Image Compression and Watermarking

Image Processing

CSE 166

Lecture 12

Announcements

- Assignment 5 will be released today
  - Due May 20, 11:59 PM
- Assignment 6 will be released May 27
  - Due Jun 3, 11:59 PM
- Reading
  - Chapter 8: Image Compression and Watermarking
    - Sections 8.1, 8.9, 8.10, and 8.12

Data compression

- Data compression is reducing the amount of data needed to represent the same amount of information
- Representations that contain irrelevant or repeated information contain redundant data

Data compression

- Data redundancy

$$R = 1 - \frac{1}{C}$$

where compression (ratio)

$$C = \frac{b}{b'}$$

where

b and b' are the number of bits in two different representations of the same information

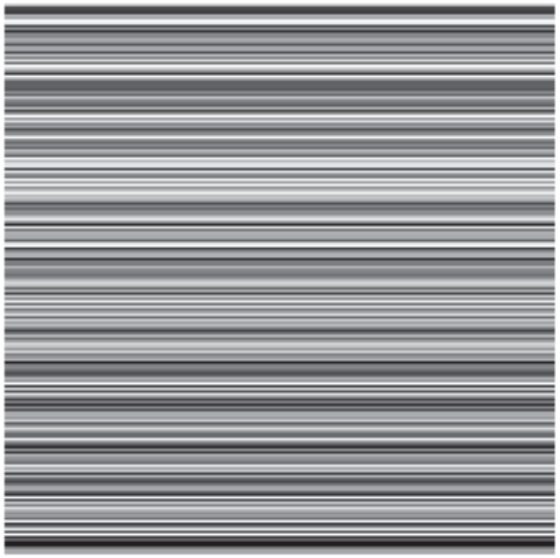
Data redundancy in images

Coding redundancy



Does not need all 8 bits

Spatial redundancy



Information is unnecessarily replicated

Irrelevant information



Information is not useful

Image information

- Entropy

$$\tilde{H} = -\sum_{k=0}^{L-1} p_r(r_k) \log_2(p_r(r_k))$$

Average information (in bits) per pixel

where

L is the number of intensity or gray levels
r_k is input image intensity or gray level value k
$p_r(r_k)$ is normalized histogram of input image

  - It is not possible to encode input image with fewer than $\tilde{H}$ bits/pixel

Fidelity criteria

- Objective (quantitative)

Error

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

where

$\hat{f}(x, y)$ is the approximation of the input image
$f(x, y)$ is the input image

Fidelity criteria

- Total error

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|\hat{f}(x,y)-f(x,y)|$$

- Root-mean-square error

$$e_{\text{rms}}=\left(\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\hat{f}(x,y)-f(x,y)\right)^2\right)^{1/2}$$

- Mean-square signal to noise ratio (SNR)

$$\text{SNR}_{\text{ms}}=\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left(\hat{f}(x,y)-f(x,y)\right)^2}$$

Fidelity criteria

- Subjective (qualitative)

TABLE 8.2
Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

Approximations

Objective (quantitative) quality

rms error (in intensity levels; lower is better)

5.17 15.67 14.17



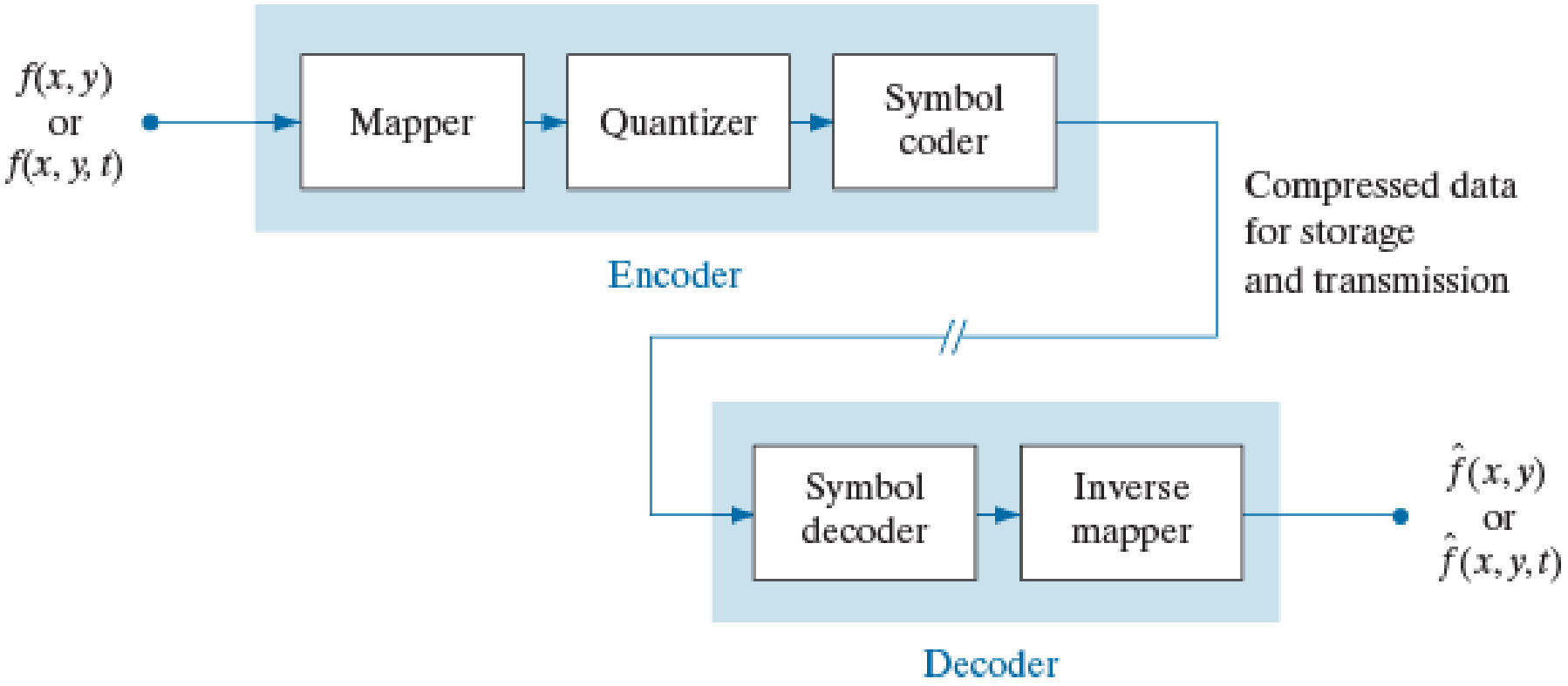
(a) (b) (c)

