Transform Coding

• Predictive coding technique is a spatial domain technique since it operates on the pixel values directly.

• Transform coding techniques operate on a reversible linear transform coefficients of the image (ex. DCT, DFT, Walsh etc.)

• Input $N \times N$ image is subdivided into subimages of size $n \times n$.

• $n \times n$ subimages are converted into transform arrays. This tends to decorrelate pixel values and pack as much information as possible in the smallest number of coefficients.

• Quantizer selectively eliminates or coarsely quantizes the coefficients with least information.

• Symbol encoder uses a variable-length code to encode the quantized coefficients.

• Any of the above steps can be adapted to each subimage (adaptive transform coding), based on local image information, or fixed for all subimages.
Walsh Transform (1-D case)

- Given a one-dimensional image (sequence) \( f(m), m = 0,1,\ldots,N-1 \), with \( N = 2^q \), its **Walsh transform** \( W(u) \) is defined as:

\[
W(u) = \frac{1}{N} \sum_{m=0}^{N-1} f(m) \prod_{i=0}^{q-1} (-1)^{b_i(m)b_{q-i}(u)}, \quad u = 0,1,\ldots,N-1.
\]

\( b_k(z) \): \( k^{th} \) bit (from LSB) in the binary representation of \( z \).

- Note that the Walsh-Hadamard transform discussed in the text is very similar to the Walsh transform defined above.

- **Example**: Suppose \( z = 6 = 110 \) in binary representation. Then \( b_0(6) = 0, b_1(6) = 1 \) and \( b_2(6) = 1 \).

- The **inverse Walsh transform** of \( W(u) \) is given by

\[
f(m) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{q-1} (-1)^{b_i(m)b_{q-i}(u)}, \quad m = 0,1,\ldots,N-1.
\]

- Verify that the “inverse works.” Let

\[
g(m) = \sum_{u=0}^{N-1} \left[ \frac{1}{N} \sum_{n=0}^{N-1} f(n) \prod_{i=0}^{q-1} (-1)^{b_i(n)b_{q-i}(u)} \right] \prod_{i=0}^{q-1} (-1)^{b_i(m)b_{q-i}(u)}
\]

This is \( W(u) \) with \( m \) replaced with \( n \).

\[
= \frac{1}{N} \sum_{n=0}^{N-1} f(n) \sum_{u=0}^{N-1} \prod_{i=0}^{q-1} (-1)^{(b_i(n)+b_i(m))b_{q-i}(u)}
\]

\( = f(m) \) (HW problem!!)
Walsh Transform (2-D case)

- Given a two-dimensional $N \times N$ image $f(m, n)$, with $N = 2^g$, its Walsh transform $W(u, v)$ is defined as:

$$W(u, v) = \frac{1}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) \prod_{i=0}^{q-1} (-1)^{b_i(m)b_{q-i}(u) + b_i(n)b_{q-i}(v)},$$

$u, v = 0, 1, \ldots, N - 1$.

- Similarly, the inverse Walsh transform is given by

$$f(m, n) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} W(u, v) \prod_{i=0}^{q-1} (-1)^{b_i(m)b_{q-i}(u) + b_i(n)b_{q-i}(v)},$$

$m, n = 0, 1, \ldots, N - 1$.

- The Walsh transform is
  - Separable (can perform 2-D transform in terms of 1-D transform).
  - Symmetric (the operations on the variables $m, n$ are identical).
  - Forward and inverse transforms are identical.
  - Involves no trigonometric functions (just $+1$ and $-1$), so is computationally simpler.
Discrete Cosine Transform (DCT)

- Given a two-dimensional $N \times N$ image $f(m,n)$, its **discrete cosine transform (DCT)** $C(u,v)$ is defined as:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos \left( \frac{(2m+1)u\pi}{2N} \right) \cos \left( \frac{(2n+1)v\pi}{2N} \right),$$

$u, v = 0, 1, \ldots, N - 1$, where

$$\alpha(u) = \begin{cases} \sqrt{\frac{2}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u = 1, 2, \ldots, N - 1 \end{cases}$$

- Similarly, the **inverse discrete cosine transform (IDCT)** is given by

$$f(m,n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u,v) \cos \left( \frac{(2m+1)u\pi}{2N} \right) \cos \left( \frac{(2n+1)v\pi}{2N} \right),$$

$m, n = 0, 1, \ldots, N - 1$.

