Coding and Data Compression

Lec. 6

Image Compression

Image Transforms

The mathematical concept of a transform is a powerful tool that is employed in many areas and can also serve as an approach to image compression. An image can be compressed by transforming its pixels (which are correlated) to a representation where they are decorrelated. Compression is achieved if the new values are smaller, on average, than the original ones. Lossy compression can be achieved by quantizing the transformed values. The decoder inputs the transformed values from the compressed stream and reconstructs the (precise or approximate) original data by applying the inverse transform.

The term decorrelated means that the transformed values are independent of one another. As a result, they can be encoded independently, which makes it simpler to construct a statistical model. An image has can be compressed if its representation redundancy. The redundancy in images stems from pixel correlation. If we transform the image to a representation where the pixels are decorrelated, we have eliminated the redundancy and the image has been fully compressed.

Image transforms are designed to have two properties:

(1) reduce image redundancy by reducing the sizes of most pixels, and

(2) identify the less important parts of the image by isolating the various frequencies of the image.

Thus, this section starts with a short discussion of frequencies. We intuitively associate a frequency with a wave. Water waves, sound waves, and electromagnetic waves have frequencies, but pixels in an image can also feature frequencies.

Image frequencies are important because of the following basic fact: Low frequencies correspond to the important image features, whereas high frequencies correspond to the details of the image, which are less important.

Thus, when a transform isolates the various image frequencies, pixels that correspond to high frequencies can be quantized heavily, while pixels that correspond to low frequencies should be quantized lightly or not at all. This is how a transform can compress an image very effectively by losing information, but only information associated with unimportant image details.

The Discrete Cosine Transform

This important transform (DCT for short) has been used and studied extensively since. Because of its importance for data compression, the DCT is treated here in detail.

The DCT in one dimension is given by

$$G_f = \sqrt{\frac{2}{n}} C_f \sum_{t=0}^{n-1} p_t \cos\left[\frac{(2t+1)f\pi}{2n}\right],$$

where

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0, \\ 1, & f > 0, \end{cases} \quad \text{for } f = 0, 1, \ldots, n-1.$$

The input is a set of n data values pt (pixels, audio samples, or other data), and the output is a set of n DCT transform coefficients (or weights) Gf . The first coefficient G0 is called the DC coefficient, and the rest are referred to as the AC coefficients (these terms have been inherited from electrical engineering, where they stand for “direct current” and “alternating current”). Notice that the coefficients are real numbers even if the input data consists of integers. Similarly, the coefficients may be positive or negative even if the input data consists of nonnegative numbers only. This computation is straightforward but slow. The decoder inputs the DCT coefficients in sets of n and uses the inverse DCT (IDCT) to reconstruct the original data values (also in groups of n). The IDCT in one dimension is given by

$$p_t = \sqrt{\frac{2}{n}} \sum_{j=0}^{n-1} C_j G_j \cos\left[\frac{(2t+1)j\pi}{2n}\right], \quad \text{for } t = 0, 1, \ldots, n-1.$$

In two dimension DCT is:

$$G_{ij} = \sqrt{\frac{2}{m}} \sqrt{\frac{2}{n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} p_{xy} \cos\left[\frac{(2y+1)j\pi}{2m}\right] \cos\left[\frac{(2x+1)i\pi}{2n}\right],$$

for $0 \le i \le n-1$ and $0 \le j \le m-1$ and for C_i and C_j. The first coefficient G_{00} is again termed the "DC coefficient," and the remaining coefficients are called the "AC coefficients." The image is broken up into blocks of n×m pixels p_{xy} (with n = m = 8 typically), and Equation (7.16) is used to produce a block of n×m DCT coefficients G_{ij} for each block of pixels. The coefficients are then quantized, which results in lossy but highly efficient compression. The decoder reconstructs a block of quantized data values by computing the IDCT whose definition is

$$p_{xy} = \sqrt{\frac{2}{m}}\sqrt{\frac{2}{n}}\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} C_i C_j G_{ij} \cos\left[\frac{(2x+1)i\pi}{2n}\right]\cos\left[\frac{(2y+1)j\pi}{2m}\right],$$

where

$$C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1\ , & f > 0, \end{cases}$$

for $0 \le x \le n - 1$ and $0 \le y \le m - 1$.

JPEG

JPEG is a compression method for color or grayscale still images (not videos). It does not handle bi-level (black and white) images very well. It also works best on continuous-tone images, where adjacent pixels have similar colors.

An important feature of JPEG is its use of many parameters, allowing the user to adjust the amount of the data lost (and thus also the compression ratio) over a very wide range. Often, the eye cannot see any image degradation even at compression factors of 10 or 20.

There are two operating modes, lossy (also called baseline) and lossless (which typically produces compression ratios of around 0.5). Most implementations support just the lossy mode.

