IIIT Vadodara
Autumn 2018-19
TE3 Computer Vision
Lab-3: Ill-posedness and ill-conditioning: Need for regularization

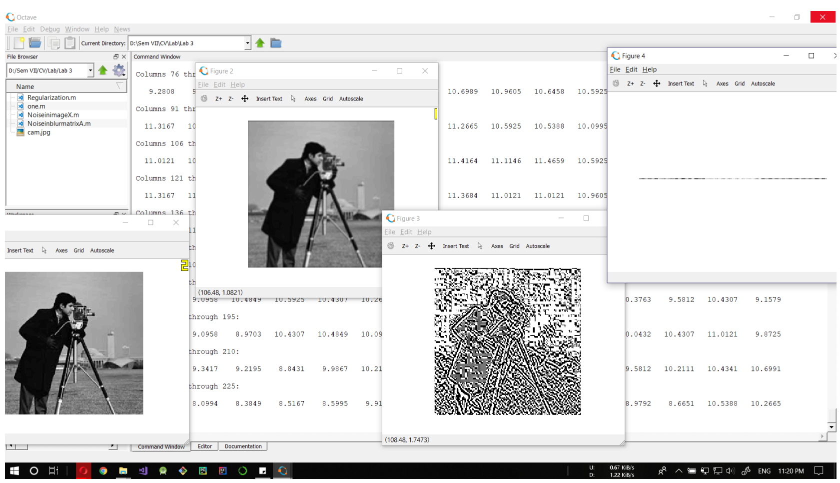
Shivvrat Arya (201551059)

August 28, 2018

Problem 1. Naive deblurring Xb = A1 Y1: Consider an image X. Assume that it is a focused image. Now, consider a blur matrix A (nonsingular, i.e., invertible, but with a high condition number). Find the condition number and the inverse of the blur matrix. Convolve the image with the blur matrix, i.e., generate blurred (defocused) version of the input image and display it, Y1 = A X. Such images are generated when your camera is out of focus. Let us try to deblur the blurred image using an algorithmic approach, i.e., estimate the deblurred (focused) image Xb. To this end, let us consider a naive approach, i.e., take the inverse of the blur matrix and convolve with the blurred image, Xb = A1 Y1. Display, compare and comment on the results. Also calculate the root mean- squared error (RMSE) between input image and estimated deblurred image. Finally display the RMSE map showing mean and standard deviation values. Observe that deblurring is an ill-posed problem, why?. Note that in the real scenario we are not even aware of the blur matrix and/or the focused image. Hence, the problem becomes severely ill-posed, why?.

Solution

```
im = imread("cam.jpg");
figure(1)
imshow(im)
windowSize = 3;
kernel = [1, 1.00004,1; 1,1,1.00004;1.0004,1,1]/9.000012
blurredImage = conv2(single(im), kernel, 'same');
figure(2)
inv_filter = pinv(kernel)
imshow(uint8(blurredImage), [0 255]);
image2 = conv2(single(blurredImage),inv_filter);
figure(3)
image2 = image2(1:225,1:225);
imshow(uint8(image2), [0 255])
RMSE = sqrt(mean((im - image2).^2))
figure(4)
I = mat2gray(RMSE);
imshow(im2single(I))
```

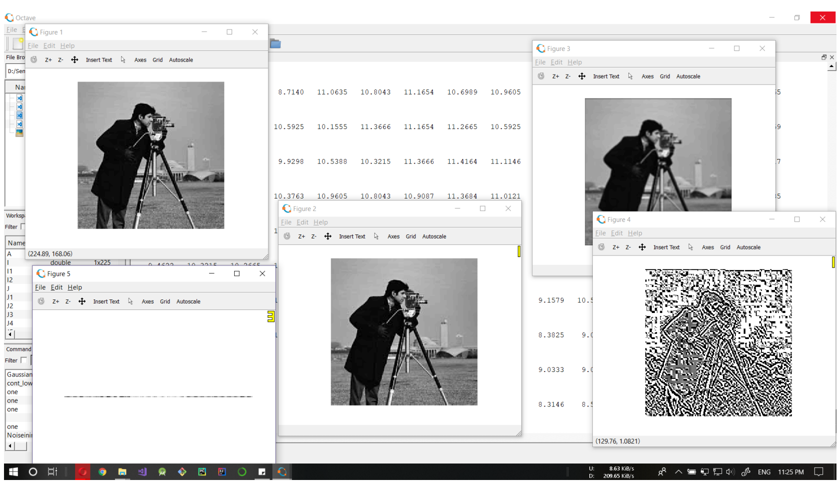


In this problem we have tried to first blur the image then deblur it by using basic inversion of matrix. Firstly the im age is convoluted with the blurring matrix and then the blurring matrix is inverted and again convoluted with the blurred image to get an approximation of the blurred image. Then we tried to find out how the image varies from the orignal one.

Question 2. Noise in image X: Reconsider the Q1. Let us add a very small amount of Gaussian noise N, i.e., zero mean with say 0.0001 variance, to the blurred image. Note that now the resultant image is not only blurred but also noisy, i.e., Y2 = A X + N. Apply the same algorithm (Q1) and try to estimate the deblurred or focused image, i.e., Xb = A1 Y2. Display the result, comment (ill-conditioning) and compare with Q1.

