Image compression fundamentals

by Gleb V. Tcheslavski: gleb@ee.lamar.edu
http://ee.lamar.edu/gleb/dip/index.htm

Preliminaries

Image compression – reducing or eliminating of redundant or irrelevant information – is one of the most used techniques in the field of image processing. Considering a two-hour standard definition (SD) TV movie that we need to store digitally… A digital movie (video) is a sequence of video frames, which are full-color digital still images. The standard rate for an NTSC video is close to 30 fps (frames per second), therefore, the bit rate required for SD video is

$$30 \cdot (720 \cdot 480) \cdot 3 = 31,104,000 \text{ bytes/sec}$$

And a storage requirement for a two-hour movie would be

$$31,104,000 \cdot 60 \cdot 60 \cdot 2 \approx 2.24 \cdot 10^{11} \text{ bytes}$$
Preliminaries

Approximately 224 GB is needed to store an uncompressed two-hour SD movie. Therefore, to fit such movie on a standard DVD-9, data must be compressed by a factor of approximately 26.3. The compression must be higher for high definition (HD) TV, where image resolution is up to 1920x1080 pixels.

Similarly, an 8-MP digital camera could store about 41 uncompressed full-color images (roughly 24 MS each) on a 1 GB memory card.

Compression can also significantly reduce transmission time needed to transmit an image over the web…

Fundamentals

Data compression is a process of reducing the amount of data required to represent a given quantity of information. Note that data and information are not the same; data are the means by which information is conveyed. Since various amounts of data can be used to represent the same amount of information, representations that irrelevant or repeated information are said to contain redundant data. Denoting the number of bits in two representations of the same information as \( b \) and \( b' \), the relative data redundancy of the representation with \( b \) bits is

\[
R = 1 - \frac{1}{C}
\]

Where \( C \) is the compression ratio:

\[
C = \frac{b}{b'}
\]
Fundamentals

If $C = 10$, for instance, the larger representation has 10 bits of data for every 1 bit of data in the smaller representation. The corresponding relative data redundancy of the larger representation is $R = 0.9$ indicating that 90% of its data is redundant.

$b$ is the number of bits needed to represent an image as a 2D array of intensity values. We notice that such arrays being preferred formats for human viewing, are not optimal from a compact data representation viewpoint. Such arrays have three types of data redundancy:

1. **Coding redundancy**: intensities represented by 8 bits contain more data than needed;
2. **Spatial and temporal redundancy**: each pixel is similar to or dependent on (correlated) neighboring pixels. In video, pixels are correlated temporally. Therefore, information is replicated.

3. **Irrelevant information**: most 2D intensity arrays contain information that is ignored by the human visual system and/or extraneous to the intended use of image. It is redundant in the sense that it is not used.

Compression is achieved when one or more of these redundancies is reduced or eliminated.
Coding redundancy

Assume that a discrete random variable $r_k$ in the interval $[0, L-1]$ is used to represent the intensities of an $M \times N$ image and that probability of occurrence of each $r_k$ is $p_r(r_k)$. Then

$$p_r(r_k) \approx \frac{n_k}{M \times N} \quad k = 0, 1, 2, \ldots, L - 1$$

Here $L$ is the number of intensity values and $n_k$ is the number of times that the $k$th intensity appears in the image. If the number of bits needed to represent each value of $r_k$ is $l(r_k)$, the average number of bits required to represent each pixel is

$$L_{\text{avg}} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

The total number of bits required to represent an $M \times N$ image is $MNL_{\text{avg}}$. If the intensities are represented using a natural $m$-bit fixed-length code (each event is assigned to one of $2^m$ codes from an $m$-bit binary sequence), $L_{\text{avg}} = m$. The constant $m$ can be taken outside of the summation leaving the sum of $p_r(r_k)$, which equals 1.

