EE6310 Image and Video Processing, Spring 2024

Indian Institute of Technology Hyderabad
Homework 5, Assigned 07.04.2024, Due **11:59 pm on 21.04.2024.**

When we think we know, we cease to learn. – Sarvepalli Radhakrishnan

Remarks:

- Do **not** use built-in functions.
- Please use the *lighthouse.png* image for the first two questions.
- Use the images *first_frame.png* and *second_frame.png* posted along with this HW.
- Divide each frame into *non-overlapping macroblocks* of size 16×16 pixels. Note that the images are of size 176×144.
- Please turn in Python Notebooks with the following notation for the file name: `your-roll-number`-hw5.ipynb.

1 Optimal Decorrelating Transform

1. Implement principal component analysis (PCA) to decorrelate an image. Divide the image into non-overlapping 8×8 patches and vectorize the patches into 64-dimensional vectors. Find the covariance matrix using all the image patches. Ensure that the data is zero-mean (feature-wise). Reconstruct the image at three different "compression" levels of 48, 32 and 16 dimensions and display it. Again, be sure to add the mean back. Is the PCA effective in decorrelating images? (10)

2 Discrete Cosine Transform

Write a program to implement a JPEG-like image encoder using the following steps. (10)
Encoder:

1. Divide the image into non-overlapping blocks of 8×8 pixels.
2. Apply the Type-II DCT to each of the blocks (defined in the slides).
3. Divide each block by the quantization matrix. Look up the quantization matrix Q in the slides. Verify that you are left with sparse coefficients at this point.

Decoder:

1. Multiply each block with the inverse Q matrix.
2. Apply the Type-II IDCT to each block.
3. Put the sub-blocks together to generate the decoded estimate of the original image.

Display the original and "lossy-compressed" images. Is the DCT effective for image compression?

3 Motion Estimation

In this problem you will implement the most critical part of the video codec – the motion estimator. Do the following:

1. Use the 3-step search to find motion vectors (check slides for description). (10)
   - Use mean absolute distance (MAD) as your metric.
   - Step 1: Search at 8 location ± 4 pixels around current macroblock including (0, 0) (relative to current macroblock).
   - Step 2: search at 8 location ± 2 pixels around best match location in Step 1 including best match location.
   - Step 3: search at 8 location ± 1 pixels around best match location in Step 2 including best match location.
2. Plot the motion vector at each macroblock. You can use the *arrow* function in *matplotlib*. (1)
3. Generate the motion compensated predicted frame using the motion vectors and the first frame. (3)
4. Compute the error between the second frame and its motion compensated predicted version and display it. (1)