Special Topics in Multimedia System

Indian Institute of Technology Delhi
(IITD)
New Delhi

SIL801

Recap

Predictive Coding

- Loss-less Predictive Coding
- Optimal Predictive Error
- Lossy Predictive Coding

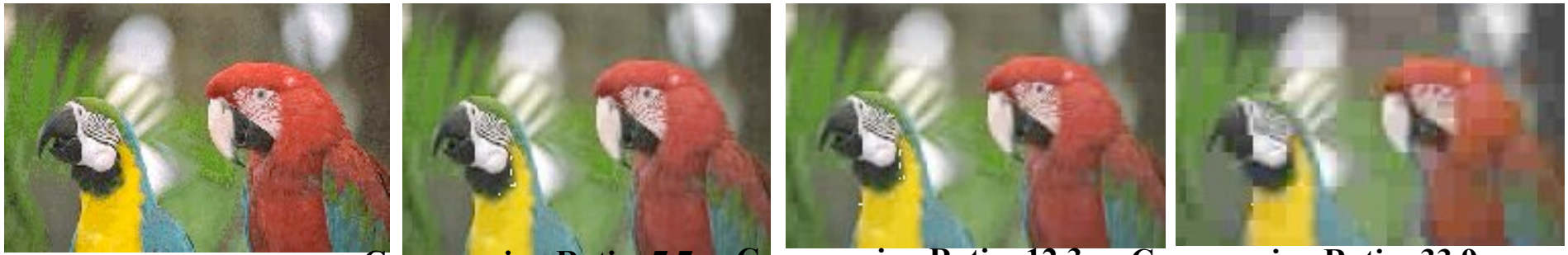
Quantization

Transform Coding

Image Compression

Lossy

- Psychovisual redundancy
- Keep more important information
- Trade off between loss (degradation) and compression



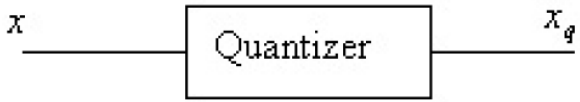
Original **Compression Ratio: 7.7** **Compression Ratio: 12.3** **Compression Ratio: 33.9**

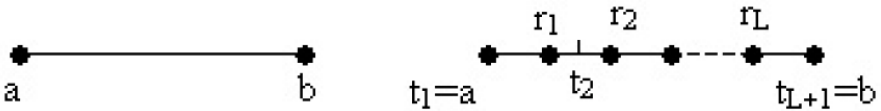
Image Compression

Theory of Quantization

Llyod Max Quantizer

A quantizer maps the continuous variable x into a discrete x_q which takes values from a finite set $\{r_1, \ldots, r_L\}$ of numbers. This mapping is generally a staircase function as shown below.





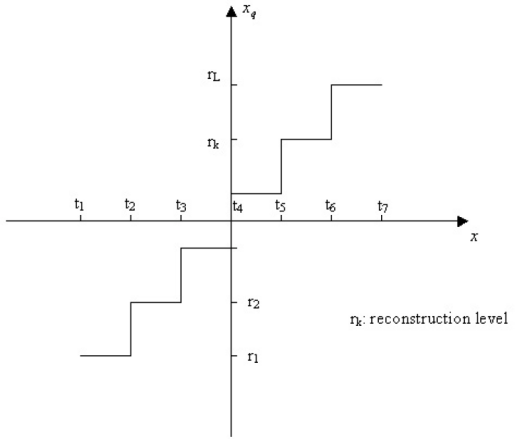


Image Compression

Theory of Quantization

Llyod Max Quantizer

$$\epsilon = E[(x - x_q)^2] = \int_{t_1}^{t_L+1} (x - x_q)^2 p_X(x) dx$$

is minimized. Note that $p_X(x)$ is the PDF of the amplitude of x i.e. $\int_0^A p_X(x)dx = 1$. The MSE can be rewritten as

$$\epsilon = \sum_{i=1}^{L} \int_{t_i}^{t_{i+1}} (x - r_i)^2 p_X(x) dx$$

Image Compression

Theory of Quantization Llyod Max Quantizer

To minimize ϵ w.r.t. t_k and r_k

$$\frac{\partial \epsilon}{\partial t_k} = 0 \quad \Rightarrow \quad (t_k - r_{k-1})^2 p_X(t_k) - (t_k - r_k)^2 p_X(t_k) = 0$$

$$\frac{\partial \epsilon}{\partial r_k} = 0 \quad \Rightarrow \quad 2 \int_{t_k}^{t_{k+1}} (x - r_k) p_X(x) dx = 0, k \in [1, L]$$

