

```

% gda12_01
% image deblurring example
% supports Figure 12.1 and 12.2
clear all;
figure(1);
clf;

% black & white image
c = colormap( [[0:255]', [0:255]',[0:255]'] / 255 );

% read in true image and demean it
mtrue = double(imread(' ../data/tern.tif', 'tif' ));
mtrue = mtrue - mean(mean(mtrue));
[I, J] = size( mtrue );

% plot true image
subplot(2,3,1);
set(gca, 'LineWidth', 2);
hold on;
axis( [1, I, 1, J] );
axis ij;
imagesc([1, I], [1, J], mtrue);
title('true image');

% data is blurred version of image
mblurred = zeros( I, J );

% blur is over rows of image with this filter
Lb = 100;

% try 2 kinds of blurs
if( 0 )
    blur = [Lb:-1:1]'/Lb; % linear ramp-down
else
    blur = ones(Lb,1)/Lb; % boxcar
end

% crease sparse data kernel
G = spdiags( ones(J,1)*blur', [0:Lb-1], J, J );

% data is blurred version of image
for i=1:I
    dobs(i,:) = (G*(mtrue(i,:)))';
end

% plot blurred image
subplot(2,3,2);
set(gca, 'LineWidth', 2);
hold on;
axis( [1, I, 1, J] );
axis ij;
imagesc([1, I], [1, J], dobs);
title('blurred image');

epsilon=1.0e-6; % damping, just in case GGT is singular
GGT = G*G'+epsilon*speye(J,J);

% damped minimum length solution
mest = zeros(I,J);

```

```

% process image row-wise
for i = [1:I]
    dobsrow = dobs(i,:);
    mestrow = G'*(GGT\dobsrow);
    mest(i,:)=mestrow';
end

% plot reconstructed image
subplot(2,3,3);
set(gca,'LineWidth',2);
hold on;
axis( [1, I, 1, J] );
axis ij;
imagesc([1, I], [1, J], mest);
title('reconstructed image');

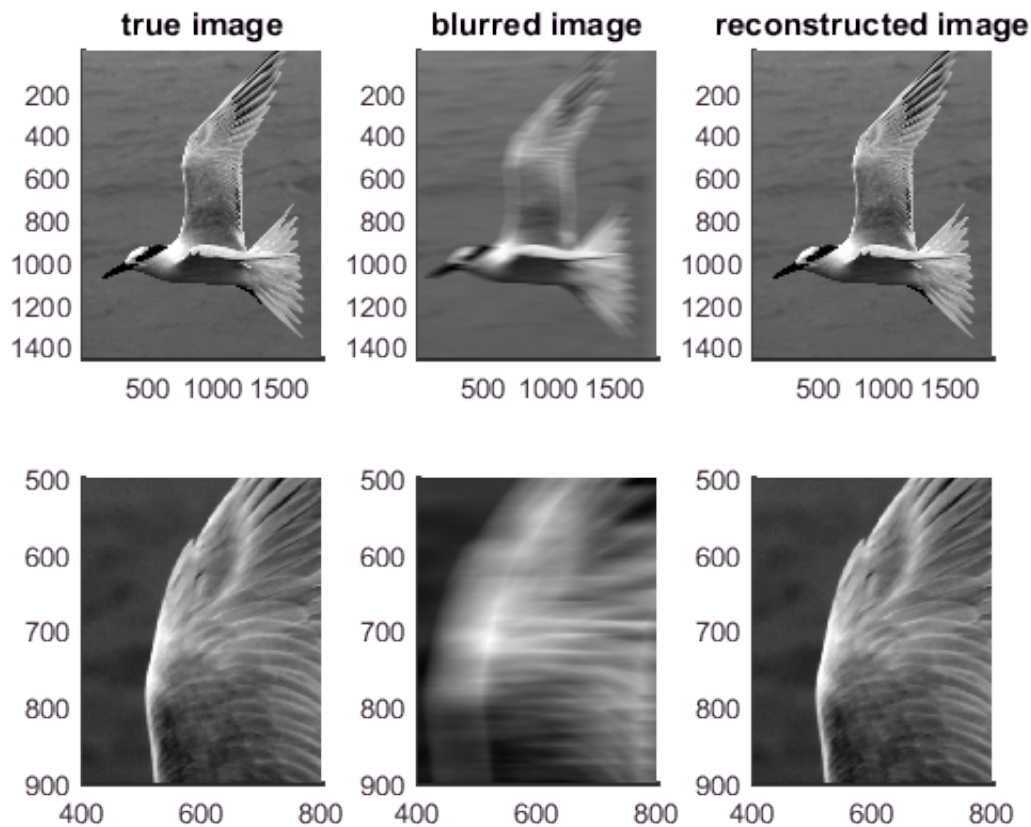
% plot closeup
I1=400;
I2=800;
J1=500;
J2=900;

subplot(2,3,4);
set(gca,'LineWidth',2);
hold on;
axis( [I1, I2, J1, J2] );
axis ij;
imagesc([I1, I2], [J1, J2], mtrue(I1:I2,J1:J2));

subplot(2,3,5);
set(gca,'LineWidth',2);
hold on;
axis( [I1, I2, J1, J2] );
axis ij;
imagesc([I1, I2], [J1, J2], dobs(I1:I2,J1:J2));

subplot(2,3,6);
set(gca,'LineWidth',2);
hold on;
axis( [I1, I2, J1, J2] );
axis ij;
imagesc([I1, I2], [J1, J2], mest(I1:I2,J1:J2));

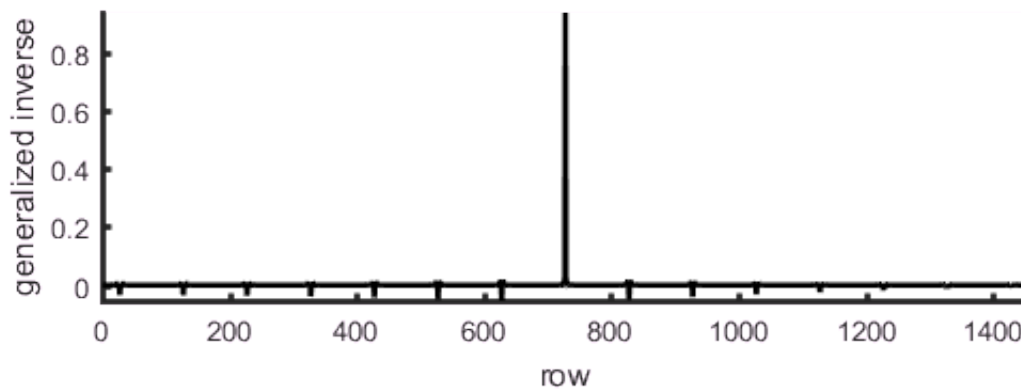
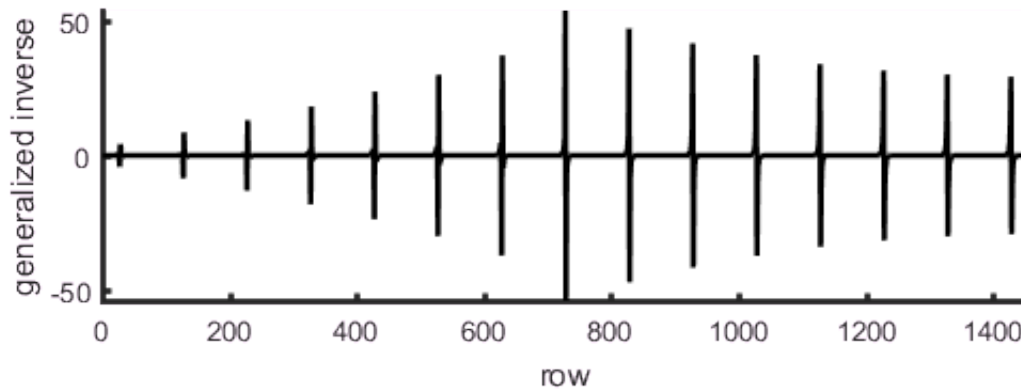
```



