Techniques for Video Compression and Analysis (5LSE0), Module 02 - B

Transformation theory, and Example transforms

Peter H.N. de With
(p.h.n.de.with@tue.nl)

slides version 1.0

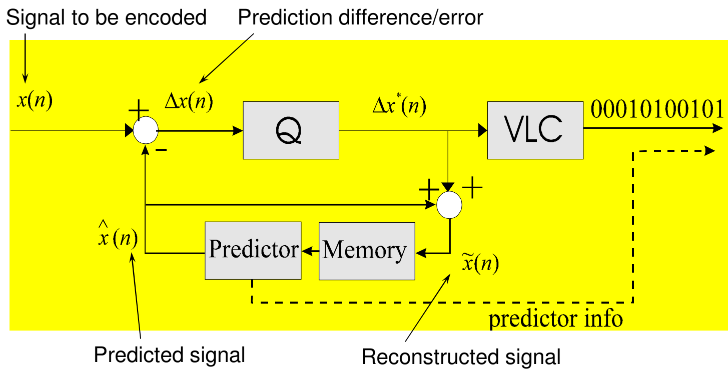
5LSE0 - Mod 02 - B
Part 1

Basic principles of transformation, Fundamental example and theory

Principle of Transform Coding

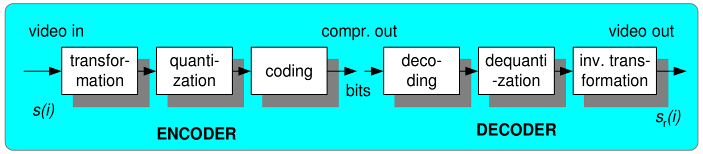
* **Predictive coding: decorrelate video signal as a signal stream**
  - **Sample-by-sample** processing in time domain
  - Mutually decorrelated sample differences
  - **One difference signal** variance (exploited by bit allocation)
* **Transform coding: Alternative approach**
  - Break signal up into **vectors**
  - Decorrelate samples within this vector
  - Quantize transformed samples (transform coefficients) *independently* and with *simple* quantizers (PCM/DPCM)

Diagram of DPCM Encoder & Signals



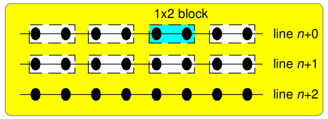
Transform Coding diagram / key steps

- **Transform coding**
  - Transformation (case of decorrelation)
  - Quantization of transform coefficients
  - Coding of coefficients (transformed samples)



Transform / Basic principles – (1)

- **Consider the simplest transform of 1x2 samples**



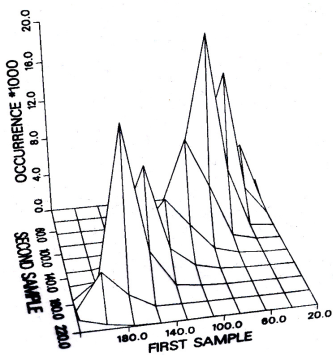
- ***Rewrite*** sample combination $(x0, x1)$ in vector form as: $(x0, x1) = x0.(1, 0) + x1.(0, 1)$
- Sample sequence $(x0, x1)$ can be ***approximated*** by $(x0, x1)$ „= +/-“($(x0+x1)/2$, $(x0+x1)/2$)
- The sample sequence $(x0, x1)$ is ***exactly*** described by taking the approximation and adding a detailed modification

  $(x0, x1) = (x0+x1)/2\ (1, 1) + (-x0+x1)/2\ (-1, 1)$

Transform / Basic principles – (2)

Example 1x2 blocks:
statistics of all $(x0, x1)$ combinations

„Car1.y“ sequence,
f = 13.5 MHz



Note that

- **„mountain“ landscape**
- **all grey values occur**
- **extends over both (2) dimensions**

Transform / Basic principles – (3)

