Programming Assignment 2

COMPUTATIONAL PHOTOGRAPHY-EE5176

PRAVEENPRASATH B
EP21B028
INDIAN INSTITUTE OF TECHNOLOGY, MADRAS

1 Motion deblurring with conventional camera

You are given a clean image of a scene having M rows and N columns (fish.png). We capture the scene using a static conventional camera with the aperture kept open for the full exposure time. The exposure time is 52 seconds. For simplicity, consider that the camera captures the scene at times t = 0,...,51 (i.e. at 52 instants). At time t = 0, the image has zero translation. Then, the whole image as seen by the camera moves at 1 pixel per second to the right (horizontal translation)

1.1 Generate the blurred image having M rows and N + 51 columns captured by the camera. Generate translated versions of the image for t = 0,...,51, and average them. Add Gaussian noise of mean 0 and standard deviation 1 (with respect to the maximum intensity of 255) from gaussNoise.mat. Display the blurred image. [5 marks]

Figure 1: Original Image

Figure 2: Gaussian Noise

Figure 3: Blurred and Noise added Image

1.2 Form the blur matrix A of size $(N + 51) \times N$ corresponding to the horizontal translation described above. Note that this matrix will be a Toeplitz matrix. Display A using imshow command. [5 marks]

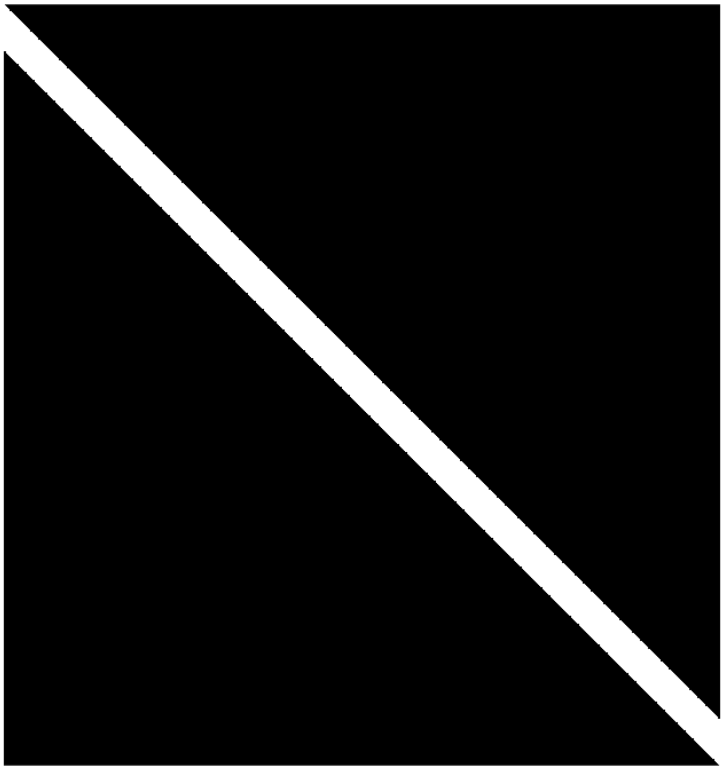

Figure 4: Blur Kernel A

1.3 Use A to deblur the blurred image using least squares. The size of the deblurred image should be $M \times N$ for each colour channel. (Note: Since the blurring is horizontal, you have to operate A on each row of the blurred image.) Display the deblurred image. Determine the RMSEs between the given clean image and the deblurred image. Use the formula

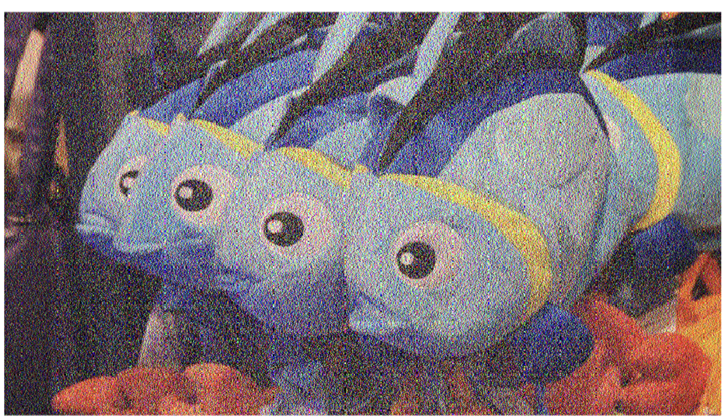

Figure 5: Deblured Image using A

Root mean square error between clean image and deblured image is

```
RMSE = 0.4889
```

1.4 What do you observe about the deblurred image? Explain the reason.

The Blur Kernel(A) with which blur was simulated and blurred image was deblurred is not well conditioned . The Condition number of Blur Matrix (A) is 2.6952e+03, Which indicates the Matrix is not nicely invertible and the Noise gets amplified and is not a good of Motion Deblurring.