<table>
<thead>
<tr>
<th>$r_k$</th>
<th>$p_r(r_k)$</th>
<th>Code 1</th>
<th>$l_1(r_k)$</th>
<th>Code 2</th>
<th>$l_2(r_k)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r_{67}$ = 87</td>
<td>0.25</td>
<td>01010111</td>
<td>8</td>
<td>01</td>
<td>2</td>
</tr>
<tr>
<td>$r_{128}$ = 128</td>
<td>0.47</td>
<td>10000000</td>
<td>8</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$r_{186}$ = 186</td>
<td>0.25</td>
<td>11001000</td>
<td>8</td>
<td>000</td>
<td>3</td>
</tr>
<tr>
<td>$r_{255}$ = 255</td>
<td>0.03</td>
<td>11111111</td>
<td>8</td>
<td>001</td>
<td>3</td>
</tr>
<tr>
<td>$r_k$ for $k \neq 87, 128, 186, 255$</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The “start” image has the intensity distributions shown above. “code 1” – natural 8-bit code: $l_1(r_k) = 8$ bits for all $r_k$. 

Spring 2008  
ELEN 4304/5365 DIP
Coding redundancy

On the other hand, if the “code 2” is used, the average length of the encoded pixels is

\[ L_{\text{avg}} = 0.25 \cdot 2 + 0.47 \cdot 1 + 0.25 \cdot 3 + 0.03 \cdot 3 = 1.81 \text{ bits} \]

The total number of bits needed to represent the entire image is

\[ M_{N L_{\text{avg}}} = 256 \cdot 256 \cdot 1.81 = 118,621 \text{ bits} \]

As compared to the number of bits needed if 8-bit “code1” was used:

\[ M_{N L_{\text{avg}}} = 256 \cdot 256 \cdot 8 = 524,288 \text{ bits} \]

Therefore, the resulting compression is

\[ C = \frac{524,288}{118,621} = \frac{8}{1.81} \approx 4.42 \]

Coding redundancy

And the data redundancy is

\[ R = 1 - \frac{1}{4.42} = 0.774 \]

suggesting that 77.4% of the data in the original 8-bit image is redundant.

The compression achieved by “code 2” results from assigning fewer bits to the more probable intensity values and more bits to the less probable ones. The result is a variable-length code.

Coding redundancy is present when the codes assigned to events (such as intensity values) do not take full advantage of the probability of events.
Spatial and temporal redundancy

A histogram for the “strip” image shows that:

1. All 256 intensities are equally probable;
2. Since the intensity of each line was selected randomly, its pixels are independent of one another in the vertical direction.
3. Since the pixels along each line are identical, they are maximally correlated in the horizontal direction.

The first observation indicates that the image cannot be compressed by variable-length code along. On the other hand, observations 2 and 3 reveal a significant spatial redundancy that can be eliminated, for instance, by representing the image as a sequence of run-length pairs.

Each run-length pair specifies the start of new intensity and the number of consecutive pixels having that intensity. A run-length based representation compresses the original 2D 8-bit intensity array by

$$\frac{256 \cdot 256 \cdot 8}{(256 + 256) \cdot 8} = \frac{128}{1}$$

Each 256-pixel line of the image is replaced by a single 8-bit intensity value and length 256 in the run-length representation.

In most images, pixels are correlated spatially and in time, therefore, information carried by individual pixels is small. To reduce this redundancy, a 2D intensity array must be transformed into more efficient representation, such as run-lengths or differences between adjacent pixels.
Irrelevant information

One of the simplest ways to compress a set of data is to remove superfluous data. For images, information that is ignored by human visual system or is extraneous to the intended use of an image are obvious candidates for omission.

The “gray” image, since it appears as a homogeneous field of gray, can be represented by its average intensity alone – a single 8-bit value. Therefore, the compression would be

\[
\frac{256 \cdot 256 \cdot 8}{8} = 65,536:1
\]

Irrelevant information

However, the histogram of the “gray” image reveals that there are several intensity values that we just perceive as a single averaged value!

A histogram equalized version of the image reveals its “invisible” structure. Therefore, representing the image by its average value would – in this case – lead to loss of information.
Measuring image information

An important question to answer is "How many bits are necessary to represent the information in an image?" or alternatively, "Is there a minimum amount of data that is sufficient to describe an image without loss of information?" The information theory states that generation of information can be modeled as a probabilistic process.

A random event $E$ with probability $P(E)$ contains

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

units of information. For true events ($P(E) = 1$), $I(E) = 0$ and such events do not contain information. Therefore, information is a measure of uncertainty.

Measuring image information

The base of the logarithm determines units used to measure information. If the base $m$ logarithm is used, the measurements are in $m$-ary units. If the base 2 is selected, the unit of information is the bit.

