Computer Vision and Pattern Recognition - Assignment 1

Ufuk DOGAN

2019/04/10

1 Question 1

According to the shift theorem, when we have a delay in the time domain, it corresponds to a linear phase term in the frequency domain. So, delaying a signal $x(t)$ by τ seconds multiplies its Fourier Transform by $e^{-j\omega\tau}$

Let's assume $X(\omega)$ is the Fourier Transform of $X(t)$;

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

then the transform of a shifted time version of $X(t)$ will be

$$\begin{aligned} x(t-\tau) \Longleftrightarrow \int_{-\infty}^{\infty} x(t-\tau)e^{-j\omega t}dt = \int_{-\infty}^{\infty} x(t)e^{-j\omega(t+\tau)}dt = e^{-j\omega\tau}\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \\ = e^{-j\omega\tau}X(\omega) \end{aligned}$$

As we can see, magnitude is still same, only phase changed.

2 Question 2

- Part A
  I implemented the forward and inverse transform inside the `Assignment1_Question_2_Part_A` script. Also, my error is;

$$0.0018$$

- Part B
  I implemented the forward and inverse transform inside the `Assignment1_Question_2_Part_B` script. Also, my error is;

$$1.8665e-32$$

- Part C

  We should add zero padding to the image and also the filter in order to make the size;

$$(M_f + M_h - 1) \times (N_f + N_h - 1)$$

  Description of the parameters are below;

  - M_f equals to the number of the columns of the filter
  - M_h equals to the number of the columns of the image
  - N_f equals to the number of the rows of the filter
  - N_h equals to the number of the rows of the image

- Part D

  If we do not do the padding, we will have errors in both sides and also filter will be overlap. Adding zero paddings avoid contamination in the filtering process when realized in the spectral domain.

3 Question 3

- Part A
  I implemented the forward and inverse transform inside the `Assignment1_Question_2_Part_A` script.

Figure 1: Low Pass Filtered Lena

Figure 2: Original Lena

- Part B
  A direct implementation of bilateral filtering might require O(n^2) time, where n is the number of pixels in the image. So, in high dimension pictures this algorithm will work slowly, that's why we should make it faster. To do this, we can use Piecewise-linear bilateral filtering which is a convolution such as Gaussian filtering can be greatly accelerated using Fast Fourier Transform. By this way, we can decrease our complexity to O($n * log(n)$).

  However, we can not use this directly to the bilateral filtering, the reason of this is not a convolution. That's why we should create a convolution using discretization.

  The pseudo code is below:

PiecewiseBilateral
(Image I, spatial kernel f_{σ_s}, intensity influence g_{σ_r})
J=0 /* *set the output to zero* */
for j=0..NB_SEGMENTS
i^j= minI+j $\times$ (max(I)-min(I))/NB_SEGMENTS
G^j=g_{σ_r} (I - i^j) /* evaluate g_{σ_r} at each pixel */
K^j=$G^j \otimes f_{\sigma_s}$ /* normalization factor */
H^j=G^j $\times$ I /* compute H for each pixel */
H^{*j}=$H^j \otimes f_{\sigma_s}$
J^j=H^{*j}/K^j /* normalize */
J=J+J^j $\times$ InterpolationWeight(I, i^j)

Figure 3: Pseudo code of the piecewise-linear acceleration of bi- lateral filtering

- Part C

Figure 4: Bilateral Filtered Lena

- Part D

  We can say that when we use higher values for sigma r and sigma d our output will be more blurred, more flat than the previous output. Also, as we can see when we do the sigma r 1000 the picture color seems more gray, there is no contrast, variation as before.

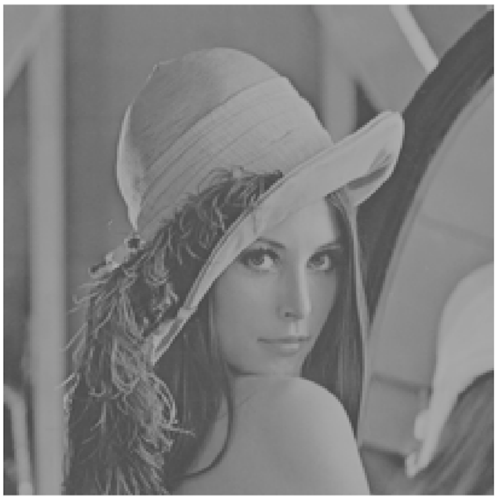

Figure 5: Bilateral Filtered Lena, parameters, sigma r = 50, sigma d = 1000

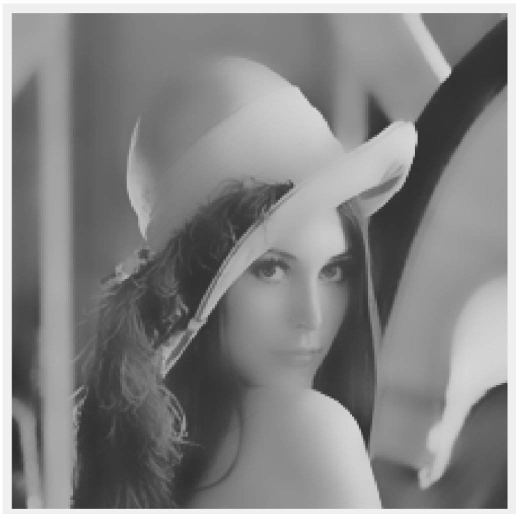

Figure 6: Bilateral Filtered Lena, parameters, sigma r = 50, sigma d = 2