Image Compression

Theory of Quantization Llyod Max Quantizer

$$t_k = \frac{r_k + r_{k-1}}{2} \qquad (1)$$

$$r_k = \frac{\int_{t_k}^{t_{k+1}} x p_X(x) dx}{\int_{t_k}^{t_{k+1}} p_X(x) dx} \qquad (2)$$

Image Compression

Theory of Quantization Llyod Max Quantizer

For uniform distribution, the Lloyd-Max quantizer become a linear quantizer. Let

$$p_X(x) = \begin{cases} \frac{1}{t_{L+1} - t_1} & t_1 \leq x \leq t_{L+1} \\ 0 & otherwise \end{cases}$$

Image Compression

Theory of Quantization

Llyod Max Quantizer

Then from (2) we get $r_k = \frac{(t_{k+1}^2 - t_k^2)}{2(t_{k+1} - t_k)} = \frac{t_{k+1} + t_k}{2}$. Now combining this result with (1) yields $t_k = \frac{t_{k+1} + t_{k-1}}{2}$. Thus,

$$t_k - t_{k-1} = t_{k+1} - t_k = constant \stackrel{\Delta}{=} q.$$

Image Compression

Theory of Quantization Llyod Max Quantizer

Also

$$q = \frac{t_{L+1} - t_1}{L}, t_k = t_{k-1} + q, r_k = t_k + q/2$$

and we have $r_k - r_{k-1} = t_k - t_{k-1} = q$.

i.e. all transition and reconstruction levels are equally spaced.

Image Compression

Theory of Quantization Llyod Max Quantizer

Two commonly used PDF's for quantization of image related data are:

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma_x^2}\right) \quad Gaussian$$

$$p_X(x) = \frac{\alpha}{2} \exp(-\alpha|x-\mu|) \quad Laplacian$$

μ is the mean, σ_x^2 is the variance (variance of the Laplacian PDF is $\sigma_x^2 = \frac{2}{\alpha}$).

Image Compression

Theory of Quantization

Entropy constraint

Minimize $D = E\left[(s-\dot{s})^2\right]$ with respect to d_i and r_i $(i = 1,2,\cdots\cdots,L)$, subject to the constraint that the source entropy $H(\dot{s}) = C$ is a constant and the number of output states *L* may vary. These quantizers are called *entropy-constrained quantizers*.

