Information Technology
Inside and Outside
- David Cyganski & John A. Orr

IV. Data Compression

8. Image Compression

Hoon-Jae Lee
hjlee@dongseo.ac.kr
http://cg.dongseo.ac.kr/~hjlee

8. Image Compression

- Objectives:
  - a review of the amount of data storage needed for images;
  - the distinction between lossless and lossy image compression;
  - basic techniques for lossless image compression;
  - an explanation of lossy compression, and the many tradeoffs it encompasses; and
  - a nonmathematical introduction to the most popular lossy compression technique: JPEG coding.

8.1 Introduction / 8.2 Image-Specific Compression Methods

8.1 Introduction

- Image data is amenable to specialized compression methods that do not actually preserve the original message

8.2 Image-Specific Compression Methods

- 1. **Imagery data has more redundancy** than we can generally find in other types of data. For example, a pair of adjacent rows of picture elements in an image are usually nearly identical, while two adjacent lines of text in a book have essentially no commonality. ➔ "***loss less coding techniques***"
- 2. As we saw in preceding chapters, **the human eye is very tolerant of approximation error in an image**. We may decide to exploit this tolerance to produce increased compression, at the expense of image quality (that is, we may intentionally introduce some error into the image). This approach would never be applied to financial data, for example, but for imagery the information loss may be irrelevant. ➔ "***lossy coding techniques***"

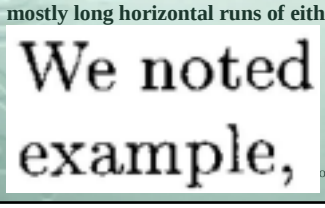
8.2 Image -Specific Compression Methods (2)/8.3 Lossless Image Compression

8.2 Image-Specific Compression Methods(2)

- More efficient storage ➔ minimize the size of the resulting data
- *Faster processing of the data* ➔ *minimize the computational time*
- Possibility: that by trading off some of the quality of the image we might obtain significantly reduced data size

8.3 Lossless Image Compression

Figure 8.1: This is a close-up view of an image of the first paragraph this section at fax resolutions. Note that the image clearly consists of **mostly long horizontal runs of either black or white pixels**.



➔ **"Run length coding"**

8.3 Lossless Image Compression(2)

8.3.1 Run Length Encoding

- a **loss less** compression technique
- **[Idea]** if a long series of pixels are identical, we can extract considerable compression simply by **sending a special code that represents the entire string of pixels**.
- **[Ex]** Counting the number of men and the number of women exiting a certain building
  - ``**man, man, man, woman, woman, woman, woman, man, woman,...,**''
  - ➔ "**Three men, four women, man, woman,....**"

8.3.1.1 PCX Format

- Zsoft Co., PC Paintbrush series
- Encode using **the special "flag" byte** that contains the number of identical pixels followed by **a byte** with the color value of those pixels.
- Idea based on that all 256 colors do not used.

8.3 Lossless Image Compression(3)

8.3.2 The JPEG Standard for Lossless Compression

- **JPEG** = ``**Joint Photographic Experts Group**"
- For natural images is **2:1, or a 50% reduction** in image data
- 1. (Seven) **prediction methods** available in the loss less JPEG coding standards. ➔ divided into the following categories:
  - 1) Predict the next pixel on a line in the image as having the same value as the previous pixel on that line.
  - 2) Predict the next pixel on the line as having the same value as the pixel on the line directly above it.
  - 3) Predict the next pixel on the line as having a value related to the values of three nearby pixels, such as the average of the three nearby pixel values.
- 2. (Next step) **comparing** predicted value to the actual pixel value.
  - The difference, if any, forms that output stream of information to be sent to the next encoder step.
  - If these differences are large, nothing is gained by this prediction process.

8.3 Lossless Image Compression(4) / 8.4 Virtual Lab Demonstrations of Lossless Compression

8.3.3 GIF: Another Lossless Image Compression System

- **GIF** stands for **Graphics Interchange Format**
- Developed by **CompuServe**
- the GIF compression to be much faster than would be a direct application of the LZ algorithm

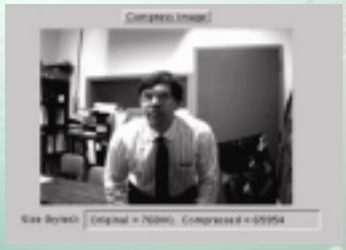
8.4.1 Run Length Encoding Java Applet

- The run length encoding (RLE) algorithm simply looks for long runs of the same pixel value and encodes it into a single copy of the value and a number representing the number of times that value repeats
- an original image with a size of 76,800 bytes is reduced to a file of 65,954 bytes after compression. That is, a compression ratio of only 1.16:1 has been obtained. Thus, while run length encoding does reduce the file, the impact is not huge for this image.

