3 cases

1. $6_f^2 > 6_n^2$

2. $6_f^2 = 6_n^2$

If energy/variance of $f$ is much larger than noise, let the signal thrive.

Apr. 4, 2007
3. $6^2 \ll 6^2$

$M = 1$

$g = v + f \implies 6g = 6v + 6f$

$f, v$ are independent
\[ \mathbf{c}_t = 3 \mathbf{b}_t - 6 \frac{\mathbf{e}_t}{e^2} \]

\[ \mathbf{b}_0 = \mathbf{c}_t \]

\[ \mathbf{b}_g \mathbf{g}_{(m+n+1)^2} = \frac{1}{(2m+1)^2} \]

\[ \sum_{m=1}^{\infty} \frac{\mathbf{e}_t}{e^2} \]

\[ (z_2 + m_{2+n-1}) \mathbf{k}_t = -n_{2+n-1} \]
Objectives of Image Coding

- Representation of an image with acceptable quality, using as small a number of bits as possible

Applications:

- Reduction of channel bandwidth for image transmission
- Reduction of required storage

Figure 10.1 Typical environment for image coding.
Issues in Image Coding

1. What to code?
   a. Image density
   b. Image transform coefficients
   c. Image model parameters

2. How to assign reconstruction levels
   a. Uniform spacing between reconstruction levels
   b. Non-uniform spacing between reconstruction levels

3. Bit assignment
   a. Equal-length bit assignment to each reconstruction level
   b. Unequal-length bit assignment to each reconstruction level

Figure 10.2 Three major components in image coding.
Methods of Reconstruction Level Assignments

Assumptions:
- Image intensity is to be coded
- Equal-length bit assignment

Scalar Case

1. Equal spacing of reconstruction levels (Uniform Quantization)

(Ex): Image intensity \( f : 0 \sim 255 \)

\[
\begin{array}{ccc}
\text{f} & \rightarrow & \text{Uniform Quantizer} \\
& & \rightarrow \hat{f}
\end{array}
\]

Number of reconstruction levels: 4 (2 bits for equal bit assignment)
A uniform quantizer. The number of reconstruction levels is 4, $f$ is assumed to be between 0 and 1, and $\hat{f}$ is the result of quantizing $f$. The reconstruction levels and decision boundaries are denoted by $n_i$ and $d_i$, respectively.

Figure 10.3 Example of uniform quantizer.
Scalar Case (cont.)

2. Spacing based on some error criterion

\( r_i \): reconstruction levels (32, 96, 160, 224)

\( d_i \): decision boundaries (0, 64, 128, 192, 256)

Optimally choose \( r_i \) and \( d_i \).

To do this, assume \( f_{\min} \leq f \leq f_{\max} \)

\( J \triangleq \) the number of reconstruction levels

\( p(f) \): probability density function for \( f \)

Minimize

\[
\begin{align*}
\varepsilon &= E[(f - \hat{f})^2] = \int_{f_{\min}}^{f_{\max}} (f - \hat{f})^2 \cdot p(f) \cdot df \\
&= \sum_{j=0}^{J-1} \int_{d_j}^{d_{j+1}} (f - r_j)^2 \cdot p(f) \cdot df
\end{align*}
\]

\[
\begin{align*}
\frac{\partial \varepsilon}{\partial r_j} &= 0 \quad \Rightarrow \quad r_j = 2d_j - r_{j-1} \\
\frac{\partial \varepsilon}{\partial d_j} &= 0 \quad \Rightarrow \quad r_j = \frac{\int_{d_j}^{d_{j+1}} f \cdot p(f) \cdot df}{\int_{d_j}^{d_{j+1}} p(f) \cdot df}
\end{align*}
\]

