Image Processing and Analysis – Math Part: The Discrete Cosine Transform

Ivar Farup

Based upon Chapter 3 of Broughton and Bryan's *Discrete Fourier Analysis and Wavelets* and Maciej Piętka's Lecture Notes from 2010

Overview

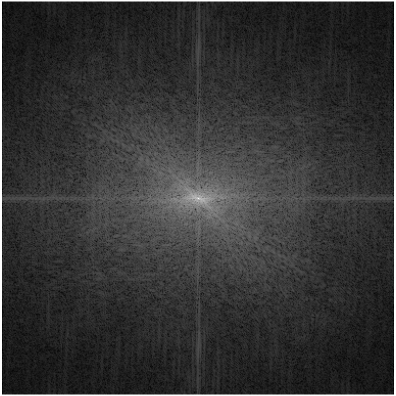
- Data compression issues
- Thresholding as a compression technique
- Thresholding examples
- The Discrete Cosine Transform, DCT
- DCT thresholding compression
- The two-dimensional DCT
- Block transforms

- The DCT is related to the DFT
  - Signal decomposed to sum of basic waveforms
  - Different frequencies and amplitudes
- DCT used in JPEG image compression algorithm
- ‘Lossy’ compression
  - Original image cannot be perfectly reconstructed from the output
  - ‘Lossless’ (reversible) algorithms exist, but are not so efficient

Data Compression Issues

Frequency Representation

- In the frequency domain, most of the signal energy is condensed in a few low-frequency terms



- The rest of the signal can be approximated as zeros
- Easy to compress

1. Transform the image linearly from the spatial domain to a frequency domain by using DFT, DCT or DWT (lossless)

$$\mathbf{x} \to \mathbf{Tx}$$

2. Quantize the coefficients in the transformed representation (lossy)

$$\mathbf{Tx} \to q(\mathbf{Tx})$$

3. Compress the resulting representation (lossless, e.g., Huffman, RLE etc.)

- Decompress the representation (ok)
- The quantisation is *not* reversible/invertible

$$q^{-1}(q(\mathbf{Tx})) \neq \mathbf{Tx}$$

- Information loss at step 2
- In principle, many different images share the same quantized version
- Decompressed image $\neq$ original image $\mathbf{x}$

- General Scheme
  - Compute the DFT of the signal
  - Replace components not exceeding a certain threshold with zeros
- Properties:
  - Works well with signals with energy concentrated in a few frequency components
  - The more terms can be zeroed out, the better compression
  - More sophisticated approach: allow the remaining terms to be stored more economically (using as few bins as possible)

Thresholding as a Compression Technique
Detailed Procedure

- Let $f(t)$ be an analog signal defined on $[0, 1]$
- $\mathbf{x}$ is a sampled signal with $x_k = f(k\Delta T)$, $0 \leq k \leq N-1$
- Choose a threshold parameter $0 \leq c \leq 1$
- Compute $\mathbf{X} = DFT(\mathbf{x})$
- Let $M = \max_{0 \leq k \leq N-1}(|X_k|)$
- Define $\tilde{\mathbf{X}} \in \mathbb{C}^N$ with components

$$\tilde{X}_k = \begin{cases} X_k & \text{if } |X_k| \geq cM, \\ 0 & \text{if } |X_k| < cM. \end{cases}$$

- The vector $\tilde{\mathbf{X}} \in \mathbb{C}^N$ is the compressed version of the signal
- Decompressed signal, $\tilde{\mathbf{x}} = IDFT(\tilde{\mathbf{X}})$, is an approximation to $\mathbf{x}$

- A simple measure of compression efficiency:

$$P(c) = \frac{\text{the number of elements above the threshold}}{N}$$

- $P(c)$ is monotonically decreasing
- $P(c) = 1$ means no compression at all
- $P(c = 1) = 0$ is a perfect compression ($\tilde{\mathbf{X}} = 0$)