8.4 Virtual Lab Demonstrations of Lossless Compression(2)

8.4.1 Run Length Encoding Java Applet

- **Professor.gif ➔ 69594/76800=0.906**



8.4 Virtual Lab Demonstrations of Lossless Compression(3)

8.4.2 Predictive Filter Preprocessing Java Applet

- **Glass.gif : 45516/56726=0.802**
- **Glass.gif (Prefiltered) 45024/56726=0.793**

8.5 Lossy Compression

8.5.1 Simple Lossy Compression Methods

- **Glass.gif + Median filtering ➔ 42351/56726=0.747**

8.5 Lossy Compression(2)

8.5.1 Simple Lossy Compression Methods(2)

- **tree.gif + Median filtering ➔ 75502/76800=0.983**

8.5 Lossy Compression(3)

8.5.2 The JPEG Standard for Lossy Compression

- The **JPEG standard** was the result of years of effort by the **Joint Photographic Experts Group**, which was formed as a joint effort by two large, standing, standards organizations,the CCITT (European Telecommunications Standards Organization) and the ISO (International Standards Organization).
- The JPEG lossy compression algorithm :
  - **an image simplification stage**, which removes image complexity at some loss of fidelity,
  - followed by a loss less compression step based on **predictive filtering** and **Huffman or arithmetic coding**

8.5 Lossy Compression(4)

8.5.2.1 Optional Discussion: JPEG Details

- **DCT (Discrete Cosine Transform)**
- In the JPEG image reduction process, the DCT is applied to **8 x 8 pixel blocks** of the image.
- Hence, if the image is **256 x 256** pixels in size, we break it into **32 x 32 square blocks** of **8 x 8 pixels** and treat each block independently ➔ The **64 pixel values** in each block are transformed by the DCT into a new set of 64 values.
- The 64 pixel values in each block are **transformed by the DCT** into a new set of 64 values ➔ "**the DCT weights"**
  ➔ all **64 images** with appropriate **scaling of the intensities** of each one.
- **a few ``weight" values** instead of all the pixels in the original picture and still obtain a reasonably **high fidelity version of the original picture on reconstruction**.

8.5 Lossy Compression(5)

8.5.2.1 Optional Discussion: JPEG Details(2)

- **DCT (Discrete Cosine Transform) (2) :** features
  - It is **fast to compute** compared to many other suitable image transformations,
  - It usually tends to result in **low values for the weights** of some of its basis images.
  - The image that results from deleting low weight components is **generally pleasing to the eye**.
- **JPEG lossy compression algorithm** with the DCT transform
  - 1). The **smallest weights** are **eliminated**, by setting them to zero.
  - 2). The **remaining weights** are **rounded off** so that they may be represented with few bits. The amount of rounding, and hence then umber of bits transmitted, varies among all the DCT weights,according to observed levels of sensitivity of viewers to these degradations.

8.5 Lossy Compression(6)

8.5.2.1 Optional Discussion: JPEG Details(3)

- **JPEG lossy compression algorithm** (Cont'd)
  - 3). One of the underlying images for which we obtain a DCT weight value is actually a *flat* image of constant intensity. The weight forth is particular image is just proportional to the average value oft hat block of pixels. This particular weight tends to vary slowly from block to block, so prediction of this value from surrounding blocks works well. We **replace** each such weight with **the difference from the previous weight**.
  - 4). The coefficients for each block now tend to have many zeros and similar values. **Run length encoding** is then applied to the whole set.
  - 5). Finally, **Huffman or Arithmetic (entropy or universal) coding** is applied to the resulting data.

8.5 Lossy Compression(6)

8.5.2.2 The Performance of JPEG Compression

- Typically quoted performance for JPEG is that photographic quality images of **natural scenes can be preserved** despite the lossiness of the method, with **compression ratios** of up to about **20:1 or 25:1**. ➔ Usable quality (that is, for non critical purposes) can result for **compression ratios to 100:1, or more in special cases**.

Figure 8.3: The test image file **compressed via JPEG loss less compression to 252,906 bytes. ➔ Ratio= 252906/921600 =0.2744**

8.5 Lossy Compression(7)

8.5.2.2 The Performance of JPEG Compression(2)

Figure 8.4: The test image file compressed via **JPEG lossy compression to 33,479 bytes**. This is the first level of lossy compression at which change in the picture from the original can be detected by a person.➔ **Ratio = 33479/921600=0.0363**

17

8.5 Lossy Compression(8)

8.5.2.2 The Performance of JPEG Compression(3)

Figure 8.5: The test image file compressed via JPEG lossy compression to 22,418 bytes.
➔ **Ratio = 22418/921600=0.0243**

Information Technology 18

8.5 Lossy Compression(9)

8.5.2.2 The Performance of JPEG Compression(4)

Figure 8.6: The test image file compressed via JPEG lossy compression to 14,192 bytes. At this level of compression, the **distortion of the image** becomes quite **noticeable.**

➔ **Ratio = 14192/921600=0.0154**

8.5 Lossy Compression(10)

8.5.2.2 The Performance of JPEG Compression(5)

Figure 8.7: The test image file compressed via JPEG lossy compression to 8,978 bytes. The image is essentially **unusable** at this compression level.

➔ **Ratio = 8978/921600=0.0097**