\( \Rightarrow \) Lloyd-Max Quantizer

These are not simple linear equations.
Solution to the Optimization Problem

<table>
<thead>
<tr>
<th>Bits</th>
<th>Uniform</th>
<th></th>
<th>Gaussian</th>
<th></th>
<th>Laplacian</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r_i</td>
<td>d_i</td>
<td>r_i</td>
<td>d_i</td>
<td>r_i</td>
<td>d_i</td>
</tr>
<tr>
<td>1</td>
<td>-0.5000</td>
<td>-1.0000</td>
<td>-0.7979</td>
<td>-0.7071</td>
<td>-0.7071</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>0.5000</td>
<td>0.0000</td>
<td>0.7979</td>
<td>0.7071</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>2</td>
<td>-0.7500</td>
<td>-1.0000</td>
<td>-1.5104</td>
<td>-1.8340</td>
<td>-1.8340</td>
<td>-1.8340</td>
</tr>
<tr>
<td></td>
<td>-0.2500</td>
<td>-0.5000</td>
<td>-0.4528</td>
<td>-0.4198</td>
<td>-0.4198</td>
<td>-1.1269</td>
</tr>
<tr>
<td></td>
<td>0.2500</td>
<td>0.0000</td>
<td>0.4528</td>
<td>0.4198</td>
<td>0.4198</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>0.7500</td>
<td>0.5000</td>
<td>1.5104</td>
<td>1.8340</td>
<td>1.8340</td>
<td>1.1269</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>-0.8750</td>
<td>-1.0000</td>
<td>-2.1519</td>
<td>-3.0867</td>
<td>-3.0867</td>
<td>-3.0867</td>
</tr>
<tr>
<td></td>
<td>-0.6250</td>
<td>-0.7500</td>
<td>-1.3439</td>
<td>-1.6725</td>
<td>-1.6725</td>
<td>-2.3796</td>
</tr>
<tr>
<td></td>
<td>-0.3750</td>
<td>-0.5000</td>
<td>-0.7560</td>
<td>-0.8330</td>
<td>-0.8330</td>
<td>-1.2527</td>
</tr>
<tr>
<td></td>
<td>-0.1250</td>
<td>-0.2500</td>
<td>-0.2451</td>
<td>-0.2334</td>
<td>-0.2334</td>
<td>-0.5332</td>
</tr>
<tr>
<td></td>
<td>0.1250</td>
<td>0.0000</td>
<td>0.2451</td>
<td>0.2334</td>
<td>0.2334</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>0.3750</td>
<td>0.2500</td>
<td>0.7560</td>
<td>0.8330</td>
<td>0.8330</td>
<td>0.5332</td>
</tr>
<tr>
<td></td>
<td>0.6250</td>
<td>0.5000</td>
<td>1.3439</td>
<td>1.6725</td>
<td>1.6725</td>
<td>1.2527</td>
</tr>
<tr>
<td></td>
<td>0.8750</td>
<td>0.7500</td>
<td>2.1519</td>
<td>3.0867</td>
<td>3.0867</td>
<td>2.3769</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>-0.9375</td>
<td>-1.0000</td>
<td>-2.7326</td>
<td>-4.4311</td>
<td>-4.4311</td>
<td>-4.4311</td>
</tr>
<tr>
<td></td>
<td>-0.8125</td>
<td>-0.8750</td>
<td>-2.0690</td>
<td>-3.0169</td>
<td>-3.0169</td>
<td>-3.7240</td>
</tr>
<tr>
<td></td>
<td>-0.6875</td>
<td>-0.7500</td>
<td>-1.6180</td>
<td>-2.1773</td>
<td>-2.1773</td>
<td>-2.5971</td>
</tr>
<tr>
<td></td>
<td>-0.5625</td>
<td>-0.6250</td>
<td>-1.2562</td>
<td>-1.5778</td>
<td>-1.5778</td>
<td>-1.8776</td>
</tr>
<tr>
<td></td>
<td>-0.4375</td>
<td>-0.5000</td>
<td>-0.9423</td>
<td>-1.1110</td>
<td>-1.1110</td>
<td>-1.3444</td>
</tr>
<tr>
<td></td>
<td>-0.3125</td>
<td>-0.3750</td>
<td>-0.6568</td>
<td>-0.7287</td>
<td>-0.7287</td>
<td>-0.9198</td>
</tr>
<tr>
<td></td>
<td>-0.1875</td>
<td>-0.2500</td>
<td>-0.3880</td>
<td>-0.4048</td>
<td>-0.4048</td>
<td>-0.5667</td>
</tr>
<tr>
<td></td>
<td>-0.0625</td>
<td>-0.1250</td>
<td>-0.1284</td>
<td>-0.1240</td>
<td>-0.1240</td>
<td>-0.2664</td>
</tr>
<tr>
<td></td>
<td>0.0625</td>
<td>0.0000</td>
<td>0.1284</td>
<td>0.1240</td>
<td>0.1240</td>
<td>0.0000</td>
</tr>
<tr>
<td></td>
<td>0.1875</td>
<td>0.1250</td>
<td>0.3880</td>
<td>0.4048</td>
<td>0.4048</td>
<td>0.2644</td>
</tr>
<tr>
<td></td>
<td>0.3125</td>
<td>0.2500</td>
<td>0.6568</td>
<td>0.7287</td>
<td>0.7287</td>
<td>0.5667</td>
</tr>
<tr>
<td></td>
<td>0.4375</td>
<td>0.3750</td>
<td>0.9423</td>
<td>1.1110</td>
<td>1.1110</td>
<td>0.9198</td>
</tr>
<tr>
<td></td>
<td>0.5625</td>
<td>0.5000</td>
<td>1.2562</td>
<td>1.5778</td>
<td>1.5778</td>
<td>1.3444</td>
</tr>
<tr>
<td></td>
<td>0.6875</td>
<td>0.6250</td>
<td>1.6180</td>
<td>2.1773</td>
<td>2.1773</td>
<td>1.8776</td>
</tr>
<tr>
<td></td>
<td>0.8125</td>
<td>0.7500</td>
<td>2.0690</td>
<td>3.0169</td>
<td>3.0169</td>
<td>2.5971</td>
</tr>
<tr>
<td></td>
<td>0.9375</td>
<td>0.8750</td>
<td>2.7326</td>
<td>4.4311</td>
<td>4.4311</td>
<td>3.7240</td>
</tr>
<tr>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>
1.5104

Figure 10.5 Example of a Lloyd-Max quantizer. The number of reconstruction levels is 4, and the probability density function for $f$ is Gaussian with mean of 0 and variance of 1.