- The DCT is
  - Separable (can perform 2-D transform in terms of 1-D transform).
  - Symmetric (the operations on the variables $m, n$ are identical)
  - Forward and inverse transforms are identical

- The DCT is the most popular transform for image compression algorithms like JPEG (still images), MPEG (motion pictures).

- The more recent JPEG2000 standard uses wavelet transforms instead of DCT.

- We will now look at a simple example of image compression using DCT. We will come back to this in detail later.
Discrete Cosine Transform Example

Original Image

Fraction of DCT coeff. Used 0.85, MSE: 0.45

Fraction of DCT coeff. Used 0.65, MSE: 1.6

Fraction of DCT coeff. Used 0.41, MSE: 4
Discrete Cosine Transform Example

Fraction of DCT coeff. Used 0.19, MSE: 7.7

Fraction of DCT coeff. Used 0.08, MSE: 12
Transform Selection

- Commonly used ones are Karhunen-Loeve (Hotelling) transform (KLT), discrete cosine transform (DCT), discrete Fourier transform (DFT), Walsh-Hadamard transform (WHT).

- Choice depends on the computational resources available and the reconstruction error that can be tolerated.

- This step by itself is **lossless** and does not lead to compression. The quantization of the resulting coefficients results in compression.

- The KLT is optimum in terms of packing the most information for any given fixed number of coefficients.

- However, the KLT is data dependent. Computing it requires computing the correlation matrix of the image pixel values.

- The “non-sinusoidal” transforms like the WHT are easy to compute and implement (no multiplications and no trigonometric function evaluation).

- Performance of “sinusoidal” transforms like DCT, DFT, in terms of information packing capability, closely approximates that of the KLT.

- DCT is by far the most popular choice and is used in the **JPEG** (Joint Photographic Experts Group) image standard.
DCT/DFT/WHT comparison for 8×8 subimages, 25% coefficients (with largest magnitude) retained. Note also the blocking artifact.
Subimage Size Selection

- Images are subdivided into subimages of size $n \times n$ to reduce the correlation (redundancy) between adjacent subimages.

- Usually $n = 2^k$, for some integer $k$. This simplifies the computation of the transforms (ex. FFT algorithm).

- Typical block sizes used in practice are $8 \times 8$ and $16 \times 16$.
Bit Allocation

- After transforming each subimage, only a fraction of the coefficients are retained. This can be done in two ways:
  - **Zonal coding**: Transform coefficients with large variance are retained. Same set of coefficients retained in all subimages.
  - **Threshold coding**: Transform coefficients with large magnitude in each subimage are retained. Different set of coefficients retained in different subimages.

- The retained coefficients are quantized and then encoded.

- The overall process of truncating, quantizing, and coding the transformed coefficients of the subimage is called **bit-allocation**.
Comparison of Zonal and Threshold coding for 8×8 DCT subimages, with 12.5% coefficients retained in each case.
Zonal Coding:

- Transform coefficients with large variance carry most of the information about the image. Hence a fraction of the coefficients with the largest variance is retained.

- The variance of each coefficient is calculated based on the ensemble of \((N/n)^2\) transformed subimages, or using a statistical image model. Recall Project #2 on DCT.

- The coefficients with maximum variance are usually located around the origin of an image transform. This is usually represented as a 0-1 mask.

- The same set of coefficients are retained in each subimage.

- The retained coefficients are then quantized and coded. Two possible ways:
  - The retained coefficients are normalized with respect to their standard deviation and they are all allocated the same number of bits. A uniform quantizer then used.
  - A fixed number of bits is distributed among all the coefficients (based on their relative importance). An optimal quantizer such as a Lloyd-Max quantizer is designed for each coefficient.
Threshold Coding:

- In each subimage, the transform coefficients of largest magnitude contribute most significantly and are therefore retained.

