EE5175: Image Signal Processing

Non-blind deblurring using gradient regularization

In this experiment, we will perform non-blind deblurring (NBD) using L_2 and L_1 norm based gradient regularization schemes. Let f, h, and g be the clean image, blur kernel and the blurred image respectively and let $\mathbf{f}$, $\mathbf{h}$, and $\mathbf{g}$ denote the lexicographically arranged column vector forms of f, h, and g.

- To perform NBD using L_2 norm based gradient regularization, we will solve the constrained least squares optimization problem of the following form.

$$\hat{f} = \arg\min_{f}\{||h * f - g||_2^2 + \lambda\big(||q_x * f||_2^2 + ||q_y * f||_2^2\big)\} \tag{1}$$

or equivalently

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}}\{||H\mathbf{f} - \mathbf{g}||_2^2 + \lambda\big(||Q_x\mathbf{f}||_2^2 + ||Q_y\mathbf{f}||_2^2\big)\} \tag{2}$$

where the first term is used to enforce the data constraint and the second one is the regularization term. H, Q_x and Q_y are doubly block circulant matrices formed using the blur kernel h, and the gradient operators q_x = [1 -1] and $q_y = q_x^T$ (H, Q_x and Q_y are used to express the underlying 2D convolution operations in the form of matrix-vector multiplication). By differentiating the cost function in Eq. 2 with respect to $\mathbf{f}$ and by equating the differential to zero one can arrive at the following form of closed form solution to the problem in Eq. 2.

$$\hat{\mathbf{f}} = (H^TH + \lambda Q_x^TQ_x + \lambda Q_y^TQ_y)^{-1}H^T\mathbf{g} \tag{3}$$

By taking Fourier transform of the above expression (along with the assumptions as discussed in the class), one can arrive at the following equivalent expression in frequency domain.

$$\hat{f} = IDFT\left(\frac{\mathbf{H}^*}{\mathbf{H}^*\mathbf{H} + \lambda\mathbf{Q}_x^*\mathbf{Q}_x + \lambda\mathbf{Q}_y^*\mathbf{Q}_y}\mathbf{G}\right) \tag{4}$$

where $\mathbf{H}$, $\mathbf{Q}_x$, $\mathbf{Q}_y$, and $\mathbf{G}$ are the 2D Fourier transforms (of the same size as that of g) of h, q_x, q_y, and g. Also, in Eq. 4 all the multiplications and divisions involved are element-wise multiplications and divisions.

- NBD using L_1 norm based gradient regularization is done by solving the constrained least squares optimization problem of the following form.

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{ ||H\mathbf{f} - \mathbf{g}||_2^2 + \lambda \big(||Q_x \mathbf{f}||_1 + ||Q_y \mathbf{f}||_1 \big) \} \tag{5}$$

where the regularization term enforces L_1 norm constraint on the gradients of restored image.

Q1 L_2 regularized NBD: PSNR vs Visual comparison - For the given image `lena.png`, perform NBD using L_2 gradient regularization (using Eq. 4) for the following scenarios (σ_n - Gaussian noise σ, σ_b - Gaussian blur σ) :

- $\sigma_n = 8$, a) $\sigma_b = 0.5$, b) $\sigma_b = 1.0$, c) $\sigma_b = 1.5$ 0.15, 0.2, 0.45
- $\sigma_b = 1.0$, a) $\sigma_n = 5$, b) $\sigma_n = 10$, c) $\sigma_n = 15$ 0.09, 0.4, 0.9

For each case, vary λ from 0.01 to 2.0 in steps of 0.01 and pick the λ that gives minimum RMS error between the original image and the estimated image. Also, for each case, find the λ that gives visually most appealing restored image (for this experiment vary λ according to your convenience). Comment on your observations.

Q2 L_2 Vs L_1 regularization - For the given image `lena.png`, perform NBD using L_2 gradient regularization (using Eq. 4) and compare the results with that of L_1 gradient regularization. To obtain the result for L_1 gradient regularization, we need to solve Eq. 5. Use the code 'admmfft.m' to solve the optimization problem in Eq. 5. Find the outputs from L_2 and L_1 regularization which is visually most appealing (by varying λ) for the following scenarios.

- $\sigma_n = 1$, $\sigma_b = 1.5$
- $\sigma_n = 5$, $\sigma_b = 1.5$
- $\sigma_n = 5$, h = `mb-kernel.png` (motion blur)

L1: lambda 0.00005 too noisy, .0001 best, 0.0005 is too pastel/smooth looking. lambda 0.000001 gives some weird speckled appearance. L2: lambda 0.005 best. 0.01 a bit smooth. 0.001 a bit speckled. Change much less drastic then L1 case

Note: The code 'admmfft.m' takes g,h, and λ as the input arguments (call the function as admmfft(g,h,λ,1)) and return the restored image corresponding to Eq. 5. While solving Eq. 4, to compute **H** (Fourier transform of the kernel), use matlab built-in function `psf2otf`.

–end–

L1 much more sensitive to lambda compared to L2

L1: 0.003 best. 0.001 visible noise. 0.005 pastel. 0.0001 has weird speckles. L2: 0.2 best. 0.3 a bit too smooth. 0.1 a bit too noisy.

L1: Prominent speckles at lambda 0.001. Best at 0.002. Higher looks pastel. L2: Best looking at 0.07. Noisy at 0.05. Fine till 0.1. Too soft after 0.2