Image Compression

Scalar Quantization

Each sample (pixel) is quantized independently

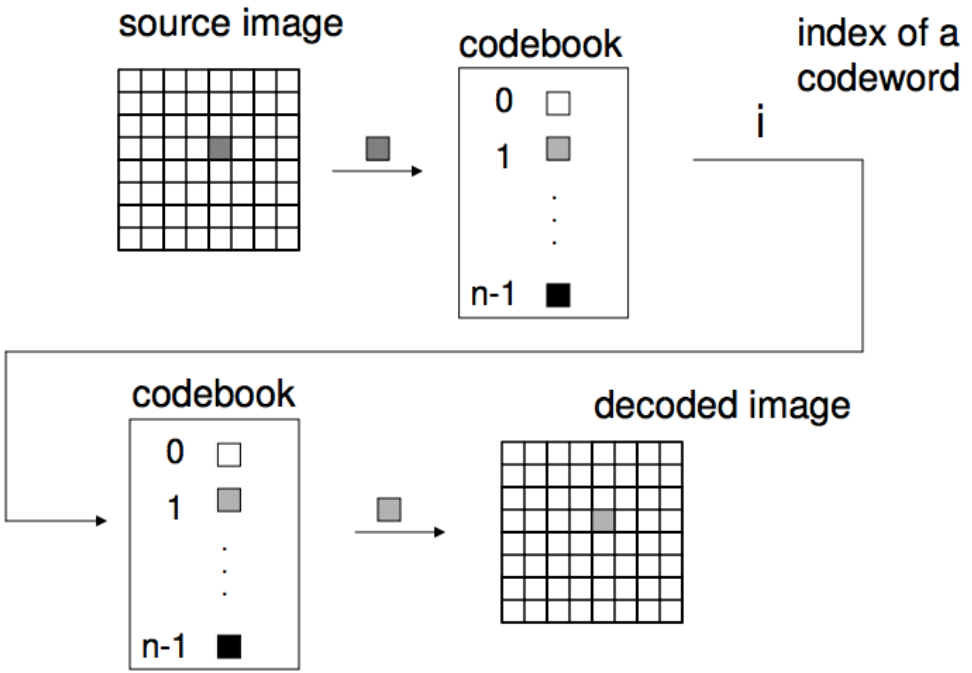


Image Compression

Vector Quantization

block | w | x | / | y | z | = (w,x,y,z) vector

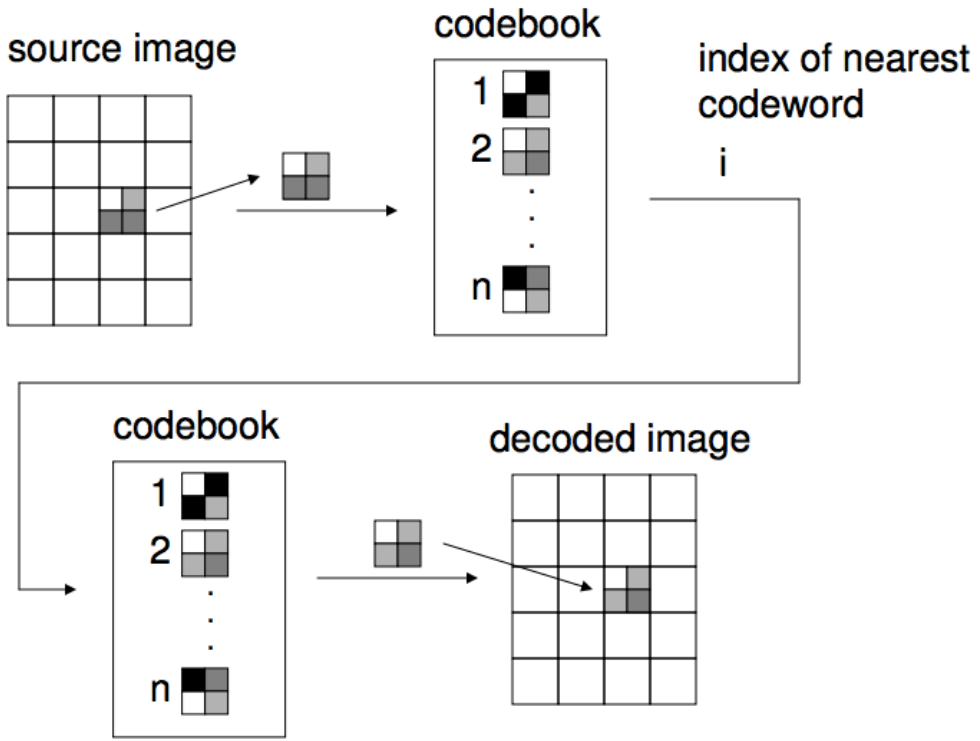


Image Compression

Transform Coding

Transformation

- Represent information in anther space
- Identify and remove correlation
- Quantization of transform coefficients

Information loss!

