Discrete Cosine Transform (DCT)

Dr. Xiqun Lu

College of Computer Science

Zhejiang University

Image Representation

PRIORI BASIS FOR NATURAL IMAGES

DCT

- A priori basis for natural images (not data-driven)
- DCT Basis Functions is defined as:

$$t_{k+1} = \{\sqrt{\frac{2}{N}} a_k \cos\frac{(2n+1)k\pi}{2N}\}_{n=0,1,\cdots,N-1}$$

where $k = 0,1,\cdots,N-1$

$$a_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k \neq 0 \end{cases}$$

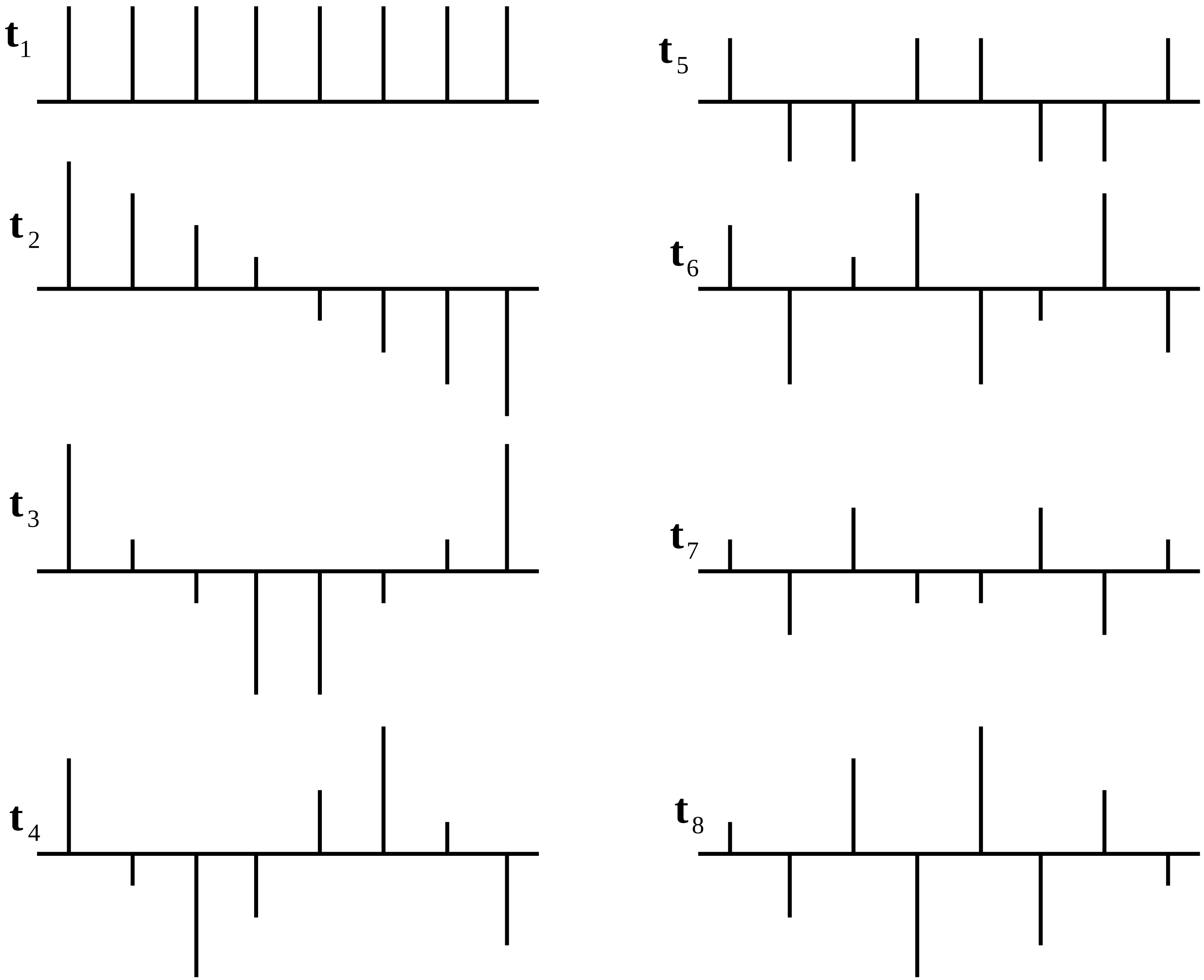
8-Point DCT Example

- Suppose we have a 8-point signal, that is $n = 0, 1, \ldots 7$, then the 8-point DCT transform matrix will be:

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

  – Each column in T^t is computed with a fixed "k", for example the 1st column is computed by $k = 0$, the last column is computed by $k = 7$.
  – You can consider each column of T^t as a basic vector.