- Relative measure of distortion (cf. Lecture 1)

$$mD(c) = 100\frac{||\mathbf{x} - \tilde{\mathbf{x}}||^2}{||\mathbf{x}||^2}$$

- Change in the signal as the fraction of the total energy
- Distortion is due to quantisation of $\mathbf{X}$ or thresholding
- Ideally, $mD(c) = 0$ (perfect reconstruction)

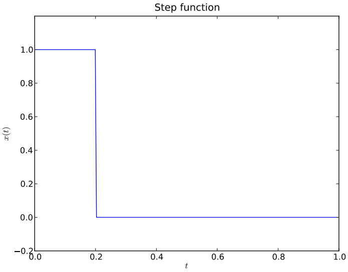
Compression efficiency vs. distortion of the signal

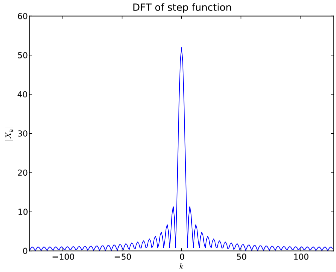
- We would like to have both $P(c)$ and $mD(c)$ as close to zero as possible
- Not possible simultaneously
- When $P(c)$ decreases, $mD(c)$ grows, and vice versa

Thresholding Examples
Discontinuous Step Function

$$x(t) = \begin{cases} 1, & t \leq 0.2 \\ 0, & t > 0.2 \end{cases}, \quad t \in [0, 1)$$

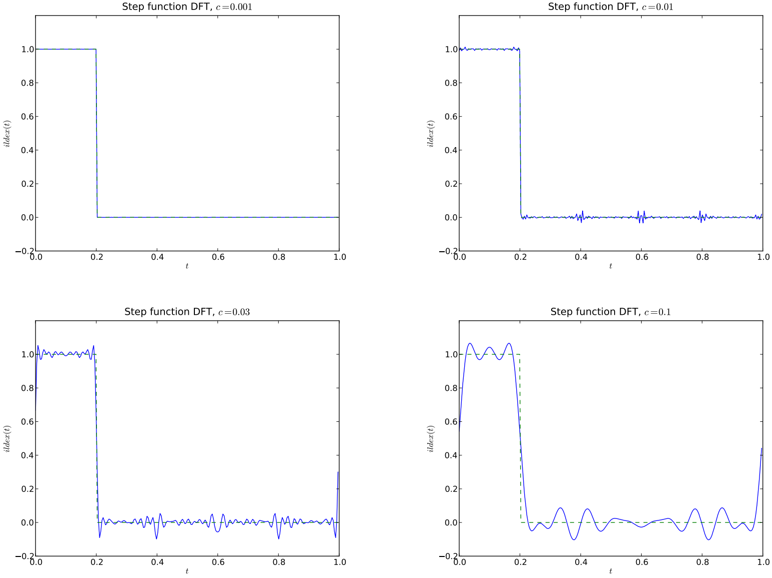
- Sampled at 1/256 s (N = 256)
- $DFT(\mathbf{x})$ condensed around $k = 0$ wiht significant high-requency contribution (to synthesize a discontinuity)





Thresholding Examples

Discontinuous Step Function



Thresholding Examples

Discontinuous Step Function

A difficult case to compress due to the step

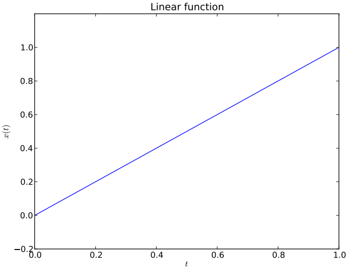
c	0.001	0.01	0.03	0.1	0.5
X_{min}	0.052	0.52	1.56	5.2	26.0
$P(c)$	0.98	0.80	0.26	0.074	0.020
$mD(c)$	$3.6 \cdot 10^{-5}\%$	0.035%	1.0%	4.2%	21%