Figure 10.6 Comparison of average distortion $D = E[(\hat{f} - f)^2]$ as a function of $L$, the number of reconstruction levels, for a uniform quantizer (dotted line) and the Lloyd-Max quantizer (solid line). The vertical axis is $10 \log_{10} D$. The probability density function is assumed to be Gaussian with variance of 1.
Scalar Case (cont.)

For some densities, the optimal solution can be viewed as follows:

\[ p(g) \text{ is flat density equally likely between } g_{\text{max}}, g_{\text{min}} \]

![Figure 10.7 Nonuniform quantization by companding.](image)

**TABLE 6.1-2. Companding quantization transformation**

<table>
<thead>
<tr>
<th>Probability Density</th>
<th>Forward Transformation</th>
<th>Inverse Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gaussian</strong></td>
<td>( p(f) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{f^2}{2\sigma^2}\right) )</td>
<td>( g = \frac{1}{\sigma} \text{erf} \left(\frac{f}{\sqrt{2} \sigma}\right) )</td>
</tr>
<tr>
<td><strong>Rayleigh</strong></td>
<td>( p(f) = \frac{f}{\sigma^2} \exp\left(-\frac{f^2}{2\sigma^2}\right) )</td>
<td>( g = \frac{1}{\sigma} \exp\left(-\frac{f^2}{2\sigma^2}\right) )</td>
</tr>
<tr>
<td><strong>Laplace</strong></td>
<td>( p(f) = \frac{\alpha}{\sigma} \exp\left(-\frac{</td>
<td>f</td>
</tr>
</tbody>
</table>

where \( \text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_{0}^{z} \exp(-y^2) \, dy \)
Vector Case

$f_1, f_2$: two image intensities

One approach: Separate level assignment for $f_1, f_2$
  - previous discussions apply

Another approach: Joint level assignment for $(f_1, f_2)$
  - typically more efficient than separate level assignment

Minimize reconstruction levels/decision boundaries:

$$\varepsilon = \int \int (f_1 - \hat{f}_1)^2 + (f_2 - \hat{f}_2)^2 \cdot p(f_1, f_2) \cdot df_1 \cdot df_2$$

- efficiency depends on the amount of correlation between $f_1$ and $f_2$
- finding joint density is difficult

(Ex): Extreme case—vector level assignment for $256 \times 256$
  Joint density $P(x_1, x_2, \cdots, x_{256})$ is difficult to get
  Law of diminishing returns comes into play
Codeword Design: Bit Allocation

- After quantization, need to assign binary codeword to each quantization symbol.
- Options:
  - uniform: equal number of bits to each symbol → inefficient
  - non-uniform: short codewords to more probable symbols, and longer codewords for less probable ones.
- For non-uniform, code has to be uniquely decodable:
  - Example: \( L = 4, r_1 \rightarrow 0, r_2 \rightarrow 1, r_3 \rightarrow 10, r_4 \rightarrow 11 \)
  - suppose we receive 100.
  - Can decode it two ways: either \( r_3r_1 \) or \( r_2r_1r_1 \).
  - Not uniquely decodable.
  - One codeword is a prefix of another one.
- Use Prefix codes instead to achieve unique decodability.
Overview

- Goal: Design variable length codewords so that the average bit rate is minimized.
- Define entropy to be:
  \[ H \equiv \frac{1}{L} \sum_{i=1}^{L} p_i \log_2 p_i \]
  
  \( p_i \) is the probability that the \( i \)th symbol is \( a_i \).
- Since \( \sum_{i=1}^{L} p_i = 1 \), we have \( 0 \leq H \leq \log_2 L \)
- Entropy: average amount of information a message contains.
- Example: \( L = 2 \), \( p_1 = 1 \), \( p_2 = 0 \) \( \Rightarrow \) \( H = 0 \) A symbol contains NO information.
- \( p_1 = p_2 = 1/2 \) \( \Rightarrow \) \( H = 1 \) Maximum possible value.
Overview

- Information theory: $H$ is the theoretically minimum possible average bite rate.
- In practice, hard to achieve
- Example: $L$ symbols, $p_i = \frac{1}{L} \rightarrow H = \log_2 L$ uniform length coding can achieve this.
- Huffman coding tells you how to do non-uniform bit allocation to different codewords so that you get unique decodability and get pretty close to entropy.