8 Basis Vectors of the 8-Point DCT



The DCT in Matrix-Vector Form

$$\mathrm{X}(k) = \sum_{n=0}^{N-1} t_{k+1} x[n]$$

- Example: 8-Point (N = 8) DCT

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{pmatrix} = \begin{pmatrix} 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 \\ 0.49 & 0.416 & 0.278 & 0.098 & -0.098 & -0.278 & -0.416 & -0.49 \\ 0.462 & 0.191 & -0.191 & -0.462 & -0.462 & -0.191 & 0.191 & 0.462 \\ 0.416 & -0.098 & -0.49 & -0.278 & 0.278 & 0.49 & 0.098 & -0.416 \\ 0.354 & -0.354 & -0.354 & 0.354 & 0.354 & -0.354 & -0.354 & 0.354 \\ 0.278 & -0.49 & 0.098 & 0.416 & -0.416 & -0.098 & 0.49 & -0.278 \\ 0.191 & -0.462 & 0.462 & -0.191 & -0.191 & 0.462 & -0.462 & 0.191 \\ 0.098 & -0.278 & 0.416 & -0.49 & 0.49 & -0.416 & 0.278 & -0.098 \end{pmatrix} \begin{pmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \\ x[4] \\ x[5] \\ x[6] \\ x[7] \end{pmatrix}$$

2-D DCT Transform

- Forward DCT

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

- Backward DCT

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

where $C(u), C(v) = {}^{1}\!/\!_{\sqrt{2}} \quad u,v = 0$

$C(u), C(v) = 1 \quad otherwise$

4×4 DCT Transform

$$T = \begin{pmatrix} 0.5000 & 0.6533 & 0.5000 & 0.2706 \\ 0.5000 & 0.2706 & -0.5000 & -0.6533 \\ 0.5000 & -0.2706 & -0.5000 & 0.6533 \\ 0.5000 & -0.6533 & 0.5000 & -0.2706 \end{pmatrix}$$

$$f_1 = \begin{pmatrix} 0.5000 \\ 0.5000 \\ 0.5000 \\ 0.5000 \end{pmatrix} \quad f_2 = \begin{pmatrix} 0.6533 \\ 0.2706 \\ -0.2706 \\ -0.6533 \end{pmatrix} \quad f_3 = \begin{pmatrix} 0.5000 \\ -0.5000 \\ -0.5000 \\ 0.5000 \end{pmatrix} \quad f_4 = \begin{pmatrix} 0.2706 \\ -0.6533 \\ 0.6533 \\ -0.2706 \end{pmatrix}$$

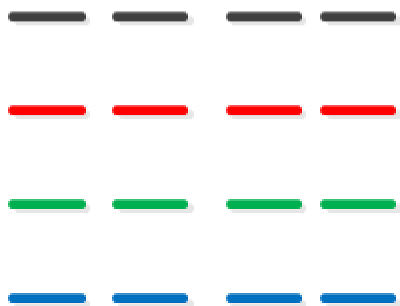
$$f_1 f_1^T = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix} \quad f_1 f_2^T = \begin{pmatrix} 0.3266 & 0.1383 & -0.1383 & -0.3266 \\ 0.3266 & 0.1383 & -0.1383 & -0.3266 \\ 0.3266 & 0.1383 & -0.1383 & -0.3266 \\ 0.3266 & 0.1383 & -0.1383 & -0.3266 \end{pmatrix}$$

$$f_1 f_3^T = \begin{pmatrix} 0.25 & -0.25 & -0.25 & 0.25 \\ 0.25 & -0.25 & -0.25 & 0.25 \\ 0.25 & -0.25 & -0.25 & 0.25 \\ 0.25 & -0.25 & -0.25 & 0.25 \end{pmatrix} \quad f_1 f_4^T = \begin{pmatrix} 0.1353 & -0.3266 & 0.3266 & -0.1353 \\ 0.1353 & -0.3266 & 0.3266 & -0.1353 \\ 0.1353 & -0.3266 & 0.3266 & -0.1353 \\ 0.1353 & -0.3266 & 0.3266 & -0.1353 \end{pmatrix}$$