Linear algebra

- **Rewrite the sample combination $(x0, x1)$ in vector form**

$$\begin{pmatrix} x0 \\ x1 \end{pmatrix} = x0.\begin{pmatrix} 1 \\ 0 \end{pmatrix} + x1.\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

- **Or, alternatively**

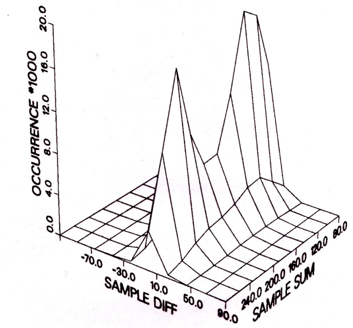
$$\begin{pmatrix} x0 \\ x1 \end{pmatrix} = \frac{x0+x1}{\sqrt{2}.\sqrt{2}}.\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{-x0+x1}{\sqrt{2}.\sqrt{2}}.\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \frac{x0+x1}{\sqrt{2}}.\underline{b}0 + \frac{-x0+x1}{\sqrt{2}}.\underline{b}1$$

- **The sample sequence $(x0, x1)$ is a linear combination with**
  - $(x0+x1)/\sqrt{2}$ as weight or **low-frequency** transform coefficient
  - $(-x0+x1)/\sqrt{2}$ as weight or **high-frequency** transform coefficient
  - the vectors $\underline{b}0$ and $\underline{b}1$ are the basis vectors

Transform / Basic principles – (4)

Example 1x2 blocks:

„Car1.y“ sequence, f = 13.5 MHz, based on ***sample sum*** and ***difference***



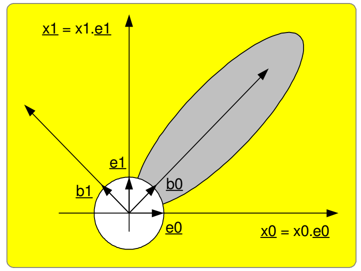
Note that

* same landscape occurs
* extends now over sample sum only!
* Concentration to one dimension

Transform / Basic principles – (5)

Geometric interpretation

* $(e0, e1)$ and $(b0, b1)$ are ***orthogonal*** vectors
* These vectors have normalized length (norm)
* $(b0, b1)$ are along axis (projected) $x0=x1$ and axis $x0=-x1$
* Thus: a ***rotated*** coordinate system!



Transform / Basic principles – (6)

Results of geometric interpretation

* **Transformation is rotation**
  – Rotation to a different coodinate system for the input samples. The amount of samples gives the amount of dimensions.
* **Transformation concentrates energy**
  – Coordinates are chosen such that global approximations are found in the total signal and additions that make the description exact.
* **Transformation to perform ... decorrelation**
  – There is statistical dependence between adjacent samples. This is exploited by the global approximation. New coordinates are decorrelated ones.

Transform / Basic principles – (7)

Matrix Notation

* Rewrite in vector / matrix form (forward)

$$\underline{y}=\begin{pmatrix}y0\\y1\end{pmatrix}=\frac{x0+x1}{\sqrt{2}}\underline{b}0+\frac{-x0+x1}{\sqrt{2}}\underline{b}1=\frac{1}{\sqrt{2}}\begin{bmatrix}+1&+1\\-1&+1\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}x0\\x1\end{pmatrix}=A\underline{x}$$

* Hence, $\underline{y} = \mathbf{A}\,\underline{x}$, and similarly, for the backward transform

$$\underline{x_r}=\begin{pmatrix}x_r0\\x_r1\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}+1&-1\\+1&+1\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}y0\\y1\end{pmatrix}=A^{-1}\cdot\underline{y}$$

* Some properties
  – **Matrices** $\mathbf{A}$ and $\mathbf{A}^{-1}$ are the *transform* and *inverse transform matrix*. Mostly, they are not equal and they are **orthogonal**.
  – **Basis vectors** $\underline{b}0$ and $\underline{b}1$ are the rows in the transform matrix
  – **Reconstructed** samples x_r0 and x_r1 are **linear combination** of basis vectors
  – The **weight** of each vector is called the ***transform coefficient***