- A different set of coefficients is retained in each subimage. So this is an adaptive transform coding technique.

- The thresholding can be represented as \( T(u,v)m(u,v) \), where \( m(u,v) \) is a masking function:

\[
m(u,v) = \begin{cases} 
0 & \text{if } T(u,v) \text{ satisfies some truncation criterion} \\
1 & \text{otherwise}
\end{cases}
\]

- The elements of \( T(u,v)m(u,v) \) are reordered in a predefined manner to form a 1-D sequence of length \( n^2 \). This sequence has several long runs of zeros, which are run-length encoded.

<table>
<thead>
<tr>
<th>1 1 1 0 1 0 0 0</th>
<th>0 1 5 6 14 15 27 28</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1 1 0 0 0 0 0</td>
<td>2 4 7 13 16 26 29 42</td>
</tr>
<tr>
<td>1 1 0 0 0 0 0 0</td>
<td>3 8 12 17 25 30 41 43</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0 0</td>
<td>9 11 18 24 31 40 44 53</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0 0</td>
<td>10 19 23 32 39 45 52 54</td>
</tr>
<tr>
<td>0 0 1 0 0 0 0 0</td>
<td>20 22 33 38 46 51 55 60</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>21 34 37 47 50 56 59 61</td>
</tr>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>35 36 48 49 57 58 62 63</td>
</tr>
</tbody>
</table>

Typical threshold mask    Zigzag ordering of coefficients
• The retained coefficients are encoded using a suitable variable-length code.

• The thresholding itself can be done in three different ways, depending on the “truncation criterion:”
  
  - **A single global threshold** is applied to all subimages. The level of compression differs from image to image depending on the number of coefficients that exceed the threshold.
  
  - **N-largest coding**: The largest $N$ coefficients are retained in each subimage. Therefore, a different threshold is used for each subimage. The resulting code rate (total # of bits required) is fixed and known in advance.
  
  - **Threshold is varied** as a function of the location of each coefficient in the subimage. This results in a variable code rate (compression ratio).

  ➤ Here, the thresholding and quantization steps can be together represented as:

  $\hat{T}(u, v) = \text{round} \left( \frac{T(u, v)}{Z(u, v)} \right)$

  ![Diagram](image-url)
- $Z(u,v)$ is a transform normalization matrix. Typical example is shown below.

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>11</td>
<td>10</td>
<td>16</td>
<td>24</td>
<td>40</td>
<td>51</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>14</td>
<td>19</td>
<td>26</td>
<td>58</td>
<td>60</td>
<td>55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>16</td>
<td>24</td>
<td>40</td>
<td>57</td>
<td>69</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>17</td>
<td>22</td>
<td>29</td>
<td>51</td>
<td>87</td>
<td>80</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>37</td>
<td>56</td>
<td>68</td>
<td>109</td>
<td>103</td>
<td>77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>35</td>
<td>55</td>
<td>64</td>
<td>81</td>
<td>104</td>
<td>113</td>
<td>92</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>64</td>
<td>78</td>
<td>87</td>
<td>103</td>
<td>121</td>
<td>120</td>
<td>101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>92</td>
<td>95</td>
<td>98</td>
<td>112</td>
<td>100</td>
<td>103</td>
<td>99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The values in the $Z$ matrix weigh the transform coefficients according to heuristically determined perceptual or psycho-visual importance. Larger the value of $Z(u,v)$, smaller the importance of that coefficient.

- The $Z$ matrix may be scaled to obtain different levels of compression.

- At the decoding end, $\hat{T}(u,v) = \hat{T}(u,v)Z(u,v)$ is used to denormalize the transform coefficients before inverse transformation.
Example: Coding with different $Z$

Reconstructed Image

Error Image

Quantization matrix $Z$
(9152 non-zero coefficients)

RMSE = 0.023

Quantization matrix $8Z$
(2389 nonzero coefficients)

RMSE = 0.048