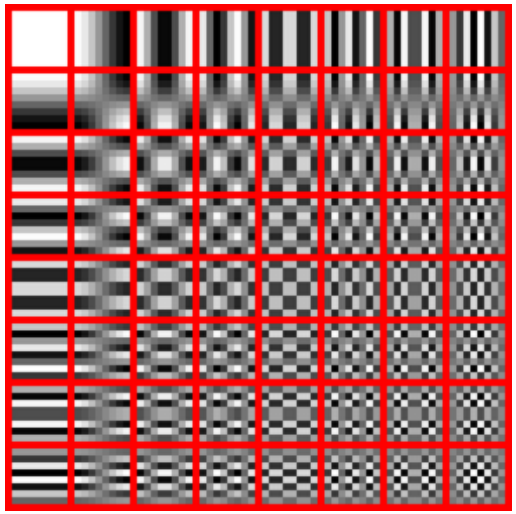
Compression of signals

- Split into blocks (frames)
- Use suitable transform to condense information
  - Signal model
  - Perception model
- Remove some information (for lossy compression)
  - Match channel capacity
  - Apply human perception model (to make less harm)
- Serialize (for 2-D)
- Encode efficiently (entropy coding, e.g. Huffmann)

DCT transform



$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]$$

(equivalent to 1/4 of a DFT of 2N length real, symmetric sequence, upsampled to 4N)

$$X(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n) \cos\left[\frac{\pi}{M}\left(m+\frac{1}{2}\right)k\right] \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)l\right]$$

JPEG compression

- 8x8 block
- Discrete Cosine Transform
  - suitable for natural images (with soft transitions)
  - non-sensitive to edge problems (opp. to DFT)
  - fast implementations (e.g. similar to FFT)
- Lossy opertion: quantization (less important values more roughly)
- Serialization along diagonal (“zig-zag”:low-low frequencies first, then high-low to low-high, high-high last: concentrate similar values)
- RLE + Huffmann coding (variable number of bits per coefficient)
- Transmit Huffmann table or use predefined one

Efficiency: `identify lena.jpg: 512x512 PseudoClass 256c 20kb`

Lena

151 kB

20 kB