2 Motion deblurring with flutter shutter

With the same setup as in the previous section, we use a flutter shutter camera with these 2 codes:

(i) 1010000111000000101000011001111011101011100100110 0111

(ii) 1010101010101010101010101010101010101010101010101010

where each bit represents one second of the exposure time.

2.1 Generate the blurred images with noise (same as in Problem-1) for the exposure codes given above. [4 marks]

Figure 6: Original Image

Figure 7: Blurred and Normalized(for visibility) Image using Code 1

Figure 8: Blurred and Normalized(for visibility) Image using Code 1

2.2 b) Form the blur matrix A of size (N + 51) × N for each case.

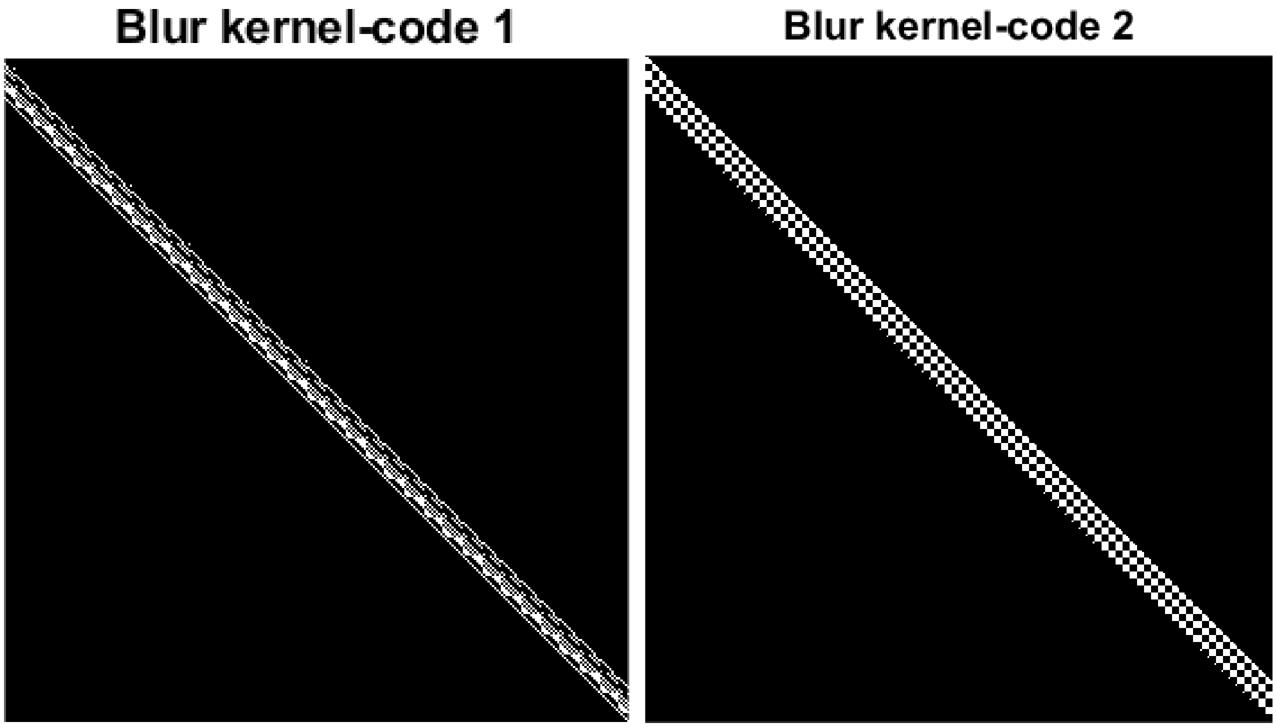

Figure 9: Blur Matrix (A) using Different codes

2.3 Plot and compare the DFTs of the conventional and flutter shutter codes. Use the first column of the respective A matrices. Plot the magnitude in decibels. Comment.

