Asisgnment 2

Name : Varun Gohil

Roll Number : 16110059

Description

In this assignment we consider an image which is corrupted by both additive Gaussian noise and defocus blur. Here I have implemented a Wiener filter to restore the image to make it less noisy and less blurry. To evaluate the restored image I use PSNR (Peak Signal to Noise Ratio).

Dataset

The dataset used for taken from the link below:

http://decsai.ugr.es/cvg/CG/base.htm

The datset has 48 images. Each image is of size 512×512. Also each image is in grayscale.

Splitting of dataset

For training we use the first 35 images of the dataset i.e. 1.gif, 2.gif, . . ., 35.gif

The rest of the images i.e 36.gif, 37.gif, . . . , 48.gif are then used for testing.

I use the PNSR (Peak Signal to Noise Ratio) metric for comparing the noised and restored image.

If the Wiener filter is working correctly then the PSNR value of the restored image will be greater than PSNR value of the corrupted image.

The results for the all the test images can be seen in the Results section.

Code Explanation

$$\begin{aligned} G(f) &= \frac{1}{H(f)}\left[\frac{|H(f)|^2}{|H(f)|^2 + \frac{N(f)}{S(f)}}\right] \\ &= \frac{1}{H(f)}\left[\frac{|H(f)|^2}{|H(f)|^2 + \frac{1}{\mathrm{SNR}(f)}}\right] \end{aligned}$$

The code implements the Wiener filter using the formula shown above. The value of H(f) is computed by taking the Fourier transform of the blur_kernel, which is already known to us. To calculate the 1/SNR term we need the original un-corrupted image.

In the code the functionn *train_for_SNR()* trains on 35 images and computes the value of 1/SNR for each of the training images. After having computed the 1/SNR values for all the training images, their average is taken. This average value is used in the Wiener filter for denoising the test images. The average value of 1/SNR is the *invSNR_trained* variable returned by *train_for_SNR()*.

This invSNR_trained is passed to *apply_wiener_filter()* and is used to compute $G(f)$ while denoising the test images.

The function *imnoise()* is used to add Gaussian noise to the image.

A Gaussian kernel is made using *fspecial()* function. The kernel is convolved with the image to add blur using the *imfilter()* function.

Results

Variance of noise = 0.01

Blur : (5×5 ones matrix) $\times$ 1/25

Image	PSNR of Noised and Blurred Image	PSNR of Restored Image
36.gif	18.6233	20.6399
37.gif	19.2141	23.9831
38.gif	19.4176	24.1731
39.gif	19.7031	25.0508
40.gif	19.7641	26.9490
41.gif	19.0843	21.9473
42.gif	18.7143	20.6669
43.gif	19.4226	21.0257
44.gif	15.3659	15.4257
45.gif	19.1847	21.0157
46.gif	18.5082	19.9983
47.gif	17.6040	19.6125
48.gif	19.3138	22.0191

As the PSNR value for restored image is greater than PSNR value of corrupted image the Wiener filter is working correctly and denoising the corrupted images.

Now I compare the PSNR values by varying the variance of the noise.

Image = 40.gif

Blur : $(5 \times 5$ ones matrix$) \times$ 1/25

Variance	PSNR of Noised and Blurred Image	PSNR of Restored Image
1	7.2387	20.1631
0.1	11.0041	25.3522
0.01	19.7641	26.9490
0.005	22.5506	26.9851
0.002	25.9133	26.9036

As variance decreases the output performance of Wiener filter is decreasing, as difference between PSNR values of corrupted and restored images is decreasing. It seems that the Wiener filter performs well on noises with high variances.

Now I compare the PSNR values by varying the blur.

Image = 40.gif

Variance of noise = 0.01

Here (a, a), a means $(a \times a$ ones matrix$) \times 1/a^2$

Blur	PSNR of Noised and Blurred Image	PSNR of Restored Image
(2,2), 2	19.8555	28.6808
(3, 3), 3	19.9052	28.6443
(4,4), 4	19.7970	26.8957
(5, 5), 5	19.7641	26.9490
(6, 6), 6	19.6705	25.5912
(10, 10), 10	19.4694	23.9286
(100, 100), 100	17.8300	18.2148

As can be seen the Wiener filter restores the image decently well even on high blurs.

Example

Image : 38.gif

Original Image

Noisd and Blurred Image

Restored Image

References

1. Wiener Filtering ,http://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj2/wiener.html
2. A True Wiener Filter Implementation for Improving Signal to Noise and Resolution in Acoustic Images, K.W. Mitchell and R.S. Gilmore, https://lib.dr.iastate.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=3004&context=qnde
3. MATLAB Documentation, https://in.mathworks.com/help/matlab/