Transform / Basic principles – (8)

Matrix notation for generalized 2-D transforms

* In matrix notation

$$y(u,v)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j).A(u,v,i,j)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j).A_V(u,i).A_H(v,j)$$

* Perform 2-D transform *as two times a 1-D transform*, thus

$$y(u,v)=\sum_{i=0}^{N-1}\{\sum_{j=0}^{N-1}x(i,j).A_V(u,i).A_H(v,j)\}=\sum_{i=0}^{N-1}A_V(u,i).\{\sum_{j=0}^{N-1}x(i,j).A_H(v,j)\}$$

* Geometric meaning is: the reconstructed sample data is a *linear combination* of basis images B_{uv}, being an outer product of basis vectors

$$x(i,j)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}y(u,v)\underline{\underline{b}}_{uv}\ldots with.\underline{\underline{b}}_{uv}=\underline{b}_u\otimes\underline{b}_v^T$$

Transform choice / Orthogonal

Decoder equation $\underline{x}=A^T\underline{y}$

* **Structure of orthogonal transform matrix** A^T

$$A^T=\begin{pmatrix} a_{00} & a_{10} & \cdots & a_{N-10} \\ a_{01} & a_{11} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{0N-1} & a_{1N-1} & \cdots & a_{N-1N-1} \end{pmatrix}$$

Basis vector $\underline{a}_0$ Basis vector $\underline{a}_1$ $|a_k|=1$

- Columns are **mutually *orthonormal***
- Columns form the *N basis vectors* of alternate coordinate system

Transform choice / Orthogonal

* **Orthogonal transform matrix** A^T **satisfies** $I=A^T\cdot A$
* **This is a choice, as long as a basis is formed**
* **Examples are: DFT, DCT, Hadamard, Slant, etc.**

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

$$F(u,v)=\frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos(\frac{(2x+1)u\pi}{2N})\cos(\frac{(2y+1)v\pi}{2N})$$

- Columns are **mutually *orthonormal*** *and form N basis vectors*

Design of Transform Coding System

- Transform Coder
  - Decorrelating Transform
    - General requirements
    - Optimal KLT
    - Static transforms
    - Two-dimensional signals
  - Transform Coefficient Quantization
    - Correlation considerations
    - PCM versus DPCM
    - PDF of transform coefficients
    - Bit allocation
    - Perceptual Aspects
  - JPEG standard (still)
  - MPEG standard (video)
  - DV standard (video)

Decomposition 2-D Image Transforms - (1)

17

- Decorrelating transforms on images are applied to (non-overlapping) subblocks of size NxN (Usually N=8)
- Easiest interpretation: subblock is decomposed into *basis images*
- Example:

$$\underbrace{\begin{bmatrix} 6 & 0 \\ 4 & 6 \end{bmatrix}}_{x(m,n)} = \underbrace{4}_{y_{00}} \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}}_{a_{00}} + \underbrace{1}_{y_{01}} \underbrace{\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}}_{a_{01}} + \underbrace{-1}_{y_{10}} \underbrace{\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}}_{a_{10}} + \underbrace{2}_{y_{11}} \underbrace{\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}_{a_{11}}$$

2-D Image Transforms / Reconstruct.– (2)

18

- Subblocks as sum of weighted basis images:

$$x_{m,n} = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} y_{k,l} \underline{\underline{a}}_{k,l}$$

NxN subblock — NxN basis images

- **Basis *images* are built from basis *vectors* of 1-D transforms (assuming separable transform)**

$$\underline{\underline{a}}_{k,l} = \underline{a}_k \otimes \underline{a}_l$$

2-D Transf. /Example 2-D Basis Images – (3)

