CDI15

8. *Discrete Cosine Transform (DCT)*

521 SXD

8.1. 1D DCT

- Used in (lossy) compression of audio (e.g. MP3) and images (e.g. JPEG) (if small high-frequency components can be discarded, it compresses energy strongly).

The coefficients T_{ij} of the 1D DCT are defined as follows:

For $0 \leq j < n$,

$$T_{0j} = \sqrt{1/n}$$

$$T_{ij} = \sqrt{2/n}\cos\frac{(2j+1)i\pi}{2n} \quad 0 < i < n.$$

⇛ Implemented as `dct1` in `cdi_dct`. The inverse transform is implemented as `idct1`.

8.2. 2D DCT

If $f(i,j)$ represents the intensities of an 8×8 block, then the intensities $F(r,s)$ of the transformed block are given by next formula, where we set $m(0) = 1/\sqrt{2}$ and $m(k) = 1$ for $k > 0$:

$$F(r,s) = \frac{m(r)m(s)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\frac{(2i+1)\cdot r\pi}{16} \cdot \cos\frac{(2j+1)\cdot s\pi}{16} \cdot f(i,j)$$

The Inverse Discrete Cosine Transform (IDCT) is given by

$$f(i,j) = \frac{1}{4} \sum_{r=0}^{7} \sum_{s=0}^{7} m(r)m(s) \cos\frac{(2i+1)\cdot r\pi}{16} \cdot \cos\frac{(2j+1)\cdot s\pi}{16} \cdot F(r,s)$$

⇒ Implemented as `dct2` in `cdi_dct`. The inverse transform is implemented as `idct2`.

Remark. Usually the components of the DCT are *thresholded*, i.e., the components that are below a certain threshold are set to 0. This action is carried out by the function `Q` included in `cdi_dct`.

8.3. Examples

In class lab ⇛ L526.

8.4. Using DCT for image compression

The procedure works as follows (Salomon-2008, p. 163):

1. The image is divided into L blocks of 8×8 pixels each. The pixels are denoted $f(i, j)$. If the number of image rows (columns) is not divisible by 8, the bottom row (rightmost column) is duplicated as many times as needed.
2. The DCT in two dimensions (`dct2`) is applied to each block $B^{(l)}$. The result is a block (we'll call it a vector) $w^{(l)}$ of 64 transform coefficients $w_k^{(l)}$ $(k = 0, \ldots, 63)$. The L vectors $w^{(l)}$ become the rows of a matrix:

$$W = \left[w_k^{(l)}\right].$$

3. The 64 columns of W are denoted $C_0, \ldots, C_{63}$. The L elements of C_k are $\left(w_k^{(0)}, \ldots, w_k^{(L-1)}\right)$. In particular, C_0 consists of the L dominant coefficients.

4. The vector C_k is quantized separately to produce a vector Q_k (JPEG does it differently). The elements of Q_k are then written on the output. In practice, variable-length codes are assigned to the elements, and the codes, rather than the elements themselves, are written on the output. Sometimes, as in the case of JPEG, variable-length codes are assigned to runs of zero coefficients, to achieve better compression.

Decoding

The decoder reads the 64 quantized coefficient vectors Q_k of L elements each, saves them as the columns of a matrix, and considers the L rows $w^{(l)}$ of 64 elements each (they are in general different from the original vectors $w^{(l)}$). For each such row, it applies IDCT (`idct2`) to reconstruct (approximately) the 64 values of the block $B^{(l)}$ (again, JPEG does it differently).

8.5. Examples

In class lab ⇛ L528.