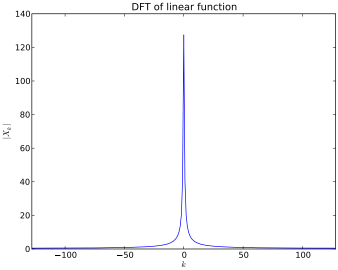
Thresholding Examples

Linear Function

$$x(t) = t, \quad t \in [0, 1)$$

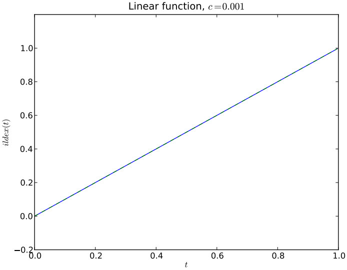
- Sampled at 1/256 s (N = 256)
- $DFT(\mathbf{x})$ distributed quite broadly around $k = 0$

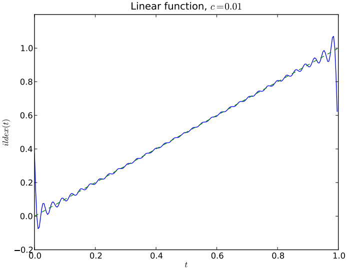


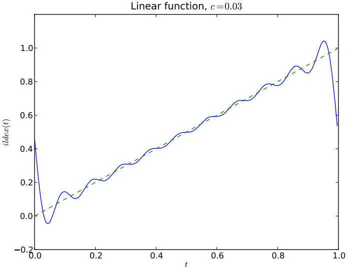


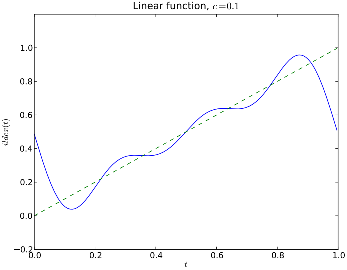
Thresholding Examples

Linear Function









Thresholding Examples

Linear Function

c	0.001	0.01	0.03	0.1	0.5
X_{min}	0.13	1.3	3.8	13	64
$P(c)$	1.0	0.25	0.082	0.027	0.0039
$mD(c)$	$2.3 \cdot 10^{-30}$%	0.44%	1.4%	4.3%	25%

- Even worse than previous example!
- No discontinuity
- Why???

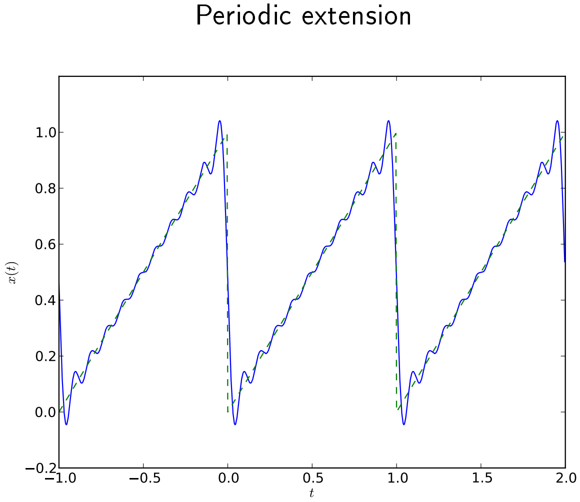
- The IDFT of the signal

$$x_m = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{2\pi i k m / N}$$

  is periodic in m with period N
- The DFT/IDFT pair treats a signal on $[0, 1)$ just as its periodic extension
- Different endpoints ($f(0) \neq f(1)$) appear as discontinuities
- The jump needs to be synthesised by DFT

