Quiz on MPEG Compression

Review questions prior to quiz.

What comes after an I frame? Maybe a P frame, maybe another I frame. Normally a P frame would occur after, but depends on if you are acquiring or transmitting.

Studio video is 601. It is just data representing Y R-Y B-Y. It is raw data compared to what is being transmitted (which has been compressed).

Digital video is used in the studio. Once the video gets to the transmitter, it will be converted to MPEG.

Diagram on page 11-58. Similar to what we did in class. Tells what order the frames are acquired and the order in which the frames are transmitted.

One MPEG picture for each field, if it is interlaced. Otherwise, one picture per frame when it is progressive.

Spatial is on the same frame (picture). Temporal refers to considering multiple frames.

I frames are just compressed spatially. P and B frames are just compressed temporally, since the previous I frame has taken care of the spatial compression. Within the P and B frames, there might be some additional compression of the inter-frame differences, but that is considered part of temporal compression.

The Quiz Itself

10 questions, 20 points.

Discrete Cosine Transform

There are many image compression techniques available but transform coding is the preferred method. Since energy distribution varies with each image, compression in the spatial domain is not easily attained. Transform coding is used to maximize compression of images in the **frequency domain**.

How to convert from amplitude domain to frequency domain in video? Can see this in RF, but pictures is more difficult to make sense of. Kind of like doing FM over AM.

Images tend to arrange their energy distribution in the frequency domain making compression in the frequency domain much more effective.

The Discrete Cosine Transform (DCT) is an example of transform coding. The highest energies are relocated to the upper left corner of the coefficient table. The lesser energies are relocated into the other areas. High frequencies have low energies, low frequencies have high energies.

DCT is fast. Can be calculated quickly and is best for images with smooth edges like photos with human subjects.

DCT is used for spatial compression.

The Inverse DCT (IDCT) can be used to retrieve the image from its transform representation.

The two-dimensional DCT is just a one-dimensional DCT applied twice, in the X direction and in the Y direction. When you apply the DCT to an 8x8 table of luminance level values, it will yield an 8x8 matrix of values corresponding to how much of the DCT function is present in the original 8x8 image.

For most 8x8 images, much of the signal energy lies at low frequencies; these appear in the upper left corner of the DCT table. The lower right values represent higher frequencies and are often smaller in value.

Shows a picture of a digitized photo, and an 8x8 pixel area within it. Shows the 8x8 picture, the sample values (IRE values) and the DCT coefficients. The DCT coefficient matrix contains mostly zeros in this example, with a few nonzero values in the upper left.

In a block of uniformly gray pixels (4x4). The DC value of the DCT output represents the average brightness (8); this appears in the upper left corner of the output matrix. This is the APL (average picture level) of that block of pixels. All other values are zero; they are based on what difference there is between that one to the one on top of it and to its left (since all pixels are uniformly gray, there is no difference, so the coefficients are zero).

Since pixels are small, there is a high likelihood of adjacent pixels being the same.

In the *leftmost* column, each pixel compared with the one *above* it. In the *other* columns, each pixel compared with the one *to the left of* it. We will see in later examples if this holds true. Doesn't look like it.

Low frequency horizontal component.

Low frequency vertical component.

The upper left corner contains generally low frequency information. The upper right corner contains high frequency vertical infomation. The lower left corner contains high frequency horizontal transition information. The lower right corner contains high frequency diagonal transition infomation.

Quantization

The 8x8 block of DCT coefficients is quantized, limiting the number of allowed values for each coefficient. **This is the first lossy compression step**.

Higher frequencies are usually quantized more coarsely (fewer values allowed) than lower frequencies. This results in many DCT coefficients being zero, especially at the higher frequencies.

Not many high frequency changes in an 8x8 block. Mostly will have subtle changes which are represented by low frequencies.

Motion could create a high frequency change.

How do we maximize the zeros? Diagram 1. Why maximize the zeros? Can compress them highly using run length encoding. If just read them out in a row-column fashion, would have to repeat many small groups of zeros, even if you run length encode them. Can read them out diagonally zig-zag (diagram 2), which results in 700, 90, 90, -89, 0, 100, and a bunch of zeros that can be **run length encoded**. Send the number of identical values (59) and the common value (0).

Almost all values are equal to zero. Because the non-zero values are concentrated at the upper left corner, the matrix is transferred to the receiver in zigzag scan order. That would result in 700 90 90 -89 0 100 .... 0. The zeros at the end are not transferred. Instead, an *End-Of-Block* sign is coded, and the default value of zero is assumed in order to fill in the matrix at the decoder.

There are more than one type of zig zag scan, and will specify which one is being used. Currently there are only two types of zig zags (at least at the time these slides were prepared). Which zig zag scan is used is specified for each block. The goal is to maximize the number of contiguous zeros so that run length encoding can compress the matrix to the smallest size.

Apply DCT, apply quantization, perform zig-zag scan to get the final compressed block. Decoder reads the zig-zag scan to recreate the quantized matrix, then scales the matrix (un-quantizes it), then applies inverse DCT. This is a lossy recreation of the original block, but it is close enough.

Using 8 bits per pixel in MPEG, unlike using 10 bits in the studio.

The DCT always tends to compute zeros. This effect is assisted by the quantization which **zeros small values**.

Apply IDCT to a matrix only containing one value of 700 at the upper left corner. Diagram 3. The picture is a gray colored square.

The value at the upper left is called the DC value. Abbreviation for direct current and refers to theory that AC can have a DC component. In DCT, the DC value determines the average brightness in the block. All other values describe the variation around this DC value. Therefore they are sometimes referred to as AC values (term coming from *alternating current*).

MPEG actually drops down to 4:2:0, while studio stays at 4:2:2. 4:2:2 supports editing, which the studio might need to do to insert commercials. End users (viewers) don't need editing capability. This is why editing computer video is so bad (not so for Avid and Final Cut Pro systems). There is a flag in the MPEG stream that indicates that this is a safe place to insert/edit; this is so that other stuff can be inserted once it is compressed.