Example

time/space frequency

(Fourier Transform)

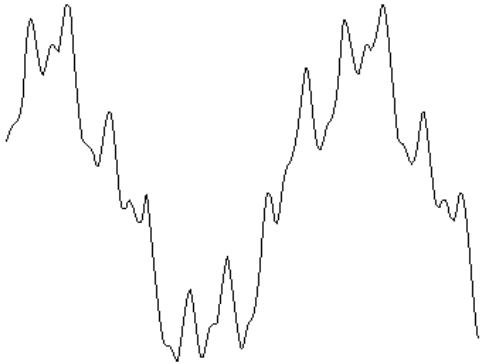
Inverse transformation

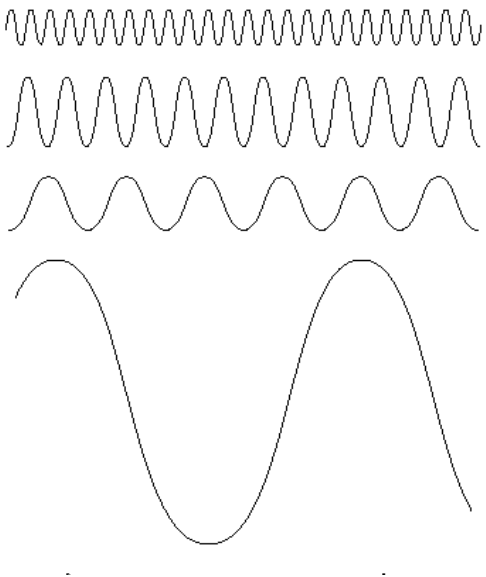
- Bring back information in the original space

Image Compression

Transform Coding

Fourier Transform (Review)





A function as sum of sines and cosines

Image Compression

Transform Coding

Fourier Transform (Review)

Mathematically

Forward

1-D:

$$F(u) \equiv \Im\{f(x)\} = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux}\, dx$$

Inverse

$$f(x) \equiv \Im^{-1}\{F(u)\} = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux}\, du$$

Discrete Fourier Transform

Fast Fourier Transform (FFT)

2-D:

$$F(u,v) = \iint f(x,y) e^{-j2\pi(ux+vy)}\, dx\, dy$$

$$f(x,y) = \iint F(u,v) e^{j2\pi(ux+vy)}\, du\, dv$$

Image Compression

Transform Coding

In general 1D transform

$$\mathbf{f} = \sum_{k=0}^{N-1} t(k)\mathbf{h}_k = \mathbf{Bt},$$

$$where \quad \mathbf{B} = [\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_{N-1}], \mathbf{t} = \begin{bmatrix} t(0) \\ t(1) \\ \vdots \\ t(N-1) \end{bmatrix}.$$

$$\Rightarrow \quad \mathbf{t} = \mathbf{B}^{-1}\mathbf{f} = \mathbf{Af}$$

f and t form a transform pair

Image Compression

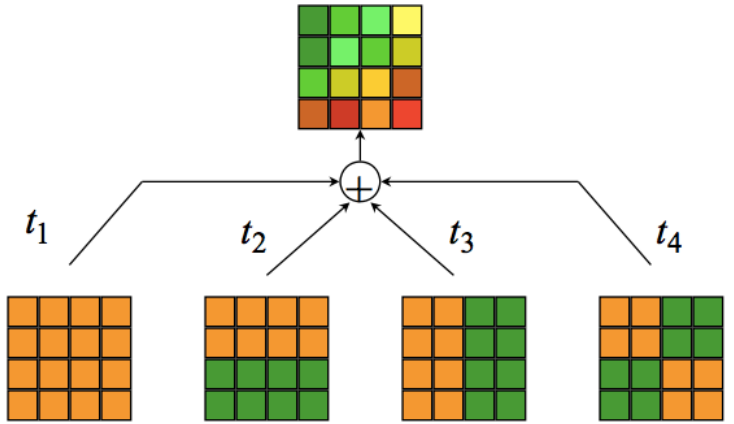
Transform Coding

In general

Represent an image (or an image block) as the linear combination of some basis images and specify the linear coefficients.

$$\mathbf{f} = \sum_{k=0}^{N-1} t(k)\mathbf{h}_k = \mathbf{Bt},$$

$$where \quad \mathbf{B} = [\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_{N-1}], \mathbf{t} = \begin{bmatrix} t(0) \\ t(1) \\ \vdots \\ t(N-1) \end{bmatrix}.$$



