JPEG continued and MPEG Introduction

JPEG Stages

Quantization:

JPEG utilizes a quantization table in order to quantize the results of the DCT. Remember that JPEG is a lossy compression scheme. Quantization allows us to define which elements should receive fine quantization and which elements should receive coarse quantization. Those elements that receive a finer quantization will be reconstructed close or exactly the same as the original image, while those elements that receive coarse quantization will not be reconstructed as accurately, if at all. A quantization table is an 8x8 matrix of integers that correspond to the results of the DCT. Each entry in this table is an 8-bit integer. To quantize the data, one merely divides the result of the DCT by the quantization value and keeps the integer portion of the result. Therefore, the higher the integer in the quantization table, the coarser and more compressed the result becomes. This quantization table is defined by the compression application, not by the JPEG standard. A great deal of work goes into creating "good" quantization tables that achieve both good image quality and good compression. Also, because it is not defined by the standard, this quantization table must be stored with the compressed image in order to decompress it.

Entropy Encoding:

Last time we saw that DCT converts the spatial intensity values into the frequency domain and orders the results in a zig-zag sequence. Also, we saw that the there tends to be little or no high-frequency change in an 8x8 block. Here is where we really use this to our advantage. The first step in entropy encoding is to Run-Length Encode the zero values. Because there will be many zeroes in the high-frequency results of the DCT, we can produce long run-length encoded strings of zeroes with the zig-zag sequence. As an example, consider an image that only has non-zero values for the DC and the first two AC coefficients. If we encode row-by-row, we get DC, AC1, {0,6}; AC2, {0,7};{0,8};{0,8};... If we use the zig-zag sequence, however, we get DC,AC1,AC2,{0,61}, a tremendous space savings! Another important point to remember here is that the DC coefficients are encoded as differences from the previous DC coefficient (except for the DC coefficient of the first block, which obviously cannot do...
Because there tends to be little change from block-to-block, this gives us additional space savings. The final step in entropy encoding is to Huffman code the results. The Huffman algorithm can also have a table specified by the JPEG application, or it can create its own by examining the data. In either case, this must also be written with the compressed results in order to decompress.

**JPEG Decompression**

Decompressing a JPEG image is basically applying the compression process in reverse. First, the Huffman table is read and the image is entropy decoded. Then, the quantization table is read and the result is de-quantized. The process of de-quantization will naturally produce a blocky result. In order to fix this, most applications will interpolate the pixel values to bring them closer to a smooth curve. Next, the IDCT (Inverse-DCT) will be performed on this result, and from this the image will have been reconstructed.

**Lossy JPEG**

Loss is introduced into the JPEG compression scheme in 3 ways:

1. Quantization
2. Subsampling the U and V components
3. Cosine (used in DCT) is a transcendental function. Mathematically speaking, this means \( \cos^{-1}(\cos(x)) \neq x \)

**Lossless JPEG**

Lossless JPEG eliminates the 3 sources of loss shown above. It encodes pixels in a 4x4 block, storing values for 3 of the pixels and encoding one as the difference from another pixel. This means that this algorithm has a maximum compression of around 25%.

**Multi-Resolution JPEG**

In order to deal with limited bandwidth, there are multiple resolution JPEG formats.

- **Progressive JPEG:** Progressive JPEG subsamples the original image by 4x4, and encodes the result, then subsamples the original image by 2x2 and encodes the pixel differences from the previous image, and finally encodes pixel differences between the original image and the result of the 2x2 subsample.
- **Alternative 1:** It is possible to pass the image through the JPEG stages using only the four most significant bits of the pixel values first, then pass the four least significant bits second.
- **Alternative 2:** It is also possible to pass the DC and 31 AC coefficients through the stages first, then pass the remaining 32 AC coefficients through afterward.
These multi-resolution formats are commonly used on the WWW. These allow a rough representation of the image to be drawn first, then the image is refined, and then the image is refined again into the final version.

Results of JPEG Compression

In general, the following results are seen with JPEG compression:

- Good Quality uses 0.25-0.5 bits per pixel
- 0.5-0.75 bpp gives better quality
- 0.75-1.5 bpp is excellent quality
- 1.5-2 bpp gives a result almost indistinguishable from the original.