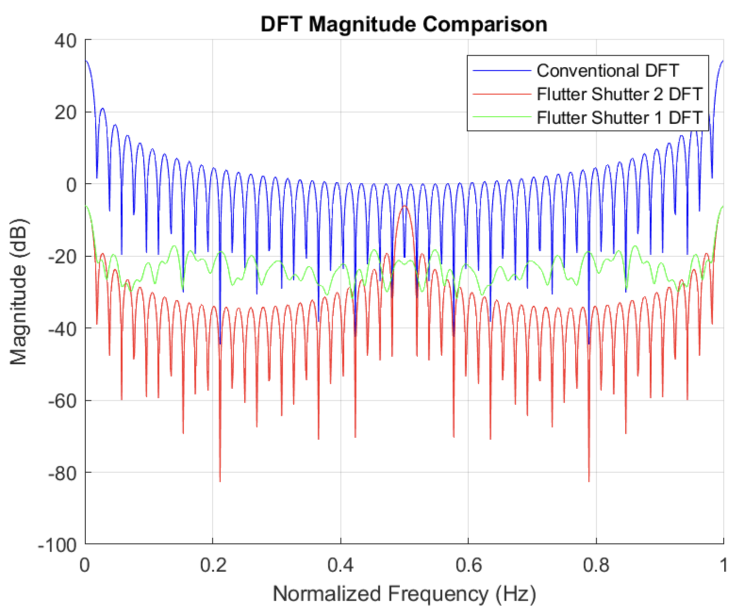

Figure 10: DFT Comparison between conventional codes and different flutter shutter codes

The blur matrix of Flutter shutter code 1 is best for motion deblurring compared to code 2 and conventional blur kernels as Dft magnitudes vary less compared to others and there are no abrupt zeros in the Dft magnitude graphs which if present affects the Condition number of matrix and ability of deblurring using Lsqr by multiplying the Noise by a lot.

2.4 Use A to deblur the blurred image using least squares. Display the deblurred images

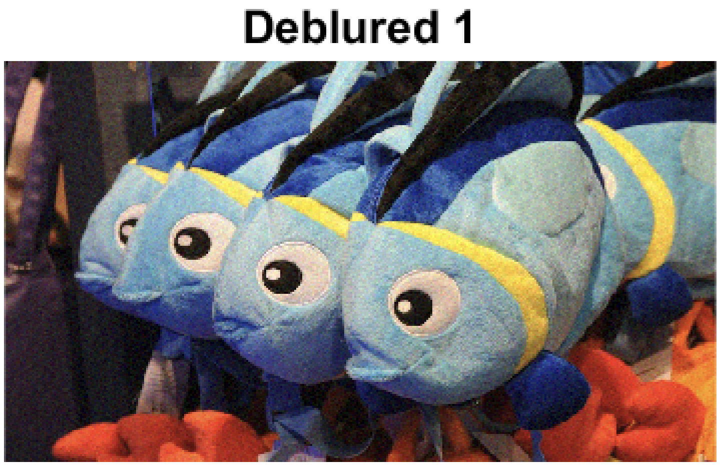

Figure 11: Deblurred Image using code 1

[ht]

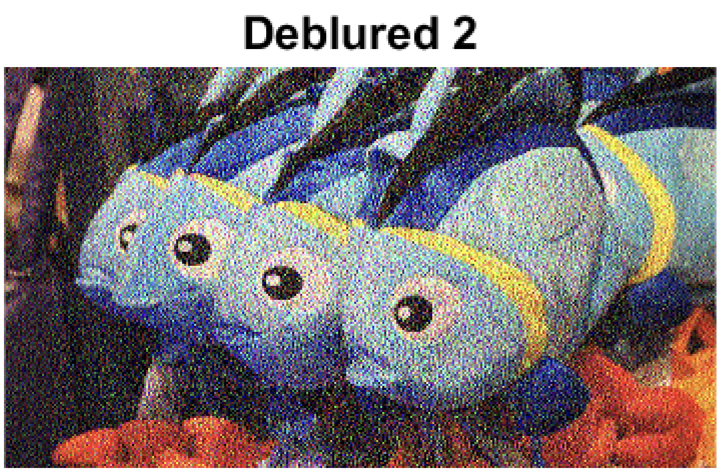

Figure 12: Deblurred Image using code 2

2.5 Using the formula given in part c of Question 1, find the RMSE for the deblurred image compared to the original for both the flutter shutter codes. Comment on your result and explain why there is a difference.

Root mean square error between clean image and deblured image is

```
RMSE = 0.4889
RMSE 1 = 0.4728
RMSE 2 = 0.6763
```

The difference in RMSE is due to Noise present in each of the deblured image is different. RMSE for code 1 is smallest as noise in deblured image using code 1 is smallest proving it is best for Motion debluring.