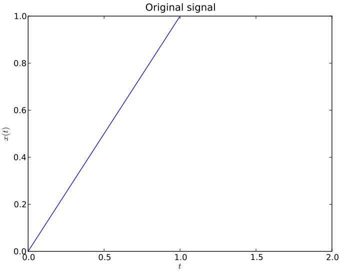
Thresholding Examples

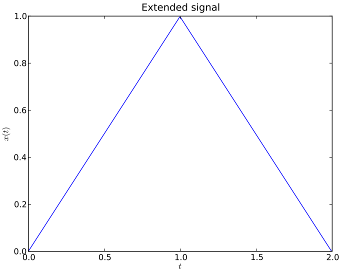
Linear Function



- A simple strategy to deal with discontinuities:
  - Divide the time-domain signal/image into smaller blocks
  - Only a few blocks will be affected by discontinuities
  - Most blocks will compress easily
- The drawback is that the DFT introducces edge discontinuities; possibly at each block boundary
- DCT is designed to overcome the effect of edge discontinuities

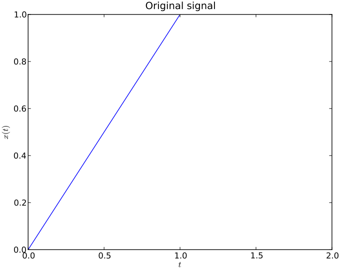
- Extend the time-domain signal to twice its original length by mirroring so that the first sample is equal to the last
- Compute the DFT of the extended signal
- Recover signal from the IDFT by restricting to the appropriate domain

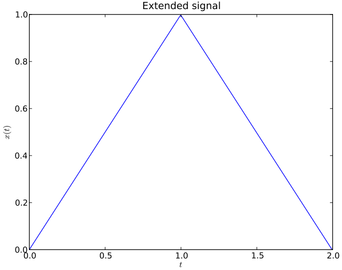




The Discrete Cosine Transform (DCT)

Overview





- Let $\mathbf{x} = (x_0, \ldots, x_{N-1}) \in \mathbb{C}^N$ be the original signal
- An extension ('half-point symmetric extension') $\tilde{\mathbf{x}} \in \mathbb{C}^{2N}$ of $\mathbf{x}$

$$\tilde{x}_m = \begin{cases} x_k, & 0 \leq m \leq N-1, \\ X_{2n-M-1}, & N \leq m \leq 2N-1 \end{cases}$$

- I.e., $\tilde{\mathbf{x}} = (x_0, x_1, \ldots, x_{N-2}, x_{N-1}, x_{N-1}, x_{N-2}, \ldots, x_1, x_0)$

- The $2N$-point DFT of $\tilde{\mathbf{x}}$ has components

$$\tilde{X}_k = \sum_{m=0}^{2N-1} \tilde{x}_m e^{-\pi i k m/N}$$

- Split the sum for $m = 0, \ldots, 2N-1$ into two sums, $m = 0, \ldots, N-1$ and $m = N, \ldots, 2N-1$
- Replace $m' = 2N - m - 1$ in the second sum (and note that $\tilde{x}_{2N-m'-1} = x_{m'}$):

$$\tilde{X}_k = \sum_{m=0}^{N-1} \left(x_m e^{-\pi i k m/N} + x_m e^{\pi i k (m+1)/N} \right)$$

$$= 2e^{\pi i k/2N} \sum_{m=0}^{N-1} x_m \cos\left(\frac{\pi k (m+1/2)}{N} \right)$$

for $k \in \mathbb{Z}$ (periodic in k with period 2N)

- The coefficients $\tilde{X}_k$ for $k = 0, \ldots, 2N-1$ let us recover $\tilde{\mathbf{x}}$ and hence $\mathbf{x}$ by means of the IDFT
- Define
$$c_k = 2\sum_{m=0}^{N-1} x_m \cos\left(\frac{\pi k(m+1/2)}{N}\right)$$
so that $\tilde{X}_k = e^{\pi i k/2N} c_k$
- Symmertries:
$$c_{-k} = c_k, \quad c_{k+2N} = -c_k, \quad c_{2N-k} = -c_k$$
- Thus c_k on the range $0 \le k \le N-1$ gives c_k for any k, and thus $\tilde{X}_k$ for any k, and thus contain all information about the original signal $\mathbf{x}$