19

N=2

$$\underline{\underline{a}}_{00} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\underline{\underline{a}}_{01} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$$

$$\underline{\underline{a}}_{10} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$$

$$\underline{\underline{a}}_{11} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

+1 times first vector

-1 times first vector

20

5LSE0 – Mod 02 - B
Part 2

Practical transforms Hadamard, DCT

Walsh-Hadamard Transform – (1)

* Basis functions are built up with +1 and -1 values
* N=2 $A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ N=4 $A = \frac{1}{\sqrt{4}}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$
* N=8

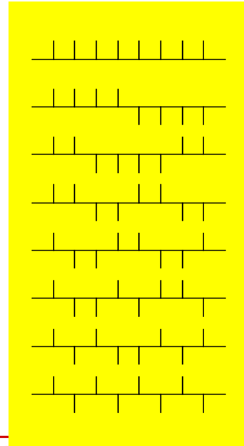
$$A = \frac{1}{\sqrt{8}}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

Build from recurrence rule, but unordered rows!

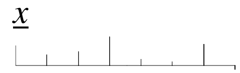
21

Walsh-Hadamard Transform – (2)

* N=8



0.707
0.324
-0.039
-0.129
0
0.326
-0.232
0.162



Hadamard Transform with ordered rows (sequency ordering)

22

(Walsh-)Hadamard transform – (3)

Hadamard transform

* Square waves as basis vectors
* Only +1 and -1 as terms in transform matrix
* Implementation based on additions and subtractions
* Sequency ordering property (increasing # zero transitions)

1-D Hadamard, N=8

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

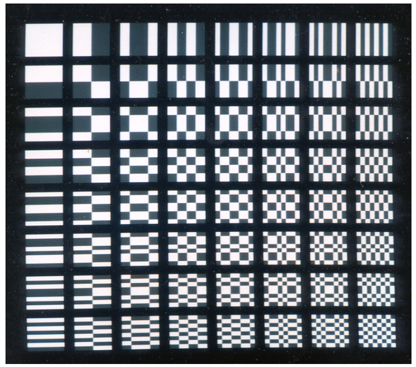
normalize with 1/sqrt(8)

23

Transform / Hadamard transform – (4)

2-D Hadamard transform

* Square wave basis vectors can be used to form 2-D basis images
* 8x8 Transform has 64 basis images of 8x8 samples



24

25

The best fixed transform:

Discrete Cosine Transform (DCT)

26

1-D Discrete Cosine Transform – (1)

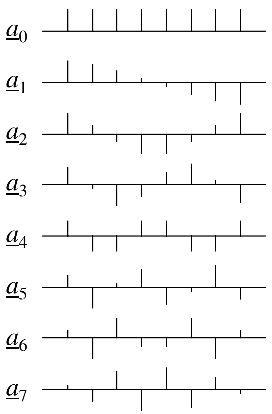
* **WHT uses too "coarse" basis functions**
* **For most "natural" signals, cosine basis functions**
  – seem to correspond with structures reasonably well
  – attractive because of relation to discrete Fourier transform
* **DCT basis functions:**

$$\{a_u\} = \left\{ \sqrt{\tfrac{2}{N}}\, C(u) \cos\left(\frac{(2n+1)u\pi}{2N}\right) \right\}_{n=0,1,\ldots,N-1} \qquad u = 0,1,\ldots,N-1$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & u \neq 0 \end{cases}$$

27

1-D Discrete Cosine Transform – (2)



