CS3570 Introduction to Multimedia

Homework #1

Due: 11:59pm, 2019/3/28

1. DCT image compression (30%)

Transform the image **cat1.png** from spatial domain to frequency domain with DCT, and reconstruct the image using inverse DCT for three different cases with reduced numbers of DCT coefficients.

- Divide the image into blocks of 8×8 pixels and apply 2D DCT for each block.
- For each block, only keep the lower-frequency (i.e. upper-left n-by-n) coefficients in the 2D DCT domain by setting the remaining coefficients to zero.
- Reconstruct the image by taking inverse 2D DCT with the modified DCT coefficients for each block.

(a) Implement the simplified DCT compression process above for n = 2, 4, and 8 and apply it to the attached image. Show the reconstructed images for these three different cases. [3 images] Compute the PSNR values of the three reconstructed images and discuss what the PSNR value means here.

(b) Use the same process in (a) with image transformed to YIQ color model and show the reconstructed image in RGB space. [3 images] Compute the PSNR values of the three reconstructed images and discuss what the PSNR value means here.

(c) Compare the differences between the results in two color spaces in (a) and (b).

Note:

You should not use the MATLAB built-in functions, such as dctmtx, dct, dct2, idct,

idct2, rgb2ntsc, ntsc2rgb, psnr, etc.

2. Dithering (30%)

Convert the image **cat2_gray.png** to binary (black and white) image with different methods of dithering, show the results, and make some comparison with the results.

(a) Apply noise (random) dithering on the provided image and show the result. [1 image]

(b) Apply average dithering on the provided image and show the result. [1 image]

(c) Apply error diffusion dithering (Floyd-Steinberg algorithm) on the provided image and show the result. [1 image]

(d) Compare the differences between the results of the three types of dithering.

Note:

You should not use the MATLAB built-in functions, such as dither, im2bw, imbinarize, etc.

3. Image Convolution (40%)

Implement the image convolution by Gaussian filter for image smoothing. Assume that there are zero-valued pixels around the edges to ensure the same-size image after image convolution. The following Matlab code creates a Gaussian filter with hsize = [3 3] and sigma = 1. *G = fspecial('gaussian', [3 3], 1);*

(a) Creates the Gaussian filter with hsize = 3x3, 5x5 and 7x7. Apply image convolution to image **cat3_LR.png** by three Gaussian filters, and compute the PSNR with the original image. [3 image]

(b) Creates the Gaussian filter with sigma = 1, 5 and 10. Apply image convolution to image **cat3_LR.png** by three Gaussian filters, and compute the PSNR with the original image. [3 image]

(c) Compare and discuss the results from the above three methods and give the meaning of PSNR values to these results.

Note:

You should not use the MATLAB built-in functions, such as imfilter, conv2, filter2, convn, psnr, etc.

Reminder

- MATLAB built-in functions listed in problem description are prohibited.
- Your code should work correctly and the generated results (display or output files) must be consistent to your results in report.
- Report format can be in Word, PowerPoint or others that can clearly describe your work and results. You should convert your report to a PDF file.
- Your report should contain at least how you implement the methods and discussion about the output results.
- **Pack {student_ID}_report.pdf, the output result images, and codes in {student_ID}.zip. Your package should also contain a README file about how to execute your program.**