The Discrete Cosine Transform (DCT)

Definition

- Let $\mathbf{x} \in \mathbb{C}^N$ be a vector $(x_0, x_1, \ldots, x_{N-1})$
- The discrete cosine transform of $\mathbf{x}$ is the vector $\mathbf{C} \in \mathbb{C}^N$:

$$C_k = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} x_m \cos\left(\frac{\pi k(m-1/2)}{N}\right)$$

  for $1 \leq k \leq N-1$, while

$$C_0 = \sqrt{\frac{1}{N}} \sum_{m=0}^{N-1} x_m$$

- The different scaling factor of C_0 is required to make the corresponding basic waveforms orthogonal

The Discrete Cosine Transform (DCT)

Definition of the Inverse Transform, IDCT

- The IDCT of the vector $\mathbf{C} \in \mathbb{C}^N$ is the vector $\mathbf{x} \in \mathbb{C}^N$ with components

$$x_m = \frac{1}{\sqrt{N}} C_0 + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} C_k \cos\left(\frac{\pi k(m+1/2)}{N}\right)$$

- If $\mathbf{x}$ is real-valued, so is $\mathbf{C}$
- The DCT (like the DFT) is a linear transform and can be represented using matrices

The Discrete Cosine Transform (DCT)

Basic Waveforms for DCT

- The basic waveforms for DCT are vectors $\mathcal{C}_{N,k} \in \mathbb{C}^N$ for $k = 0, 1, \ldots, N-1$
- $\mathcal{C}_{N,0}$ has all the components $1/\sqrt{N}$
- For $k \geq 1$, the m-th component of $\mathcal{C}_{N,k}$ is

$$
\begin{aligned}
\mathcal{C}_{N,0}(m) &= \frac{1}{\sqrt{N}} \\
\mathcal{C}_{N,k}(m) &= \sqrt{\frac{2}{N}} \cos\left(\frac{\pi k(2m+1)}{2N}\right), \quad k \geq 1
\end{aligned}
$$

- $\mathcal{C}_{N,k}$ are orthonormal:

$$(\mathcal{C}_{N,k}, \mathcal{C}_{N,l}) = \begin{cases} 0 & \text{for } k \neq l \\ 1 & \text{for } k = l \end{cases}$$

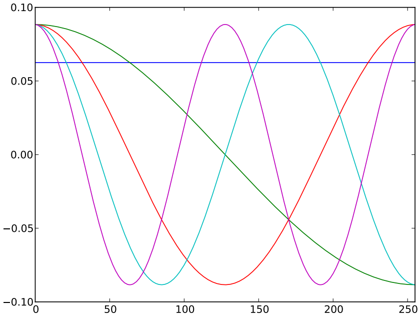
- The DCT coefficients can be computed as

$$C_k = (\mathbf{x}, \mathcal{C}_{N,k})$$

The Discrete Cosine Transform (DCT)

Properties of the Basic Waveforms

- The DCT is a frequency decomposition of the vector $\mathbf{x}$
- *Not* the equivalent to the real part of the DFT
- Basic waveforms $\{\mathcal{C}_{N,k}\}_{k=0,\ldots,N-1}$ are sampled cosine waves
- Frequency of the k-th wave is $f_k = k/2T$



DCT basis function $\mathcal{C}_{256,k}$ for $k = 0, 1, 2, 3, 4$

- Recall that

$$C_k = (\mathbf{x}, \mathcal{C}_{N,k}) = \sum_{m=0}^{N-1} \mathcal{C}_{N,k}(m) x_m$$

- C_k's are components of the vector

$$\mathbf{C} = \mathcal{C}_N \mathbf{x}$$

- $\mathcal{C}_N$ is an $N \times N$ matrix with entries $\mathcal{C}_{N,k}(m)$
- Rows of $\mathcal{C}_N$ are discrete basic waveforms $\mathcal{C}_{N,k}$
- The IDCT can be written as

$$\mathbf{x} = \mathcal{C}_N^T \mathbf{C}$$

The Discrete Cosine Transform (DCT)