Source: http://eeweb.poly.edu/~yao/EL5123/lecture10_TransformCoding_JPEG.pdf

Image Compression

Transform Coding

1D Fourier Transform

$$F(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n) e^{-j2\pi \frac{kn}{N}}, \quad k = 0, 1, ..., N-1;$$

$$f(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} F(k) e^{j2\pi \frac{kn}{N}}, \quad n = 0, 1, ..., N-1.$$

Image Compression

Transform Coding

Discrete Cosine Transform

Popular transform for its performance and efficiency

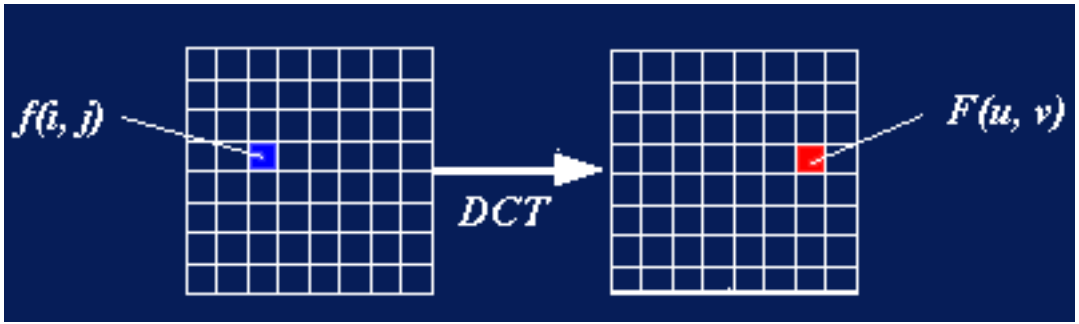


Image Compression

Transform Coding

Discrete Cosine Transform

Forward transform

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

Inverse transform

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

Image Compression

Transform Coding

Discrete Cosine Transform

Energy compaction

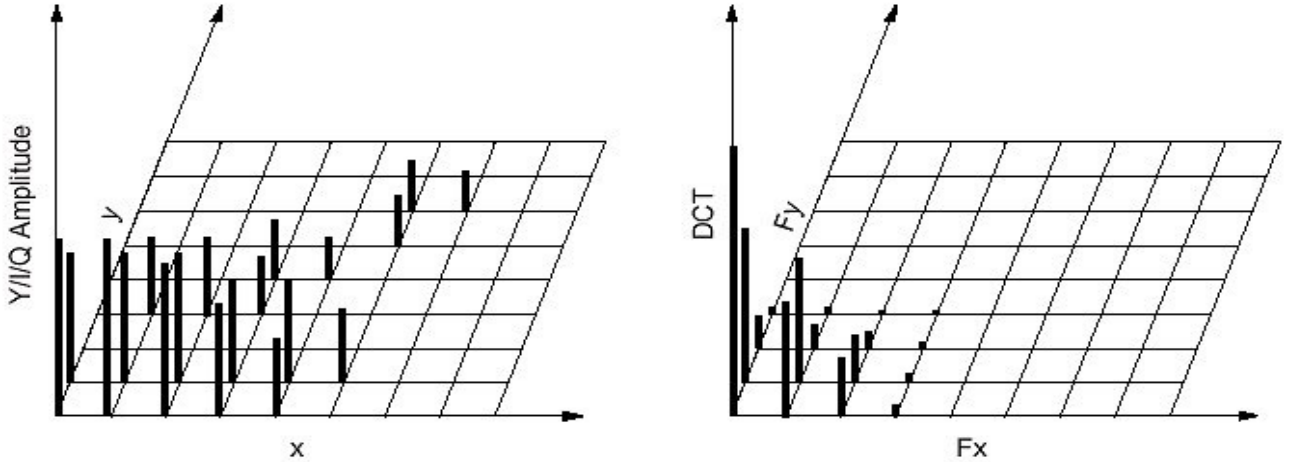
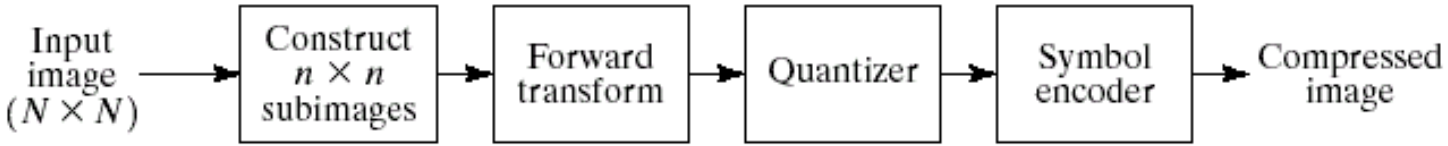