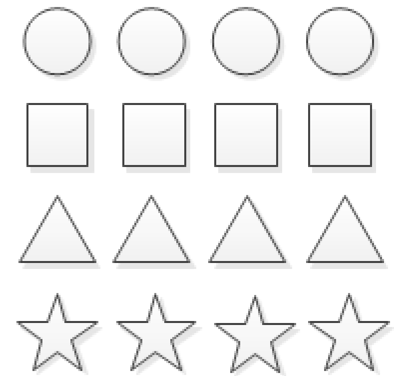
$$\vdots$$

$$f_4 f_4^T = \begin{pmatrix} 0.0732 & -0.1768 & 0.1768 & -0.0732 \\ -0.1768 & 0.4268 & -0.4268 & 0.1768 \\ 0.1768 & -0.4268 & 0.4268 & -0.1768 \\ -0.0732 & 0.1768 & -0.1768 & 0.0732 \end{pmatrix}$$

An Example of 4-by-4 DCT

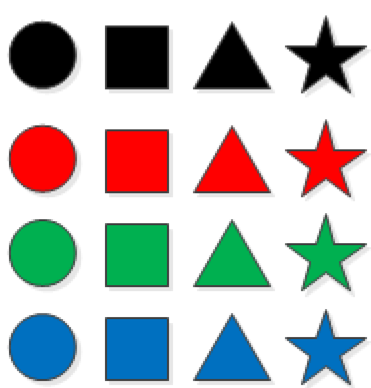


a 4-by-4 block

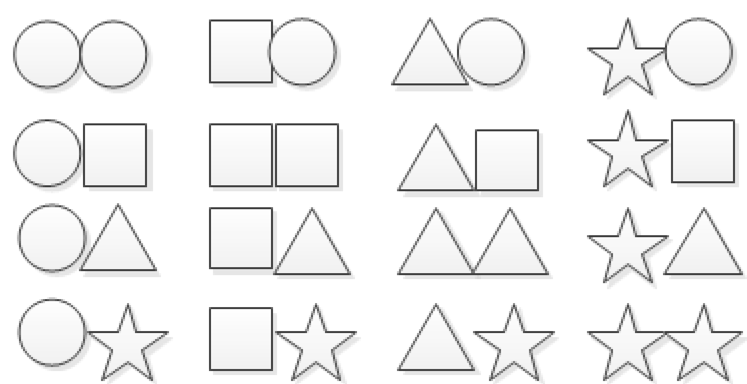


4 Filters

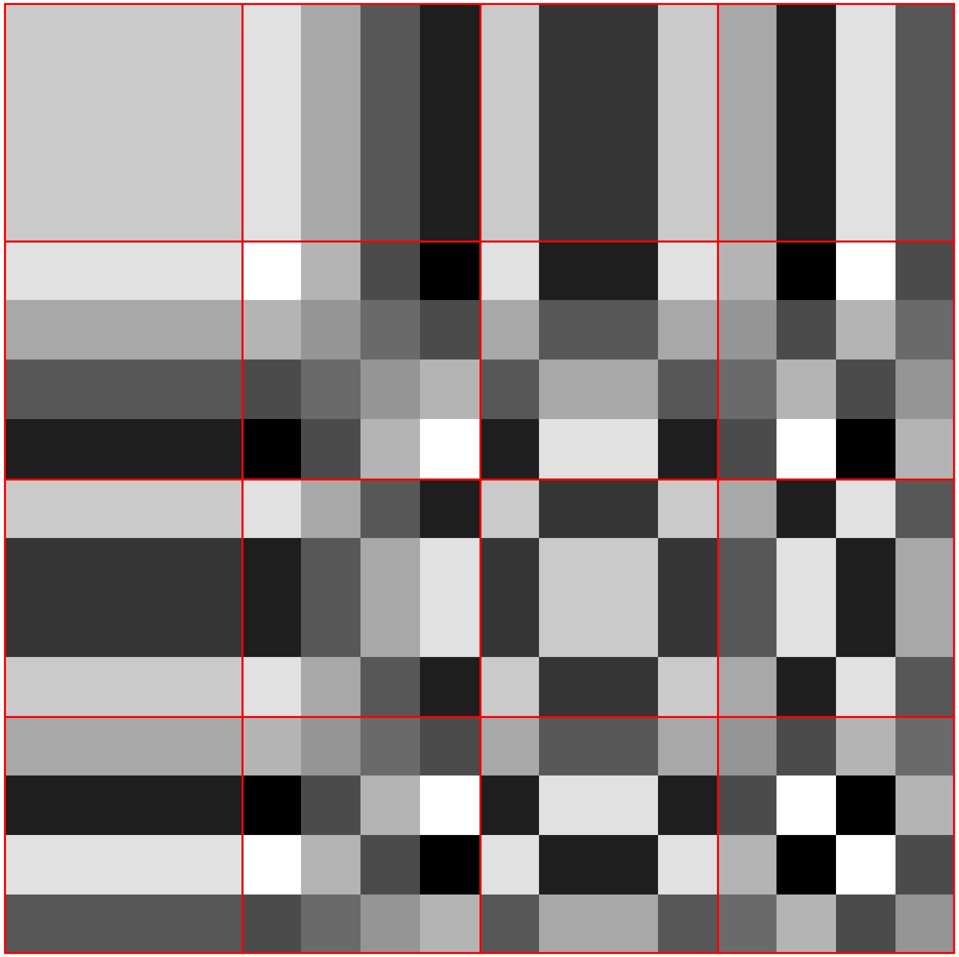
1) Row Filtering



2) Column Filtering

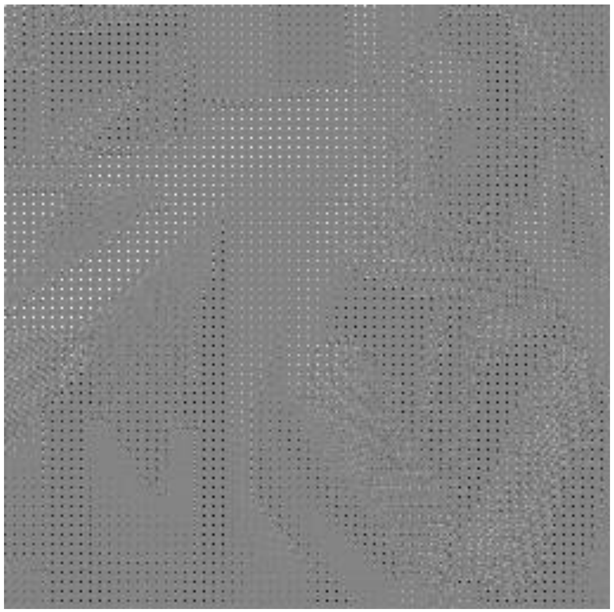


16 Basic Images of 4×4 DCT

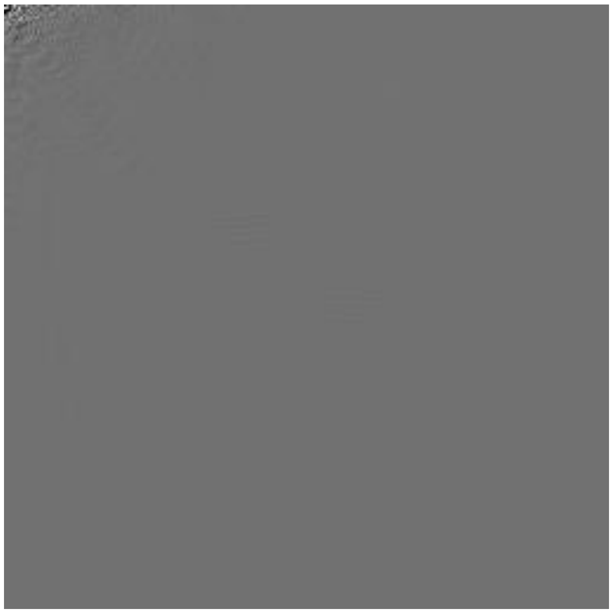


Experimental Results

"Barbara"



DCT Transform of 4×4 blocks



256×256 DCT Transform of Whole Image

Spatial and Frequency Localization Uncertainty Principle

Matlab Code

```
• im = imread('barbara.bmp','bmp');
• [H, W, dim] = size(im);
• if dim ~= 1
•       im = rgb2gray(im);
• end
• N = 4;
• im              = double(im) - 127;   % center the input image around zero
• im_blocks   = im2col(im, [N N], 'distinct');
• num_blocks = size(im_blocks, 2);

• for i = 1:num_blocks
•     DCT_coef(:,i) = dct(im_blocks(:,i), N*N);
• end
•
• im_DCT  = col2im(DCT_coef, [N N], [H W], 'distinct');
• subbands = col2im(DCT_coef', [H/N W/N], [H W], 'distinct');
```

Thank You!

Dr. Xiqun Lu

xqlu@zju.edu.cn