Matrix Formulation, Explicit Form

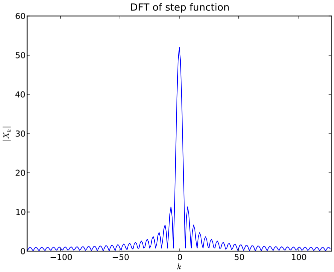
$$\mathcal{C}_N = \begin{pmatrix} 1/\sqrt{N} & 1/\sqrt{N} & \cdots \\ \sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}\frac{1}{2}\right) & \sqrt{\frac{2}{N}}\cos\left(\frac{\pi}{N}\frac{3}{2}\right) & \cdots \\ \vdots & \vdots & \ddots \\ \sqrt{\frac{2}{N}}\cos\left(\frac{\pi(N-1)}{N}\frac{1}{2}\right) & \sqrt{\frac{2}{N}}\cos\left(\frac{\pi(N-1)}{N}\frac{3}{2}\right) & \cdots \end{pmatrix}$$

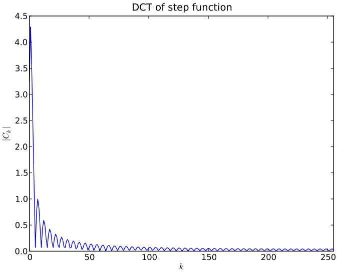
DCT Thresholding Compression

Discontinuous Step Function

$$x(t) = \begin{cases} 1, & t \leq 0.2 \\ 0, & t > 0.2 \end{cases}, \quad t \in [0, 1)$$

- Sampled at $1/256\,\mathrm{s}$ ($\mathrm{N} = 256$)

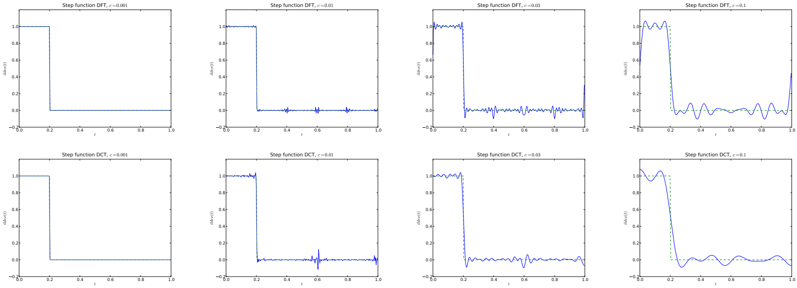




DCT Thresholding Compression

Discontinuous Step Function

DFT



DCT

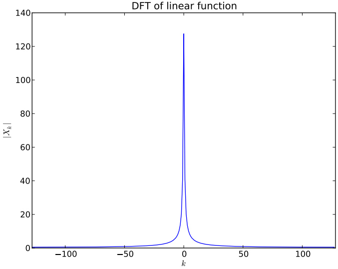
DCT not very much better than the DFT

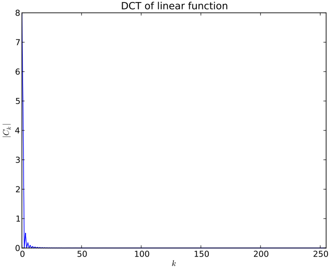
DCT Thresholding Compression

Linear Function

$$x(t) = t, \quad t \in [0, 1)$$

- Sampled at 1/256 s (N = 256)