Remembering that the original image used 24 bpp, there is a lot of savings here!

MPEG

Background of MPEG

MPEG stands for the Motion, Movie, or Moving Picture Experts Group (all 3 can be used). This is not affiliated with Hollywood; rather it is a group of computer scientists trying to make a standard for digital representation of video.

There are several flavors of MPEG:

- **MPEG-1**: MPEG-1 was the original MPEG standard, designed exclusively for computer use. It allows a 320x240, 30 frame per second video.
- **MPEG-2**: MPEG-2 is a higher-resolution version of MPEG-1 designed for digital television broadcast.
- **MPEG-3**: MPEG-3 was designed for HDTV. However, HDTV is just normal TV with a higher resolution and frame rate, so this standard was folded into MPEG-2 and is no longer used.
- **MPEG-4**: MPEG-4 is a new standard for digital video scheduled for release in November 1998. It is designed for use over low-bit-rate wireless and mobile communication systems. DCT cannot provide the required compression to operate over this type of line, so MPEG-4 does not enforce an encoding method. Instead, it leaves the choice of encoding method up to the designer.
- **MPEG-7**: MPEG-7 is yet another standard for digital video that has barely begun. The focus of MPEG-7 is supposed to be designing a representation for digital video that allows it to be stored and queried by content in a video database.
Why MPEG instead of JPEG for video?

JPEG is an algorithm designed exclusively for digital images. While MPEG does utilize JPEG to some extent, motion video has some additional properties that JPEG does not consider:

- Use and synchronization of multiple media streams, such as Video, Audio, and Closed-Captioning
- Time relationship between frames

Because video is displayed at 30 frames per second, even JPEG cannot give us the compression necessary to make digital video feasible. However, if we can exploit the relationship between successive frames (there will likely be little or no change from frame to frame), we can compress even more. How does MPEG do this?

Interframe Coding

With 30 frames per second, you will naturally expect differences between successive frames of a video sequence to be relatively small. MPEG achieves a great deal of compression by exploiting the relationship between successive frames. Rather than encoding one initial frame and then sending only differences for all the remaining frames, MPEG uses the windowing approach. Windowing breaks up the video sequence into smaller subsequences and encodes differences only within a window, not between them. This is done for two reasons,

1. Protection from errors: what if you lose a frame in transmission? It is possible that the entire remainder of the sequence could be useless without windowing.
2. Random access and Editing: How do you edit a compressed video sequence without having to decompress and then re-encode it without windowing?

Each of these windows in MPEG is called a Group of Pictures (GOP). How long is a GOP? The answer: as long as you like. The length of a GOP is not specified by the MPEG standard, and a video sequence can even contain GOPs of different lengths.

Frame Types

- I frames are intracoded frames. They don't depend on any other frames, you can think of them as JPEG images.
- P frames are predicted frames. They depend only on the previous I or P frame.
- B frames are bidirectional frames. They can depend on both the previous and the next I or P frames.

Because I and P frames are used to predict other P and B frames, they are called Reference Frames.
Motion Estimation

Motion estimation in MPEG operates on *Macroblocks*. A macroblock is a 16x16 pixel range in a frame. There are two primary types of motion estimation, forward and backward. *Forward prediction* predicts how a macroblock from the previous reference frame moves forward into the current frame. *Backward prediction* predicts how a macroblock from the next reference frame moves back into the current frame. Examples are shown below, with Forward prediction in red (left-to-right) and Backward prediction in green (right-to-left).

Motion estimation operates as follows: First, compare a macroblock of the current frame against all 16x16 regions of the frame you are predicting from. Then, select the 16x16 region with the least mean-squared error from the current macroblock and encode a *motion vector*, which specifies the 16x16 region you are predicting from and the error values for each pixel in the macroblock. This is done only for the combined Y,U, and V values. Subsampling and separation of the Y, U, and V bands comes later.

There are four types of macroblocks:

- **Forward Predicted**: (P and B only) predict from a 16x16 region in the previous reference frame
- **Backward Predicted**: (B only) predict from a 16x16 region in the next reference frame
- **Bidirectional Predicted**: (B only) predict from the average of a 16x16 region in the previous reference frame and a 16x16 region in the next reference frame.
- **Intracoded**: (I, P, or B) are not predicted, the actual pixel values are used for the macroblock.