Solution J = imread("cam.jpg");
figure(1)

```
imshow(J)
p3= 0;
p4 = 0.0001;
im = J + sqrt(p4)*randn(size(J)) + p3;
figure(2)
imshow(im)
windowSize = 3;
kernel = [1, 1.00004,1; 1,1,1.00004;1.0004,1,1]/9.000012
blurredImage = conv2(single(im), kernel, 'same');
figure(3)
inv_filter = pinv(kernel)
imshow(uint8(blurredImage), [0 255]);
image2 = conv2(single(blurredImage),inv_filter);
figure(4)
image2 = image2(1:225,1:225);
imshow(uint8(image2), [0 255])
RMSE = sqrt(mean((im - image2).^2))
figure(5)
I = mat2gray(RMSE);
imshow(im2single(I))
```

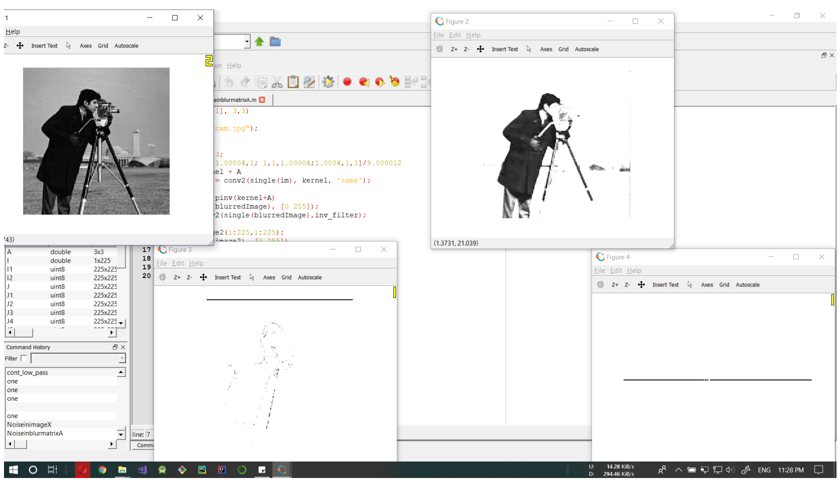


In this quesiton we did the same thing as the first question, just including a error in the blurred matrix and then aprroximating the actual image from that image.

Question 3. Noise in blur matrix A: Repeat the Q2 but add the same noise in the blur matrix A, i.e., Y3 = (A + N) X. Again apply the same deblurring algorithm (Q1) and try to recover the focused image, i.e., Xb = (A+ N) 1 Y3. Now display the result, comment (ill-conditioning) and compare with Q1 and Q2.

Solution
```
A = randi([0 1], 3,3)
im = imread("cam.jpg");
```

```
figure(1)
imshow(im)
windowSize = 3;
kernel = [1, 1.00004,1; 1,1,1.00004;1.0004,1,1]/9.000012
kernel = kernel + A
blurredImage = conv2(single(im), kernel, 'same');
figure(2)
inv_filter = pinv(kernel+A)
imshow(uint8(blurredImage), [0 255]);
image2 = conv2(single(blurredImage),inv_filter);
figure(3)
image2 = image2(1:225,1:225);
imshow(uint8(image2), [0 255])
RMSE = sqrt(mean((im - image2).^2))
figure(4)
I = mat2gray(RMSE);
imshow(im2single(I))
```

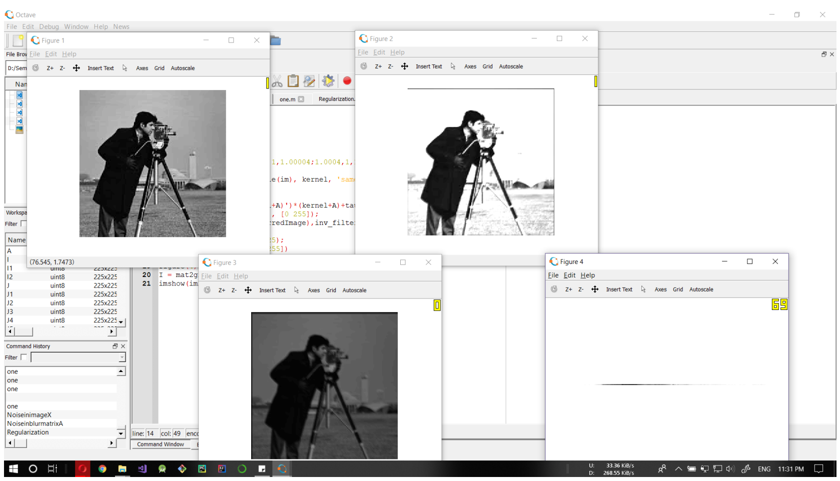


In this question we did the same thing as the first question, just including a error in the blurring matrix and then approximating the actual image from that image.

Question 4. Regularization: To use additional information (from the given problem) explicitly, at the start, to construct families of approximate solutions to the inverse ill-posed problems (IIPs). For example (referring to our deblurring problem), Xb = kY AXk2 + kLXk2, (1) where, is the regularization parameter (typically a small value) to be estimated and L is, say, matrix representative of first-order derivative operator (known as Tikhonov regularization). Regularizations are now one of the most powerful tools for the solution of the IIPs. Implement the Tikhonov regularization technique for the deblurring problem.

Solution
```
A = randi([0 1], 3,3)
im = imread("cam.jpg");
```

```
figure(1)
imshow(im)
windowSize = 3;
kernel = [1, 1.00004,1; 1,1,1.00004;1.0004,1,1]/9.000012
kernel = kernel + A
blurredImage = conv2(single(im), kernel, 'same');
figure(2)
tau = 3*eye(3)
inv_filter = pinv(((kernel+A)')*(kernel+A)+tau'*tau)
imshow(uint8(blurredImage), [0 255]);
image2 = conv2(single(blurredImage),inv_filter);
figure(3)
image2 = image2(1:225,1:225);
imshow(uint8(image2), [0 255])
RMSE = sqrt(mean((im - image2).^2))
figure(4)
I = mat2gray(RMSE);
imshow(im2single(I))
```



In this question we used the Tikhonov regularixation to get a better solution.