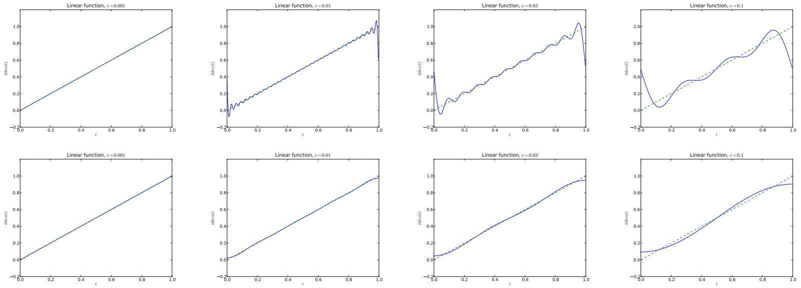




DCT Thresholding Compression

Linear Function

DFT



DCT

DCT very much better than the DFT

The Two-Dimensional DCT

Outline of Derivation

- Like for the one-dimensional DCT, extend the $m \times n$ matrix $\mathbf{A}$ to a $2m \times 2n$ matrix by mirroring about both axes
- Opposite edges of the enlarged image are equal
- No edge discontinuities
- Restrict the IDFT of the compressed image to an appropriate set of $m \times n$ pixels

- Simplified formulation: the 2D DCT can be computed as the matrix product
$$\hat{\mathbf{A}} = \mathcal{C}_m \mathbf{A} \mathcal{C}_n^T$$
- $\mathcal{C}_m$ and $\mathcal{C}_n$ are the one-dimensional DCT matrices
- Similarly to the two-dimensional DFT
- First perform an m-point DCT on the columns of $\mathbf{A}$
- Then an n-point DCT on the rows of the result
- The inverse DCT is given by

$$\mathbf{A} = \mathcal{C}_m^T \hat{\mathbf{A}} \mathcal{C}_n$$

since $\mathcal{C}_m$ is orthonormal

- The entries of the matrix $\hat{\mathbf{A}}$ can be computed explicitly

$$\hat{a}_{k,l} = u_k v_l \sum_{r=0}^{m-1} \sum_{s=0}^{n-1} a_{r,s} \cos\left(\frac{\pi k}{m}\frac{2r+1}{2}\right) \cos\left(\frac{\pi l}{n}\frac{2s+1}{2}\right)$$

with $u_0 = \sqrt{1/m}$, $u_k = \sqrt{2/m}$ for $k > 0$, $v_0 = \sqrt{1/n}$, $v_l = \sqrt{2/n}$ for $l > 0$

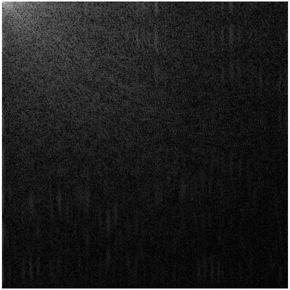
- Maximum amplitude of the DCT component: if $|a_{rs}| \leq a_{\max}$, then

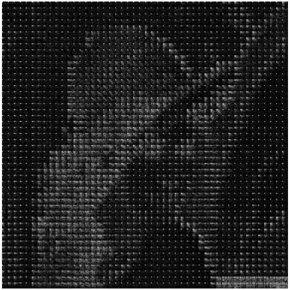
$$|\hat{a}_{kl}| \leq \sqrt{\frac{4}{mn}} \cdot mn \cdot a_{\max} = 2\sqrt{mn} \cdot a_{\max}$$

Block Transforms

Basic Idea

- Compression is better for images divided into smaller blocks
  - Thresholding compression applied to individual blocks
  - Isolated discontinuities affecting only a few blocks
  - Good compression ratio possible for remaining blocks
- JPEG compression standardized to 8×8 blocks
- 'Block DCT': each 8×8 block of the original image replaced by its 2D DCT





Compression of full image

1% 3% 5% 7% 9%

1% 3% 5% 7% 9%

Block compression

- Separate colour (into YCbCr)
- Perform DCT on the image in 8×8 blocks
- Quantize each of the $8 \times 8 = 64$ frequency components in each DCT block (lossy step)
- Compress the resulting numbers using run-length encoding on each block, then Huffman coding on the result

Do the following exercises in the text book: 3.8, 3.9, 3.10