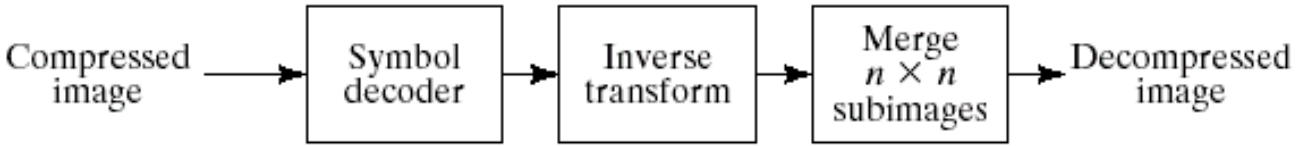
Image Compression

Transform Coding

Pipeline

Compression





Decompression

Image Compression

Transform Coding

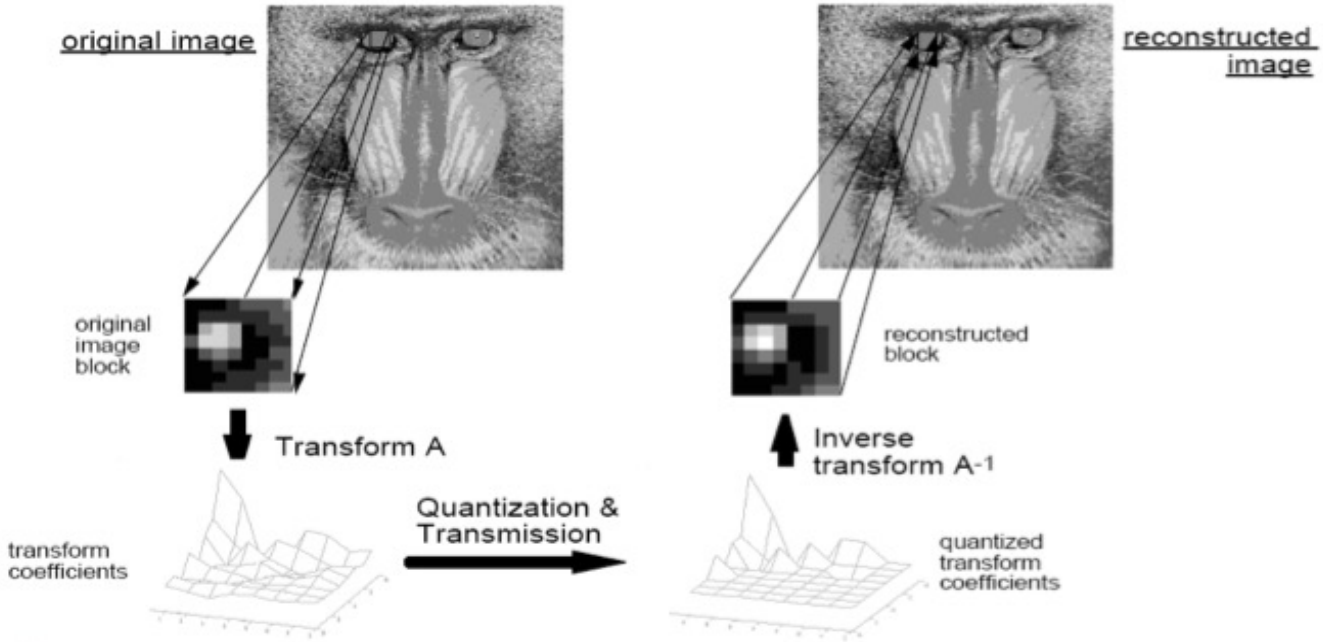


Image Compression

Transform Coding

Why sub image (nxn)?