It is important to remember that P and B frames can contain intracoded macroblocks as well as predicted macroblocks if there is no efficient way to predict the macroblock.

Decoding vs. Presentation order
MPEG is actually used in decoding order rather than presentation order. Examples of both follow:

**Presentation Order**

I₁ B₂ B₃ B₄ P₅ B₆ B₇ B₈ P₉ B₁₀ B₁₁ B₁₂ I₁₃

**Decoding Order**

I₁ P₅ B₂ B₃ B₄ P₅ B₆ B₇ B₈ I₁₃ B₁₀ B₁₁ B₁₂

The reason for the difference is that in order to decode a predicted frame, all frames that it may be predicting from must be decoded first. Therefore, since B₂..₄ may all be predicting from both I₁ and P₅, both must be decoded before B₂..₄. This distinction becomes very important when you work with MPEG.

**Independent vs. Dependent GOPs**

Independent GOPs do not depend on any frames of the previous GOP for prediction. Dependent GOPs depend on a reference frame from another GOP for prediction. Examples follow (in decoding order):

**Case 1:** GOP₁ is dependent upon GOP₂ (which starts with I₁₃)

I₁ P₅ B₂ B₃ B₄ P₅ B₆ B₇ I₁₃ B₁₀ B₁₁ B₁₂

**Case 2:** GOP₁ is not dependent upon GOP₂ (which starts with I₁₃)

I₁ P₅ B₂ B₃ B₄ P₅ B₆ B₇ P₁₂ B₁₀ B₁₁ I₁₃

To illustrate the difference, imagine trying to perform a simple edit operation and cut out GOP₂. In order to do this, I₁₃ must be removed. If this happens, B₁₀..₁₂ will not be able to be decoded since they depend on I₁₃. In the second case no frames in the first GOP depend on the second GOP, making this operation possible. As shown here, if you want to make a dependent GOP independent, end the current GOP with a P frame.

**Bandwidth**

Bandwidth is a major concern in digital video. For MPEG, you can remember the following:

- I frames require the most space, and give the least compression
- B frames require the least space, and give the most compression.
- P frames are in between.

MPEG-1 is fixed to a maximum 1.2Mbit/second bandwidth. If an encoded MPEG-1 stream is larger than this, the encoder will have to make the quantization more coarse (to
increase compression) and re-encode the sequence. This idea is called feedback, where the output of the encoder is analyzed and changes the input back into the encoder until the sequence is acceptable.

Motion Estimation and Subsampling:

In MPEG, Motion estimation is done BEFORE subsampling and separation of the Y U and V bands. This means that there is only one motion vector for a macroblock rather than one for each of the 3 bands (Y, U, and V). The results of motion estimation will then be processed similar to JPEG in the following manner:

1. Perform Motion Estimation (Result is an array of 16x16 macroblocks)
2. Break up the image into Y U and V planes
3. Subsample the U and V planes by a factor of 2 in each direction (Not 4 as in JPEG)
4. Pass the results through the DCT and remaining steps of JPEG in 8x8 blocks.

Note that due to subsampling of U and V, one 16x16 macroblock will contain 4 Y blocks, 1 U, and 1 V block.

This sequence is done for all types of frames. Although it may at first seem counterintuitive, the error matrices from motion vectors are also passed through DCT and the remaining steps of JPEG. The reason for this is that you expect little or no change in the macroblock as it moves from frame to frame. Any change that the macroblock does go through will likely be a change that will affect the entire region or low frequency gradual change from one side of the region to another. As an example, consider a macroblock that is forward predicted into a region that is covered by a shadow. The Y component of each pixel may reduce by a constant factor, and the U and V values will not change. Therefore, if you think of it in this manner, DCT will encode change that affects an entire region as the DC coefficient, and the remaining AC coefficients in this case would be near zero. This enables us to save a great deal of space.

Error Handling:

MPEG-1 is a format designed for computer use only, it is not intended for broadcast purposes. In MPEG-1, if you lose an I frame then the entire GOP is lost. If you lose a P frame, you can lose all frames until the next reference frame.