N=8

$$A^T = \begin{bmatrix} 0.354 & 0.49 & 0.462 & 0.416 & 0.354 & 0.278 & 0.191 & 0.098 \\ 0.354 & 0.416 & 0.191 & -0.098 & -0.354 & -0.49 & -0.462 & -0.278 \\ 0.354 & 0.278 & -0.191 & -0.49 & -0.354 & 0.098 & 0.462 & 0.416 \\ 0.354 & 0.098 & -0.462 & -0.278 & 0.354 & 0.416 & -0.191 & -0.49 \\ 0.354 & -0.098 & -0.462 & 0.278 & 0.354 & -0.416 & -0.191 & 0.49 \\ 0.354 & -0.278 & -0.191 & 0.49 & -0.354 & -0.098 & 0.462 & -0.416 \\ 0.354 & -0.416 & 0.191 & 0.098 & -0.354 & 0.49 & -0.462 & 0.278 \\ 0.354 & -0.49 & 0.462 & -0.416 & 0.354 & -0.278 & 0.191 & -0.098 \end{bmatrix}$$

28

Discrete Cosine Transform – (3)

* **Another way to write down the DCT:**

$$y_u = \sqrt{\frac{2}{N}}\, C(u) \sum_{i=0}^{N-1} x_i \cos\left(\frac{(2i+1)u\pi}{2N}\right) \qquad u = 0,1,\ldots N-1$$

$$x_i = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} C(u)\, y_u \cos\left(\frac{(2i+1)u\pi}{2N}\right) \qquad i = 0,1,\ldots N-1$$

* **Practical systems make use of efficient implementations of the above equations**
  – Butterfly structure: similar structure as Discrete Fourier Transform
  – Finite word length and accuracy considerations
* **DCT in many image/video compression standards!**

Discrete Cosine Transform DCT – (4)

2-D Discrete Cosine Transform (DCT)

- *Orthogonal* transform (in practice, note the constants)

$$y(u,v)=\frac{2}{N}C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j).\cos(\frac{(2i+1)u\pi}{2N}).\cos(\frac{(2j+1)v\pi}{2N})$$

- *Separable* as two times a 1-D transform, thus

$$y(u,v)=\sqrt{\frac{2}{N}}C(u)\sum_{i=0}^{N-1}\cos(\frac{(2i+1)u\pi}{2N})\{\sqrt{\frac{2}{N}}C(v)\sum_{j=0}^{N-1}x(i,j).\cos(\frac{(2j+1)v\pi}{2N})\}$$

- Implementation as: transform rows => transpose => transform rows

Discrete Cosine transform – (5)

Discrete Cosine Transform

- Originally designed for image restoration
- Real terms => real transform, has a relation with DFT with phase encrypted in amplitude information
- implementation more expensive, due to cosine terms (reals)

1-D DCT, N=8

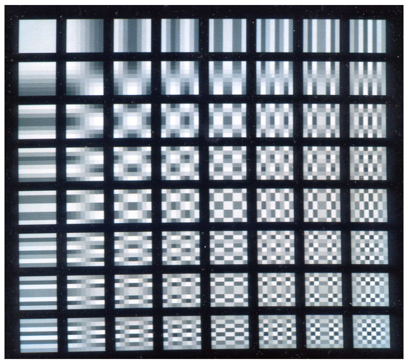
A	A	A	A	A	A	A	A
B	C	D	E	-E	-D	-C	-B
F	G	-G	-F	-F	-G	G	F
C	E	B	-D	D	B	E	-C
A	-A	-A	A	A	-A	-A	A
D	-B	E	C	-C	-E	B	-D
G	-F	F	-G	-G	F	-F	G
E	-D	C	-B	B	-C	D	-E

A=1/sqrt(2),
B=cos(1π/16), C=cos(3π/16),
D=cos(5π/16), E=cos(7π/16),
F=cos(2π/16), G=cos(6π/16),
normalize matrix with 1/2

Discrete Cosine Transform – (6)

2-D DCT transform

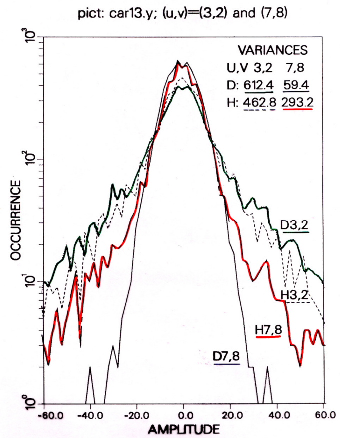
- Cosine wave basis vectors can be used to form 2-D basis images
- 8x8 Transform has 64 basis images of 8x8 samples
- High-frequency coefficients have „modulated“ nature