- Computational benefit
- Typically 8x8, 16x16
- Error is not very high

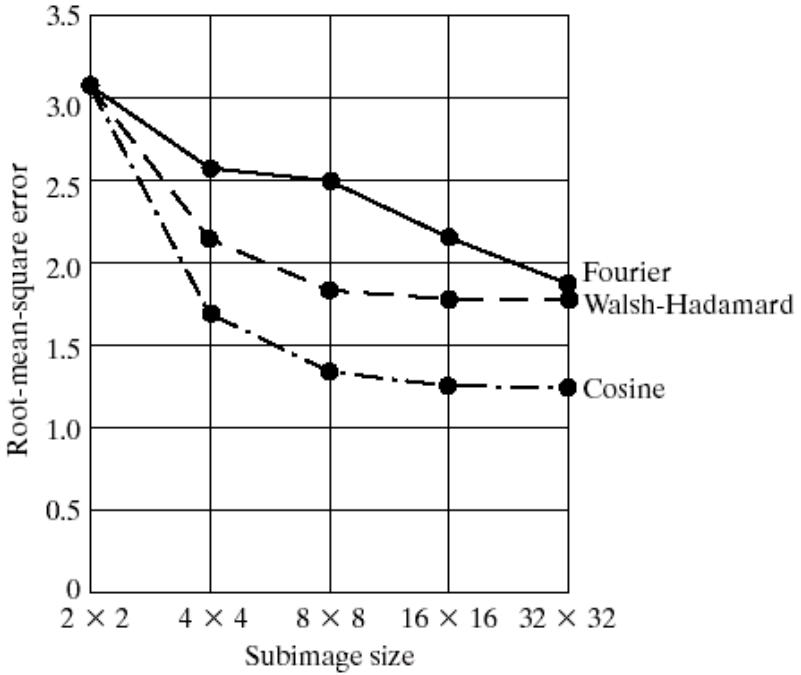


Image Compression

Transform Coding

Why sub image (nxn)?

- Computational benefit
- Typically 8x8, 16x16
- Error is not very high

Which transform?

- Low error for the same number of coefficients
- Computationally fast
- DCT is preferred

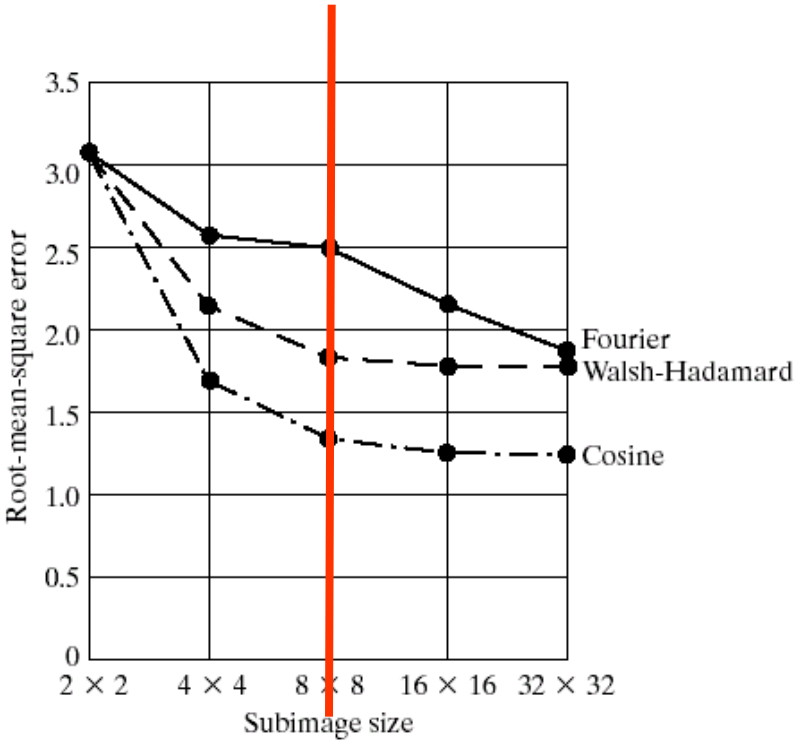


Image Compression

Transform Coding

Quantization Schemes

- Global thresholding
- Local thresholding
- For each block M out of N coefficients to retain