When $P(E) = \frac{1}{2}$, $I(E) = -\log_2 \frac{1}{2} = 1$ bit. Therefore, 1 bit is conveyed when one of two possible equally likely events occurs.

For a source of statistically independent random events $\{a_1, a_2, ... a_J\}$ with associated probabilities $\{P(a_1), P(a_2), ... P(a_J)\}$, the average information (entropy of the source) per source output is

$$H = -\sum_{j=1}^{J} P(a_j) \log P(a_j)$$

Here $a_j$ are source symbols. Since they are statistically independent, the source is called a zero-memory source.
Measuring image information

If we consider an image as an output of an imaginary zero-memory “intensity source”, the histogram of the image can be used to estimate the symbol probabilities of source. The intensity source’s entropy is

\[ H = - \sum_{k=0}^{L-1} p_{r_k} \log_2 p_{r_k} \]

It is NOT possible to code the intensity values of the imaginary source (and thus the sample image) with fewer than \( H \) bits/pixel.

For instance, the entropy of the “start” image can be estimated as

\[ H = -[0.25 \log_2 0.25 + 0.47 \log_2 0.47 + 0.25 \log_2 0.25 + 0.03 \log_2 0.03] \approx 1.6614 \text{ bits/pixel} \]

Similarly, the entropies of the “strip” and “gray” images can be computed as 8 bits/pixel and 1.566 bits/pixel respectively.

Although the “star” image appears to have the most visual information, it has almost the lowest entropy. The “strip” image has almost 5 times the entropy of the “star” image but about the same (or less) visual information.

Therefore, we conclude that the amount of entropy (and, therefore, the information) in the image is not quite intuitive.
Fidelity criteria

We have previously seen that the removal of “irrelevant visual” information may lead to a loss of real or quantitative image information. Two types of criteria can be used to quantify a loss of information:

1) Objective fidelity criteria (math expression is used);
2) Subjective fidelity criteria.

When information loss can be expressed as a mathematical function of the input and output of the compression process, it is based on an objective fidelity criterion. For instance, a root-mean-square (rms) error between two images.

Fidelity criteria

Let $f(x, y)$ be an input image and $\hat{f}(x, y)$ be an approximation of $f(x, y)$ resulting from compressing and decompressing the input image. For any values of $x$ and $y$, the error is

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

Therefore, the total error between the two images is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[ \hat{f}(x, y) - f(x, y) \right]$$

And the rms error is

$$e_{rms} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[ \hat{f}(x, y) - f(x, y) \right]^2}$$
Fidelity criteria

We can also consider \( \hat{f}(x, y) \) as the sum of the original image \( f(x, y) \) and an error (noise) signal \( e(x, y) \), the mean-square signal-to-noise ratio of the output image is defined as

\[
SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}
\]

The rms value of SNR is

\[
SNR_{rms} = \sqrt{SNR_{ms}}
\]

Fidelity criteria

While objective fidelity criteria offer a simple and convenient way to estimate information loss, images are viewed by humans. Therefore, measuring image quality by subjective evaluations of people is often more appropriate: show two images (original and decompressed) to a number of viewers and average their evaluations.

One possible absolute rating scale:

<table>
<thead>
<tr>
<th>Value</th>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Excellent</td>
<td>An image of extremely high quality, as good as you could desire.</td>
</tr>
<tr>
<td>2</td>
<td>Fine</td>
<td>An image of high quality, providing enjoyable viewing. Interference is not objectionable.</td>
</tr>
<tr>
<td>3</td>
<td>Passable</td>
<td>An image of acceptable quality. Interference is not objectionable.</td>
</tr>
<tr>
<td>4</td>
<td>Marginal</td>
<td>An image of poor quality, you wish you could improve it. Interference is somewhat objectionable.</td>
</tr>
<tr>
<td>5</td>
<td>Inferior</td>
<td>A very poor image, but you could watch it. Objectionable interference is definitely present.</td>
</tr>
<tr>
<td>6</td>
<td>Unusable</td>
<td>An image so bad that you could not watch it.</td>
</tr>
</tbody>
</table>
Fidelity criteria