2.6 Let the noise be absent in both problems (1) and (2). Compare and comment on the deblured outputs

Deblured 1

Figure 13: Image without Noise deblured using code 1

Deblured 2

Figure 14: Image without Noise deblured using code 2

As there is minimal noise present in the original image. Deblurred Image regardless of blur matrix will have less noise compared to deblured image from initial image with high Noise.

3 Deblurring with motion-invariant photography

You are given images of the top view of a road scene (background.png) and a foreground moving object (redcar.png).

3.1 Assuming static camera and the velocity of the car as 1 pixel per second to the right (horizontal translation) relative to a static camera, generate the blurred image captured for an exposure time of 52 seconds. (Hint: First create individual frames for each second, i.e. 52 frames. To create each frame, apply translation corresponding to that frame on the foreground object and merge it with background. Merging: For all pixels with a non-zero values in the foreground image, assign foreground pixel values. or else use background pixel values.

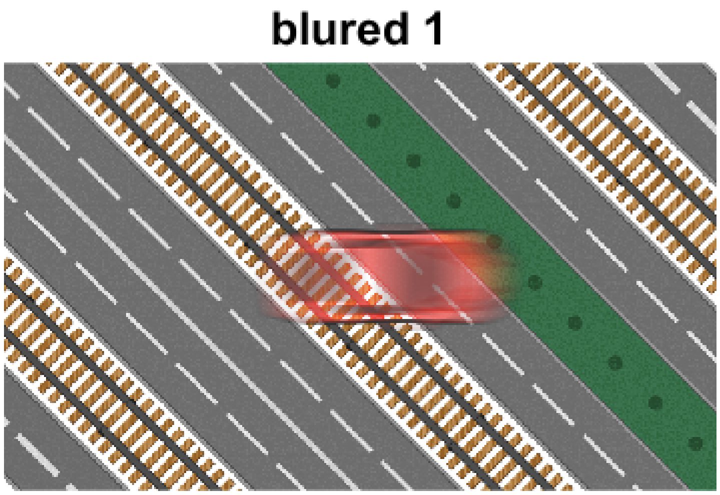

Figure 15: Foreground Motion blurred Image

3.2 Consider the same scenario captured using motion invariant photography (i.e. the camera also moves during the exposure). The translational values of the static background due to camera motion at each second are given in CameraT.mat. Generate the blurred image captured in this case with both camera and object motions for an exposure time of 52 seconds.

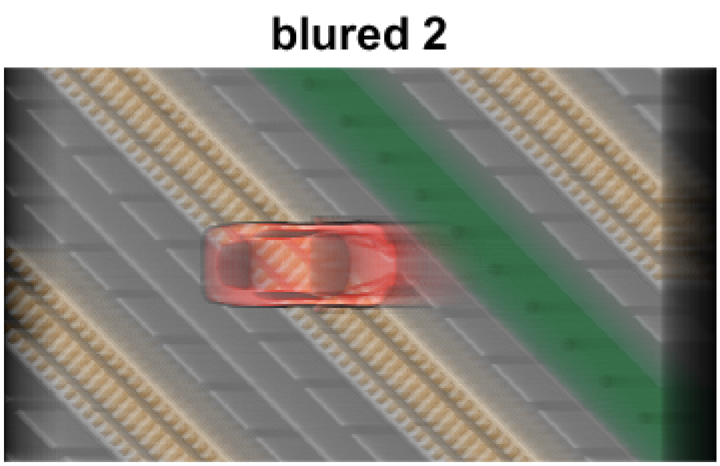

Figure 16: Motion blurred image with Uniform blur

3.3 Compare and comment on the outputs of (a) and (b)

In (a) only the foreground image object moves when back ground stas stationary. Inorder to properly deblur the image we need to segment the image try to deblur them separately which is not efficient.

In (b) ,The camera translation tries to capture every velocity in the image with same blur inorder to deblur with single kernel. It uses parabolic sweep across all velocities with equal capture time.

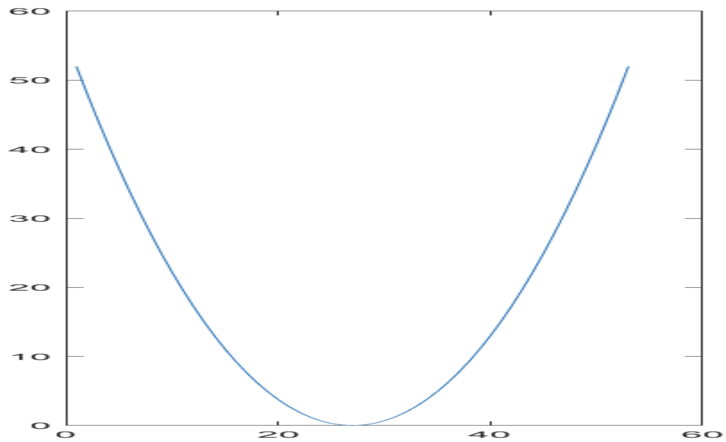

Figure 17: Camera translation

3.4 Find and plot the PSF of the blurred image in (b). If x is the camera translation then PSF weight at x is . Choose x as [0.1,1,2,...,51] and . Normalize the PSF after assigning these values.