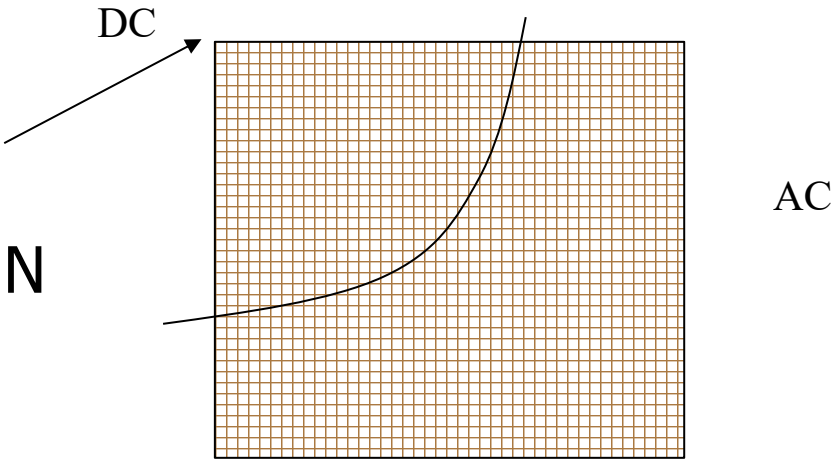


Image Compression

Transform Coding

2D transform

$$\mathbf{F} = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l)\mathbf{H}_{k,l},$$

$$F(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l)H_{k,l}(m,n)$$

Image Compression

Transform Coding

2D transform

$$\mathbf{F} = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l)\mathbf{H}_{k,l},$$

$$F(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l)H_{k,l}(m,n)$$

Discrete Cosine Transform

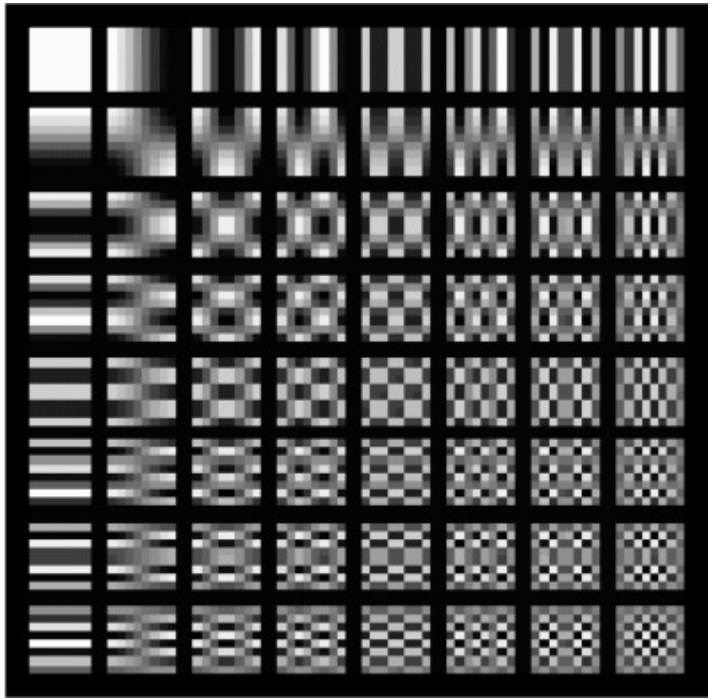
$$H_{k,l}(m,n) = \alpha(k)\alpha(l)\cos\left[\frac{(2m+1)k\pi}{2N}\right]\cos\left[\frac{(2n+1)l\pi}{2N}\right]$$

$$where\ \alpha(k) = \begin{cases} \sqrt{1/N} & k = 0 \\ \sqrt{2/N} & k = 1,\ldots,N-1 \end{cases}$$

Image Compression

Transform Coding

2D transform

Discrete Cosine Transform

8x8 DCT Basis Images