Lab 3: Intra- and Inter-Frame Coding

Introduction

In this lab, we investigate some of the core principles of coding image frames (used in both static images or video). We will not implement all aspects of JPEG encoding (which are commonly used within video encoders). Instead, we focus on examining two core ideas: namely, intra-frame (within frame) coding using 2D block-based DCT; and inter-frame (between frame) coding using differential coding (DPCM).

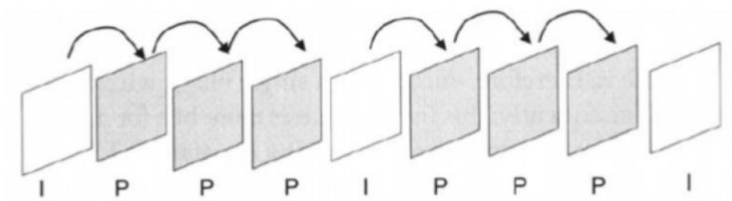


Figure 1: Intra-coded (I) and Inter-coded (P) frames

Intra-Frame for Image Data

In intra-frame coding for image data, the idea is to encode pixel information (within) a given frame. The JPEG pipeline is of course, one example of intra-frame coding, as it operates entirely within a single image frame only, working to exploit BOTH pixel-pixel spatial and spectral redundancies.

The Discrete Cosine Transform (DCT) evaluates SPECTRAL properties of a signal. In images, this is done by exploring the degree to which a 2D signal f(x,y) is correlated to cosine transitions of different spatial frequencies (u,v) in both the vertical and horizontal directions respectively. The DCT equation computes the $(u,v)^{th}$ entry of the DCT as:

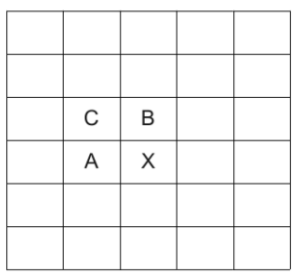
$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \cos\frac{(2x+1)\,u\pi}{2M} \cos\frac{(2y+1)\,v\pi}{2N} f(x,y) \qquad (1)$$

$$where: C(\xi) = \{ \begin{array}{ll} \frac{\sqrt{2}}{2} & if \xi = 0, \\ 1 & otherwise. \end{array} \qquad (2)$$

In JPEG, the DCT is applied to fixed sized blocks within an image. I.e. the image is subdivided into 8x8,16x16 or 32x32 sized blocks of pixels, and a 2D DCT is applied to each block, to form DCT coefficients. The DC component F(0,0) for each of these blocks is typically encoded using DPCM.

DPCM exploits SPATIAL redundancies, and in JPEG is applied to DC coefficients from each block only, the same process can be used on raw pixel values however (i.e. before and without any application of DCT at all). When DCT and Quantization is not applied, such DPCM is considered to be a lossless technique.

DPCM for images is usually applied in 2D fashion (as opposed to DPCM used in audio, which is 1D). In the 2D case, there are many options for configuring which samples from an image should be used as predictors. In the 1D case, the predictor is typically the previous sample. The current sample is then encoded as a difference between its true (original) value, and the predictor (previous sample). In the 2D case, we do not have a previous sample, but rather, several adjacent samples: e.g. A, B, C from Figure 2. The table included in Figure 2 indicates alternative modes than can be used as predictors when encoding pixels. Essentially though, a given pixel is encoded as a difference between its original pixel value, and the chosen predictor value. So, if we use mode 1, each pixel X will be predicted by the pixel to it's immediate left X' = A, thus the differential D = X - X'.



Prediction Index	Prediction
0	No prediction
1	A
2	B
3	C
4	A + B – C
5	A + ((B – C)/2)
6	B + ((A – C)/2)
7	(A + B)/2

Figure 2: DPCM modes for encoding pixels

<u>Inter-Frame for Video Data</u>

The most basic form of inter-frame encoding can be achieved by simply taking the difference between two consecutive image frames in a video sequence (i.e. using each co-located pixel in the first frame as a predictor for the corresponding pixel in the next frame). In this sense, the predictor for X in frame n+1 can be thought of as $X'_{n+1} = X_n$. With the difference at that pixel $D = X_n - X'_{n+1}$ Similarly, DPCM may be applied <u>across</u> frames, where neighbours of the co-located pixel in the previous frame are used to predict the pixel in the current frame. For e.g. $X'_{n+1} = A_n + B_n - C_n$ (for mode 4).

Frame n is thought of as a reference frame (typically intra-frame encoded), referred to as an I-frame, and frame n+1 is encoded by such inter-frame differences. Frame n+1 must therefore be predicted from frame n during decoding, and is called a P-frame. All P-frames are dependent and thus require the previous frame to be decoded first, before they can be decoded. P-frames typically yield very low overall entropy because frames are not expected to differ all that much within a video sequence (particularly of similar scenes).

Laboratory Tasks:

Part I: Intra-Frame Coding (DCT)

1. Write a function to apply an MxM DCT to an MxM image array (X). You may assume the image is single channel, and you may use the built-in OpenCV function cv2.dct() or MATLAB function: *dct2*().

2. Using a sample image (grayscale or Y channel) of your choice, crop/select a 4x4, 8X8, and 32x32 area from the image as your input for X, and output the 2D DCT for each case. Create figures (images) of your input sample, and DCT coefficient outputs. Justify the coefficient distribution you observe for each. Note: To display DCT output, you will have to scale with *log*(*abs* (*F* (u,v))).

3. For the same three 4X4, 8X8, and 32x32 cases, perform the inverse DCT transform and verify that you can recover the original image signals. You can use built-in functions for inverse DCT.
4. Explore what happens when you zero out coefficients from the DC, low frequency, and high frequency parts of the DCT output, and then attempt the inverse DCT on these coefficients.

5. Write a script to divide up an entire image (assume grayscale) into 8x8 blocks, and perform the 2D DCT for each block, outputting the results into a new image (such that the new image shows DCT coefficients for each corresponding image block).

6. Using the 8x8 quantization table from lectures, quantize, round(), de-quantize these coefficients, and reconstruct the original image. Comment on the quality of the reconstruction using distortion measures: MSE, SNR and PSNR. How does the distortion change with different images?

Part II: Inter-Frame Coding (DPCM)

Convert several frames from your video sequence to grayscale (or extract the Y channel after YCbCr conversion). We will deal with encoding only a single channel.

1. Extract the first two frames from the sequence. Calculate the frame difference and display this as an image. Repeat this for several frames over the sequence, and explore how/where the differences are emphasized.

2. Calculate the entropy for the difference frame and plot this value over 10 consecutive frame differences (you can use code from lab 2 for this)

3. Assuming an ideal entropy coder was to be used, and the first frame uses no compression, what would be the compression ratio for the entire set of 10 frames? (note: this is essentially the case $X'_{n+1} = X_n$)

4. So far you have just used the previous frame as the predictor (i.e. the co-located pixel in the previous frame itself). Modify your code to allow for modes (1-4) from the image-based DPCM prediction scheme in the above table, however, in this case, the pixel locations A,B,C each now refer to positions relating to the co-located pixel Xn in the previous frame.

5. In your report, include sample images of individual adjacent frames, images of their differences, and plot histograms of the raw and DPCM coded images

Submission:

You must demo your lab to your TA during the lab session. All group members MUST be present. Submit report (PDF in IEEE format as per D2L), and source files on your D2L submission folder. Only one member per group needs to submit.

DEMO DUE: **Week of March 18**
REPORT DUE: **March 22 - 11.59 PM**

RUBRIC		**5 (pts)**	**5 (pts)**
		Code and Demo	Report