% Figure 12.1 Example of removing blur from image. (A) True image. (B) Image  
 % blurred with 100 pixel-wide boxcar filter. (C) Estimated image, unblurred  
 % using the minimum length generalized inverse. (D-F) Enlargement of portion  
 % of images (A-C). MatLab script gda12\_01.

% plot central row of generalized inverse and resolution matrix

```
Gmg = G'*inv(GGT);
R = Gmg*G;
figure(2);
subplot(2,1,1);
set(gca,'LineWidth',2);
hold on;
tmp = Gmg( J/2, : )';
axis( [0, J, min(tmp), max(tmp)] );
xlabel('row');
ylabel('generalized inverse');
plot( [1:J]', tmp, 'k-', 'LineWidth', 2 );
subplot(2,1,2);
set(gca,'LineWidth',2);
hold on;
tmp = R( J/2, : )';
axis( [0, J, min(tmp), max(tmp)] );
xlabel('row');
ylabel('generalized inverse');
plot( [1:J]', tmp, 'k-', 'LineWidth', 2 );
```



```
% Figure 12.2 (A) Central row (row 728) of the generalized inverse of the
% image deblurring problem. (B) Central row (row 728) of the corresponding
% resolution matrix. The sidelobes are about 6% of the amplitude of the
% central peak. MatLab script gda12_01.
```

```
% some statistics of R
% Note: for some reason, max() returns Rmax as a 1x1 sparse
% matrix and fprintf refuses to print it unless it is converted
% to a full matrix by a call to full()
[Rmax, jRmax] = max(R(J/2,:));
fprintf('R(i=%d, j) had maximum %f at j=%d\n', J/2, full(Rmax), jRmax );
```

```
R(i=728, j) had maximum 0.938778 at j=728
```

```
Rpmax = max(abs([R(J/2,1:jRmax-1), R(J/2,jRmax+1:J)]));
fprintf('largest off diagonal element is %f as large\n', full(Rpmax)/Rmax );
```

```
largest off diagonal element is 0.057252 as large
```

```
% restore colormap
colormap('default');
```