Image Compression.

A. Transform Image Coding.

1. The Transform Step

   a. Apply an invertible transform T.

   b. T *decorrelates* the data, i.e., removes redundancy or hidden structure.

   c. Usually T is an orthogonal transformation.

   d. This step is *lossless*.

2. The Quantization Step.

   a. Output of T are high-precision floating-point numbers so require many bits to store.

   b. Quantization essentially rounds off these numbers so that they require fewer bits to store.

   c. This step is *lossy* and all error occurs at this stage.

3. The Coding Step.

   a. If T does a good job then most of the transformed coefficients will be close to zero. Quantization actually sets them to zero.

   b. Output of Step 2. is a bit-stream containing long stretches of zeros.

   c. Such bit-streams can be coded efficiently.

B. Scalar Quantization.

1. A quantization function $Q(x)$ is a step function whose range is the integers and such that the inverse image of each integer n is an interval.

2. The dequantizing function, Q^{-1}, defined on the integers, maps n to the midpoint of the inverse image of n under Q.

C. Coding.

Definition 1 *A* symbol source *is a finite set* $S = \{s_1, s_2, \ldots, s_q\}$ *together with associated* probabilities *given by* $p_i = P(s_i)$ *for* $1 \leq i \leq q$. *Here* $0 \leq p_i \leq 1$ *and* $\sum p_i = 1$.

A binary code*,* C*, is a finite set of finite length strings of* 0*'s and* 1*'s. Each element of* C *is called a* codeword*. A* coding scheme *is a one-to-one mapping* f *from* S *into* C*. The* average codeword length *of* f *is given by*

$$ACL(f) = p_1\, len(f(s_1)) + p_2\, len(f(s_2)) + \cdots + p_q\, len(f(s_q)).$$

Examples.

(a) Let $S = \{A,\ B,\ C,\ D\}$, and let $P(A) = 5/8$, $P(B) = 3/16$, $P(C) = 1/16$, and $P(D) = 1/8$. Consider the code $C = \{00,\ 01,\ 10,\ 11\}$ and the coding scheme

$$\begin{array}{lcl} A & \longrightarrow & 00, \\ B & \longrightarrow & 01, \\ C & \longrightarrow & 10, \\ D & \longrightarrow & 11. \end{array}$$

The average codeword length for this coding scheme is

$$5/8{\cdot}len(00){+}3/16{\cdot}len(01){+}1/16{\cdot}len(10){+}1/8{\cdot}len(11) = 5/8{\cdot}2{+}3/16{\cdot}2{+}1/16{\cdot}2{+}1/8{\cdot}2 = 2.$$

(b) Let's consider a different coding scheme.

$$\begin{array}{lcl} A & \longrightarrow & 0, \\ B & \longrightarrow & 10, \\ C & \longrightarrow & 111, \\ D & \longrightarrow & 110. \end{array}$$

The ACL for this coding scheme is

$$5/8 \cdot 1 + 3/16 \cdot 2 + 1/16 \cdot 3 + 1/8 \cdot 3 = 25/16 = 1.5625.$$

This scheme will be about $1.5625/2 = .78125$ or about 22% more efficient.

The Prefix Property.

Definition 2 *A binary coding scheme f has the* prefix property *if no codeword appears as the prefix of any other codeword.*

(a) *This property guarantees that every string of codewords can be uniquely deciphered*

(b) *Moreover it guarantees that each codeword can be deciphered* as soon as it is read.

Entropy.

Definition 3 *The* entropy *of a symbol source S is defined by*

$$H(S) = -\sum_{i=1}^{q} P(s_i)\ \log_2(P(s_i)).$$

Intuitively, $H(S)$ measures the amount of uncertainty or information in the source. The more "uncertain" a particular outcome, the more "information" is contained in the outcome.

Entropy has a number of natural properties sufficient to uniquely define it.

(a) A symbol source S for which $P(s_i) = 1$ for some i and $P(s_j) = 0$ for $j \neq i$ has no uncertainty, and the average amount of information in each output is zero.

(b) The source with the most uncertainty is one in which each symbol is equally likely.

(c) Adding symbols to a source that has no chance of occurring does not change the amount of uncertainty or the average amount of information in the source.

(d) If a pair of independent sources are putting out symbols simultaneously, then the information in the paired source is the sum of the information in each source separately. Specifically, given sources $A = \{a_1, \ldots, a_q\}$ and $B = \{b_1, \ldots, b_r\}$, define a new source

$$AB = \{a_i b_j\}_{1 \leq i \leq q; 1 \leq j \leq r}$$

with $P(a_i b_j) = P(a_i)P(b_j)$. Then

$$H(AB) = H(A) + H(B).$$

Coding and Compression.

(a) Given a symbol source S with $q = 2^s$ symbols, suppose that we are trying to code an output of that source of length M, where M is large. We call this data output a message, or a data stream, or a bit stream, or an image.

(b) Since each symbol requires s bits, we can represent the data using sM bits. By coding efficiently we want to reduce the number of bits representing each symbol.

(c) for a coding scheme f, and if M is large, we expect to be able to represent each symbol with $ACL(f)$ bits on average, so that the entire message can be represented with $ACL(f) \cdot M$ bits.

(d) In the context of image compression, we say that using the coding scheme f, we can compress the image at $ACL(f)$ bits per pixel. Also, we can say that the compression ratio is $s/ACL(f)$.

Compression and Entropy.

Theorem 1 *Let S be a symbol source, and let $minACL(S) = \min(ACL(f))$, where the minimum is taken over all coding schemes, f, of S. Then*

$$H(S) \leq minACL(S) \leq H(S) + 1.$$

Note that no matter what coding scheme we use, we have to use at least one bit for each codeword in the scheme. Hence it is always true that $ACL(f) \geq 1$. How do we get around this?

Definition 4 *Given a symbol source*

$$S = \{s_1,\ s_2,\ \ldots,\ s_q\}$$

with associated probabilities $P(s_i) = p_i$, *define the* nth extension *of* S *to be the set*

$$S^n = \{s_{i_1} s_{i_2} \cdots s_{i_n}\}_{1 \le i_1, i_2, \ldots, i_n \le q}$$

with associated probabilities

$$P(s_{i_1} s_{i_2} \cdots s_{i_n}) = p_{i_1} p_{i_2} \cdots p_{i_n}.$$

Theorem 2 *Let* S *be a symbol source and* S^n *its* n*th extension. Then* $H(S^n) = n\,H(S)$.

Theorem 3 *Let* S *be a symbol source, and let* S^n *be its* n*th extension. Then*

$$H(S) \le \frac{minACL(S^n)}{n} \le H(S) + \frac{1}{n}.$$

Here $minACL(S^n) = \min(ACL(f))$, *where the minimum is taken over all coding schemes of* S^n.

Consequently, we can use the entropy as a very good approximation to the optimal ACL, and hence as a measure of compression rate.