Three approximations to the “star” image with rms errors computed:

\[ e_{rms} = 5.17 \]
\[ e_{rms} = 15.67 \quad \text{Blurred edges} \]
\[ e_{rms} = 14.17 \quad \text{Missing parts, artifacts} \]

Subjective evaluations lead to the following ratings:

*excellent*
*passable - marginal*
*inferior - unusable*

Image compression models

The *encoder* performs compression, the *decoder* performs decompression. Both operations can be done in software (web browsers, image viewers) or in a combination of hardware and firmware (DVD players, digital cameras). A *codec* is a device or program that is capable for both encoding and decoding.
Encoding (compression) process

The **mapper** transforms an image (or their sequence) into a usually invisible format designed to reduce spatial and temporal redundancy. Usually, this operation is reversible and may or may not reduce the amount of data needed to represent the image. Run-length coding is an example of mapping. In video applications, the mapper uses previous (and future) frames to facilitate removal of temporal redundancy.

The **quantizer** reduces the accuracy of the mapper’s output according to the fidelity criterion. This operation is irreversible and targets to remove irrelevant information from the image. When error-free compression is needed, this procedure must be omitted.

Encoding (compression) process

The final stage of the encoding process, the **symbol coder**, generates a fixed- or variable-length code to represent the quantizer output according to the code. Usually, to minimize coding redundancy, the shortest code words are assigned to the most frequently occurring quantizer output values. This operation is reversible.

These three operations leads to removal (decreasing) of all three redundancies from the input image.

The decoder contains two components: **symbol decoder** and an **inverse mapper** performing the inverse operations of the encoder’s symbol coder and mapper. The inverse quantizer block is not included since quantization is irreversible.
Image formats, containers, and compression standards

An image file format is a standard way to organize and store image data. It defines how the data is arranged and the type of compression (if any) to be used.

An image container is similar to file format but handles different types of image data.

An image compression standard defines procedures for compressing and decompressing images (as needed for reducing the amount of data needed to represent an image).

Some popular image compression standards, file formats, containers: International standards are shown in black, all others are gray.
Popular image compression standards

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCITT</td>
<td>ITU-T</td>
<td>A simplified and streamlined version of the CCITT Group 3 standard supporting 2-D run-length coding only.</td>
</tr>
<tr>
<td>JPEG</td>
<td>ISO/IEC</td>
<td>A Joint Photographic Experts Group standard for progressive, lossless compression of bi-level images. Continuously-toned images of up to 8 bits/pixel can be coded on a bit-plane basis [8.27]. Context-sensitive arithmetic coding [8.23] is used and an initial low resolution version of the image can be gradually enhanced with additional compressed data.</td>
</tr>
<tr>
<td>JPEG-LS</td>
<td>ISO/IEC</td>
<td>A lossless to near-lossless standard for continuous tone images based on adaptive prediction [8.29], context modeling [8.25], and Golomb coding [8.22].</td>
</tr>
<tr>
<td>JPEG-2000</td>
<td>ISO/IEC</td>
<td>A follow-on to JPEG for increased compression of photographic quality images. Arithmetic coding [8.23] and quantized discrete wavelet transforms (DWT) [9.23] are used. The compression can be lossy or lossless.</td>
</tr>
</tbody>
</table>