Transforms / Compare DCT+Hadamard

2-D DCT transform vs. Hadamard transform

- Compare distribution of high-freq. (7,8) and low-freq. (3,2) coefficient
- High-frequency coefficient of DCT has clearly peaked probability distribution
- Energy is transferred from HF to LF coefficients



Alternative normalization of 2-D DCT

- A: *Efficient* 2-D (I)DCT, note the constants, scale down at encoder

$$y(u,v)=\frac{4}{N^2}C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j).\cos(\frac{(2i+1)u\pi}{2N}).\cos(\frac{(2j+1)v\pi}{2N})$$

$$x_r(i,j)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)y(u,v).\cos(\frac{(2i+1)u\pi}{2N}).\cos(\frac{(2j+1)v\pi}{2N})$$

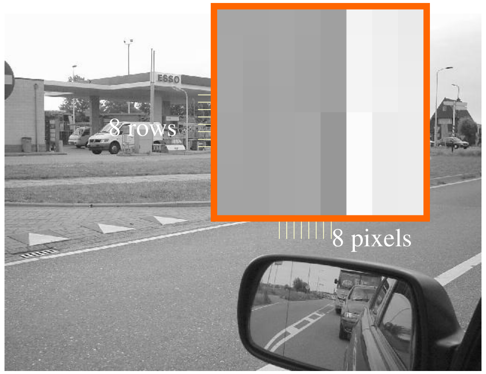
- B: Alternative *High Quality/precision*: downscale at decoder side

$$y(u,v)=C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j).\cos(\frac{(2i+1)u\pi}{2N}).\cos(\frac{(2j+1)v\pi}{2N})$$

$$x_r(i,j)=\frac{4}{N^2}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)y(u,v).\cos(\frac{(2i+1)u\pi}{2N}).\cos(\frac{(2j+1)v\pi}{2N})$$

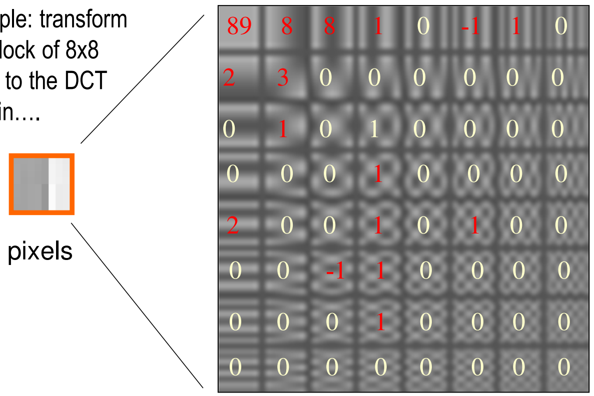
Examples pictures DCT – (1)

Example: extract one 8x8 block of pixels and …



Example pictures DCT – (2)

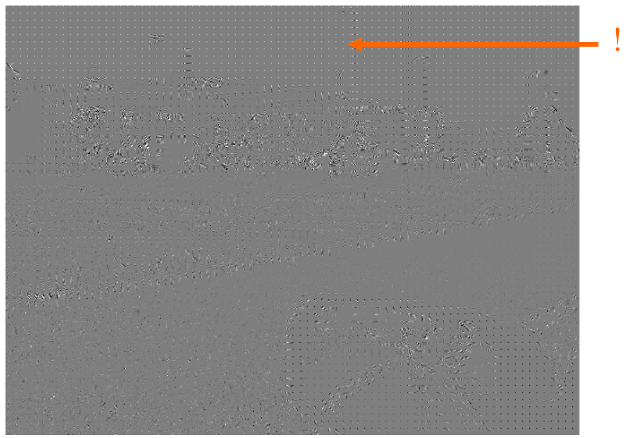
Example: transform that block of 8x8 pixels to the DCT domain....