psf =

Columns 1 through 7

0.2081 0.0465 0.0380 0.0329 0.0294 0.0269 0.0249

Columns 8 through 14

0.0233 0.0219 0.0208 0.0198 0.0190 0.0183 0.0176

Columns 15 through 21

0.0170 0.0165 0.0160 0.0155 0.0151 0.0147 0.0144

Columns 22 through 28

0.0140 0.0137 0.0134 0.0132 0.0129 0.0127 0.0124

Columns 29 through 35

0.0122 0.0120 0.0118 0.0116 0.0115 0.0113 0.0111

Columns 36 through 42

0.0110 0.0108 0.0107 0.0105 0.0104 0.0103 0.0102

Columns 43 through 49

0.0100 0.0099 0.0098 0.0097 0.0096 0.0095 0.0094

Columns 50 through 52

0.0093 0.0092 0.0091

Figure 18: PSF extended and lightened

3.5 Use the PSF obtained in (d) to form the blur matrix A.

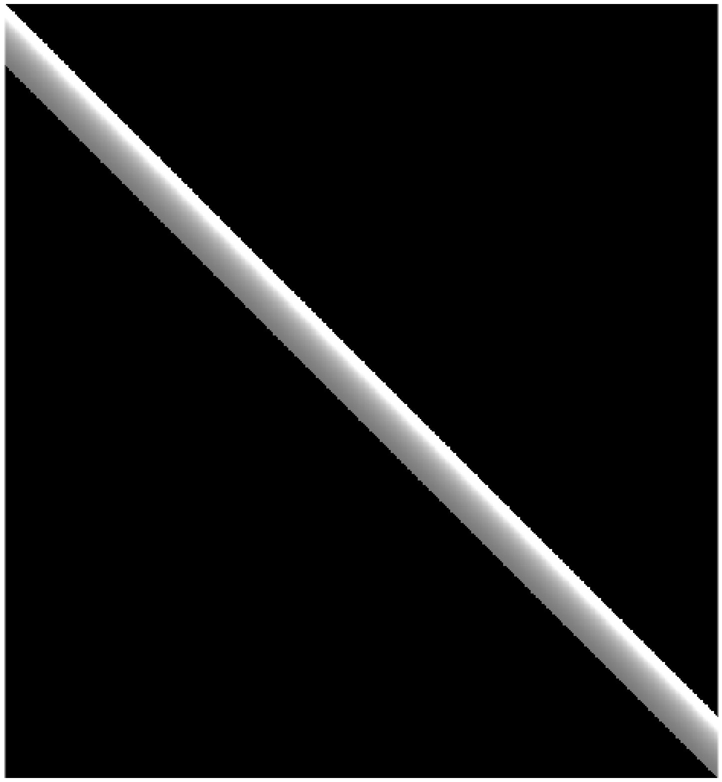

Figure 19: Blur matrix(lightened for visibility

3.6 Deblur the blurred image in (b) using least squares

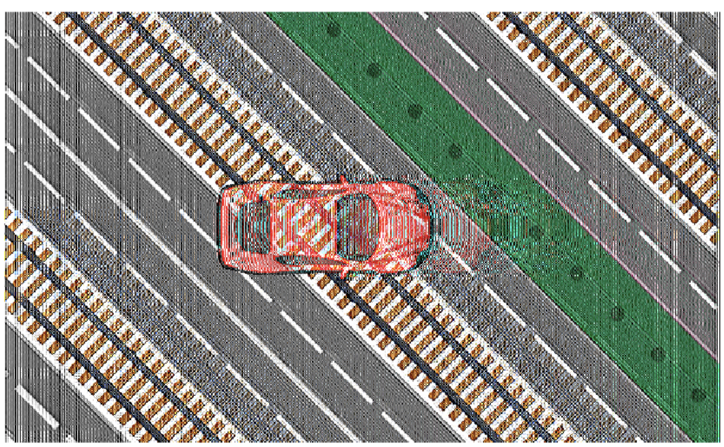

Figure 20: Deblured Image