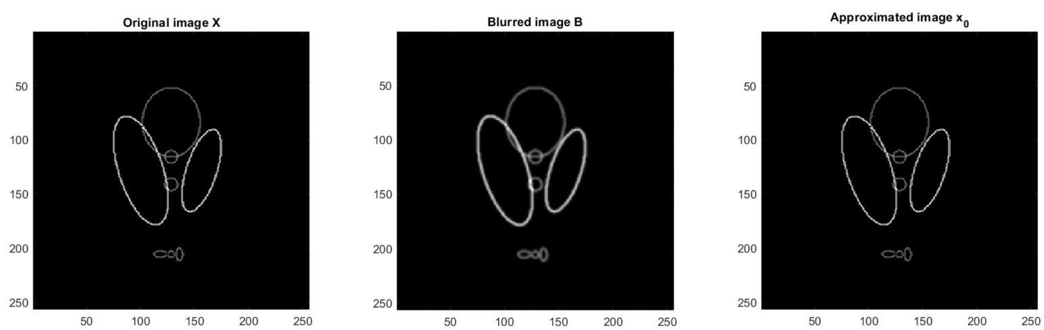

Figure 1: Result with $\lambda = 0.0001$

3 Matlab Code

3.1 Problem 2

```
clear all;clc;

tic
% simulate a blurred and noisy image J
n = 5;
sigma = 1;
psf = fspecial('gauss',[n n],sigma); % Gaussian PSF
load PartPhantomEg.mat X;
[nr,nc] = size(X);
P = zeros(nr,nc);
P(1:n,1:n) = psf; % zero pad psf
ctr = [ceil(n/2) ceil(n/2)]; % center of psf
S = fft2(circshift(P,1-ctr)); % eigenvalues of the BCCB matrix A
J = ifft2(S.*fft2(X));
J = real(J); % blurred

% get B by adding random noise
c = 0.01;
E = randn(size(J));
E = E/norm(E,'fro');
B = J + c*E*norm(J,'fro');

% initialization
x0 = zeros(nr,nc); % to be solved
w = zeros(nr,nc);
lambda = zeros(nr,nc);

% parameters
lambda0 = 0.0001; % to be tuned
rho = 1;
tol = 1e-6; % tolerance

% ADMM iteration
for k=1:1000000
    % old x0 and w
    xold=x0;
    wold=w;

    % update new x0
    %x0 = ifft2( fft2(conj(S).*B+rho*w-lambda)./(S.*conj(S)+rho) );
    x0 = ifft2( (conj(S).*fft2(B)+rho*fft2(w-lambda/rho))./(S.*conj(S)+rho) );
    %x0=real(x0);

    % update new w with new x0
    w = sign(x0+lambda/rho).*max(0,abs(x0+lambda/rho)-lambda0/rho);
    %w = real(w);

    % update lambda with new w and new x0
    lambda = lambda + rho*(x0-w);

    % stopping criteria
    dx = max(max(abs(x0-xold)));
    dw = max(max(abs(w-wold)));
    d = max(dw,dx);
    if mod(k,500)==0
        fprintf('%d   %e\n',k,d);
    end
    if d<tol
        break;
```

```
    end
end
%x0=real(x0);

% compute MSE and SNR
mse = sum(sum((x0-X).^2))/(nr*nc);
p = sum(sum(X.^2))/(nr*nc);
snr = 10*log10(p/mse);
fprintf('MSE = %f \n',mse);
fprintf('SNR = %f \n',snr);

% plot
subplot(1,3,1);
imagesc(X);axis image;colormap(gray);
title('Original image X');
subplot(1,3,2);
imagesc(B);axis image;colormap(gray);
title('Blurred image B');
subplot(1,3,3);
imagesc(x0);axis image;colormap(gray);
title('Approximated image x_0');
toc
```