Cosine patterns/DCT basis functions

Example pictures DCT – (3)

Note the **sparse** distri-bution of infor-mation!



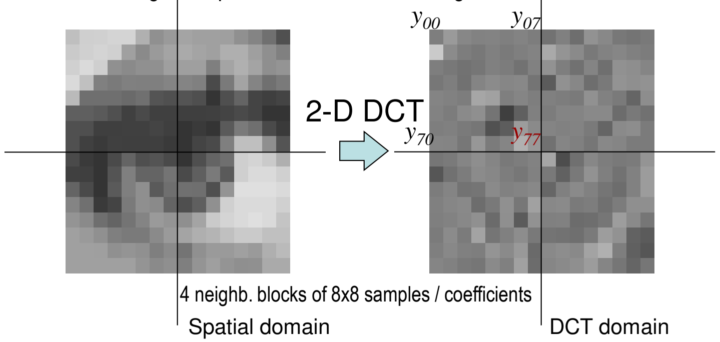
Example: 2-D DCT of Image – (1)

Two distibutions of information: samples and transformed, note sparsity!

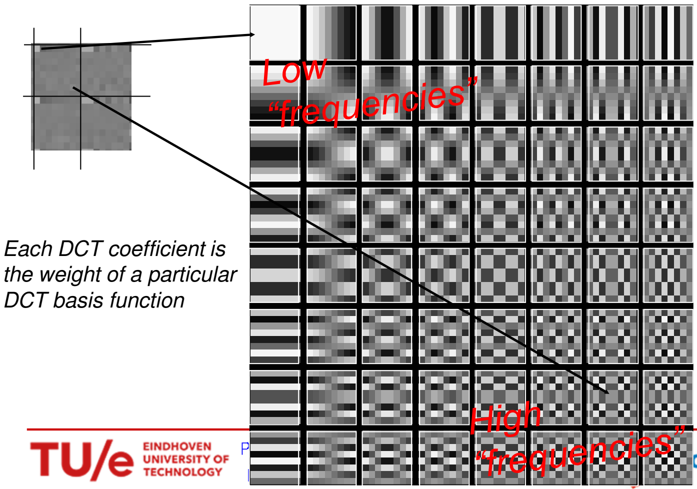
Example: 2-D DCT of Image – (2)

The data in the transformed domain is very different (condensed)!

Left: image samples Right: DCT transformed



Remember DCT Coefficients' Meaning

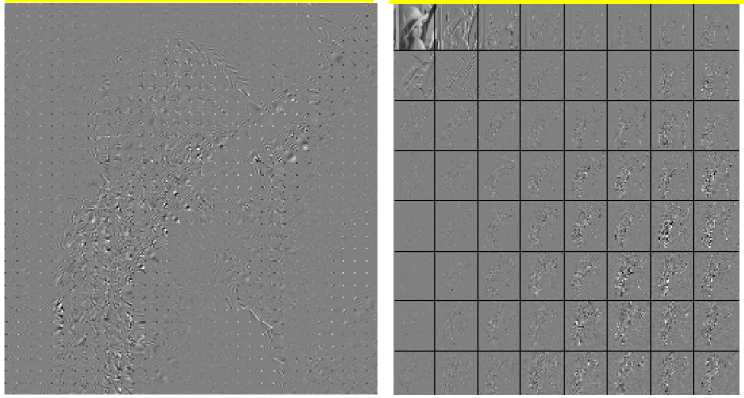


Each DCT coefficient is the weight of a particular DCT basis function

Example: 2-D DCT of Image – (3)

DCT blocks at their spatial locations

Collect all DCT coefficients belonging to same basis image into "band"



General Compression System