### Video

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DV</td>
<td>IEC</td>
<td>Digital Video. A video standard tailored to home and professional video production applications and equipment—like electronic new gathering and camcorders. Frames are compressed independently for uncomplicated coding using a DCT-based approach [8.26] similar to JPEG.</td>
</tr>
<tr>
<td>H.261</td>
<td>ITU-T</td>
<td>A two-way videoconferencing standard for ISDN (integrated services digital network) lines. It supports non-interlaced 352 × 288 and 176 × 144 resolution images called CIF (Common intermediate Format) and QCIF (Quarter CIF), respectively. A DCT-based compression approach [8.22] similar to JPEG is used, with frame-to-frame prediction differencing [8.2-9] to reduce temporal redundancy. A block-based technique is used to compensate for motion between frames.</td>
</tr>
<tr>
<td>H.262</td>
<td>ITU-T</td>
<td>See MPEG-2 below.</td>
</tr>
<tr>
<td>H.263</td>
<td>ITU-T</td>
<td>An enhanced version of H.261 designed for ordinary telephone modems (i.e., 28.8 Kbps) with additional resolutions QCIF (Sub-Quarter CIF 128 × 96), ACIF (76 × 576), and HICIF (140 × 512).</td>
</tr>
<tr>
<td>H.264</td>
<td>ITU-T</td>
<td>An extension of H.261-H.262 for videofonering, Internet streaming, and television broadcasting. It supports prediction differences within frames [8.2-9], variable block size integer transforms (rather than the DCT), and context adaptive arithmetic coding [8.23].</td>
</tr>
<tr>
<td>MPEG-1</td>
<td>ISO/IEC</td>
<td>A Motion Picture Expert Group standard for CD-ROM applications with non-interlaced video at up to 1.5 Mbps. It is similar to H.261 but frame predictions can be based on the previous frame, next frame, or an interpolation of both. It is supported by almost all computers and DVD players.</td>
</tr>
<tr>
<td>MPEG-2</td>
<td>ISO/IEC</td>
<td>An extension of MPEG-1 designed for DVDs with transfer rates in 15 Mbps. Supports interlaced video and HDTV. It is the most successful video standard to date.</td>
</tr>
<tr>
<td>AVC</td>
<td>ISO/IEC</td>
<td>MPEG-4 Part 10 Advanced Video Coding (AVC), identical to H.264 above.</td>
</tr>
<tr>
<td>BIFS</td>
<td>ISO/IEC</td>
<td>A follow-on to JPEG for bi-level images in desktops, Internet, and fax applications. The compression method used is correct based, with dictionary based methods [8.26] for text and halftone regions, and Huffman [8.21] or arithmetic coding [8.23] for other image content. It can be lossy or lossless.</td>
</tr>
</tbody>
</table>

### Continuous Tone Still Images

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP</td>
<td>Microsoft</td>
<td>Windows Bitmap. A file format used mainly for simple, uncompressed images.</td>
</tr>
<tr>
<td>GIF</td>
<td>Compuserve</td>
<td>Graphic Interchange Format. A file format that uses lossless LZW coding [8.25] for 1-channel through 8-bit images. It is frequently used to make small animations and short low resolution films for the World Wide Web.</td>
</tr>
<tr>
<td>PDF</td>
<td>Adobe Systems</td>
<td>Portable Document Format. A format for representing PDF documents in a device and resolution independent way. It can function as a container for JPEG (JPEG, PDF, TIFF, and other compressed images). Some PDF versions have become ISO standards.</td>
</tr>
<tr>
<td>PNG</td>
<td>World Wide Web consortium (W3C)</td>
<td>Portable Network Graphics. A file format that losslessly compresses full color images with transparency (up to 8 bits/pixel) by coding the differences between each pixel's value and a predicted value based on past pixels [8.29].</td>
</tr>
<tr>
<td>TIFF</td>
<td>Adobe</td>
<td>Tagged Image File Format. A flexible file format supporting a variety of image compression standards, including JPEG (JPEG-LA, JPEG-2000, H.262, and others).</td>
</tr>
<tr>
<td>AVS</td>
<td>MII</td>
<td>Audio Video Standard. Similar to H.264 but uses exponential Golomb coding [8.22]. Developed in China.</td>
</tr>
<tr>
<td>HDV</td>
<td>Company</td>
<td>High Definition Video: An extension of DV for HD television that uses MPEG-2-like compression including temporal redundancy removal by prediction differencing [8.29].</td>
</tr>
<tr>
<td>JPEG-2000</td>
<td>Various companies</td>
<td>Motion JPEG. A compression format in which each frame is compressed independently using JPEG.</td>
</tr>
<tr>
<td>QuickTime</td>
<td>Apple Computer</td>
<td>A media container supporting DV, H.261, H.262, H.264, MPEG-1, MPEG-2, MPEG-4, and other video compression formats.</td>
</tr>
<tr>
<td>VC-1</td>
<td>SMPTE</td>
<td>The most used video format on the Internet.</td>
</tr>
<tr>
<td>WMA</td>
<td>Microsoft</td>
<td>Widescreen Media Audio. Adopted for HD and Blu-ray high-definition DVDs. It is similar to H.264-AVC, using an image DCT with quarter-block sizes [8.24] and [8.23] and context dependent variable-length code tables [8.21] but no predictions within frames.</td>
</tr>
</tbody>
</table>