The main goals of JPEG compression are the following:

1. High compression ratios, especially in cases where image quality is judged as very

good to excellent.

2. The use of many parameters, allowing knowledgeable users to experiment and achieve

the wanted compression/quality trade-off.

3. Obtaining good results with any kind of continuous-tone image, anyway of image

dimensions, color spaces, pixel aspect ratios, or other image features.

4. A sophisticated, but not too complex compression method, allowing software and

hardware implementations on many platforms.

5. Several modes of operation:

(a) A sequential mode where each image component (color) is compressed in a single scan from left-to-right, top-to-bottom.

(b) A progressive mode where the image is compressed in multiple blocks (known as “scans”).

(c) A lossless mode that is important in cases where the user decides that no pixels should be lost (compression ratio is low compared to the lossy modes);

(d) A hierarchical mode where the image is compressed at multiple resolutions allowing lower-resolution blocks to be viewed without first having to decompress the following higher-resolution blocks.

The name JPEG is an acronym that stands for Joint Photographic Experts Group. This was a joint effort by the CCITT and the ISO (the International Standards Organization) that started in June 1987 and produced the first JPEG draft proposal in 1991. The JPEG standard has proved successful and has become widely used for image compression, especially in Web pages.

The main JPEG compression steps are outlined here:

1. Color images are transformed from RGB into a luminance/chrominance color space, this step is skipped for grayscale images). The eye is sensitive to small changes in luminance. This step is optional but important because the remainder of the algorithm works on each color component separately. Without transforming the color space, none of the three color components will tolerate much loss, leading to worse compression.
2. Color images are downsampled by creating low-resolution pixels from the original ones (this step is used only when hierarchical compression is selected; it is always skipped for grayscale images). The downsampling is not done for the luminance component.
3. The pixels of each color component are organized in groups of 8×8 pixels called data units, and each data unit is compressed separately. If the number of image rows or columns is not a multiple of 8, the bottom row and the rightmost column are duplicated as many times as necessary.
4. The discrete cosine transform is then applied to each data unit to create an 8×8 map of frequency components.
5. Each of the 64 frequency components in a data unit is divided by a separate number called its quantization coefficient (QC), and then rounded to an integer. This is where information is irretrievably lost. Large QCs cause more loss, so the high frequency components typically have larger QCs.

6. The 64 quantized frequency coefficients (which are now integers) of each data unit are encoded using a combination of RLE and Huffman coding. An arithmetic coding variant known as the QM coder can optionally be used instead of Huffman coding.
7. The last step adds headers and all the required JPEG parameters, and outputs the result. The compressed file may be in one of three formats (1) the interchange format, in which the file contains the compressed image and all the tables needed by the decoder (mostly quantization tables and tables of Huffman codes), (2) the abbreviated format for compressed image data, where the file contains the compressed image and may contain no tables (or just a few tables), and (3) the abbreviated format for table-specification data, where the file contains just tables, and no compressed image. The second format makes sense in cases where the same encoder/decoder pair is used, and they have the same tables built in. The third format is used in cases where many images have been compressed by the same encoder, using the same tables. When those images need to be decompressed, they are sent to a decoder preceded by one file with table specification data.

The JPEG decoder performs the reverse steps. (Thus, JPEG is a symmetric compression method.)

JPEG 2000

This section was originally written in mid-2000 and was slightly improved in early 2003. The data compression field is very active, with new approaches, ideas, and techniques being developed and implemented all the time.

JPEG is widely used for image compression but is not perfect. The use of the DCT on 8×8 blocks of pixels results sometimes in a reconstructed image that has a blocky appearance (especially when the JPEG parameters are set for much loss of information).

This is why the JPEG committee has decided, as early as 1995, to develop a new, wavelet-based standard for the compression of still images, to be known as JPEG 2000 (or JPEG Y2K). Perhaps the most important milestone in the development of JPEG 2000 occurred in December 1999, when the JPEG committee met in Maui, Hawaii and approved the first committee draft of Part 1 of the JPEG 2000 standard.

At its Rochester meeting in August 2000, the JPEG committee approved the final draft of this International Standard. In December 2000 this draft was finally accepted as a full International Standard by the ISO and ITU-T.

Following is a list of areas where this new standard is expected to improve on existing methods:

- High compression efficiency. Bitrates of less than 0.25 bpp are expected for highly detailed grayscale images.
- The ability to handle large images, up to 232 ×232 pixels (the original JPEG can handle images of up to 216×216).
- Easy, fast access to various points in the compressed stream.
- The decoder can pan/zoom the image while decompressing only parts of it.
- The decoder can rotate and crop the image while decompressing it.
- Error resilience. Error-correcting codes can be included in the compressed stream, to improve transmission reliability in noisy environments.

H.W: How does JPEG 2000 work?