Subjective (qualitative) quality, relative

(a) is better than (c), (c) is better than (b)

(a) is better than (b), (b) is better than (c)

Compression system

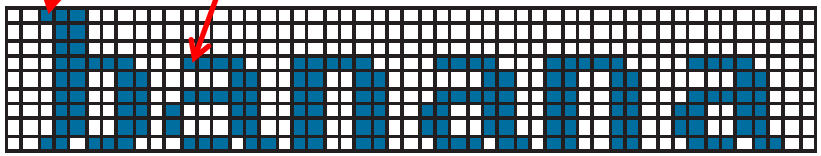


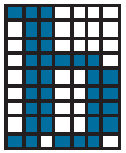
Compression methods

- Huffman coding
- Golomb coding
- Arithmetic coding
- Lempel-Ziv-Welch (LZW) coding
- Run-length coding
- Symbol-based coding
- Bit-plane coding
- Block transform coding
- Predictive coding
- Wavelet coding

Symbol-based coding

(0,2) (3,10) …

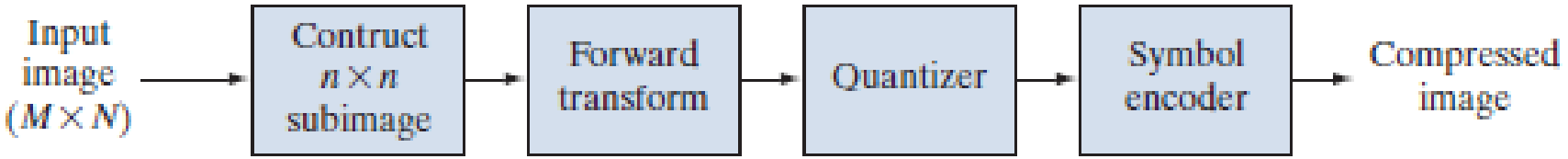


Token	Symbol
0	
1	
2	

Triplet
(0, 2, 0) (3,10, 1) (3, 18, 2) (3, 26, 1) (3, 34, 2) (3, 42, 1)

Block-transform coding

- Encoder



- Decoder



Block-transform coding

- Example: discrete cosine transform

$$n \times n \text{ subimage } g(x,y)$$

$$T(u,v) = \sum_{x=0}^{n-1}\sum_{y=0}^{n-1} g(x,y)\alpha(u)\alpha(v)\cos\left(\frac{(2x+1)u\pi}{2n}\right)\cos\left(\frac{(2y+1)v\pi}{2n}\right)$$

where

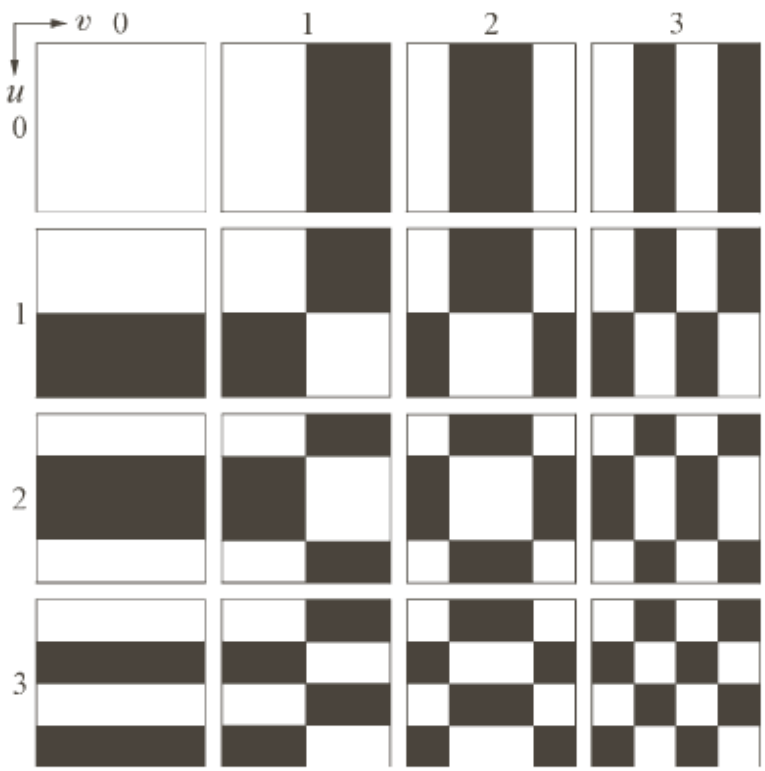
$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{if } u = 0 \\ \sqrt{\frac{2}{n}} & \text{otherwise} \end{cases} \quad \text{and} \quad \alpha(v) = \begin{cases} \sqrt{\frac{1}{n}} & \text{if } v = 0 \\ \sqrt{\frac{2}{n}} & \text{otherwise} \end{cases}$$

- Inverse

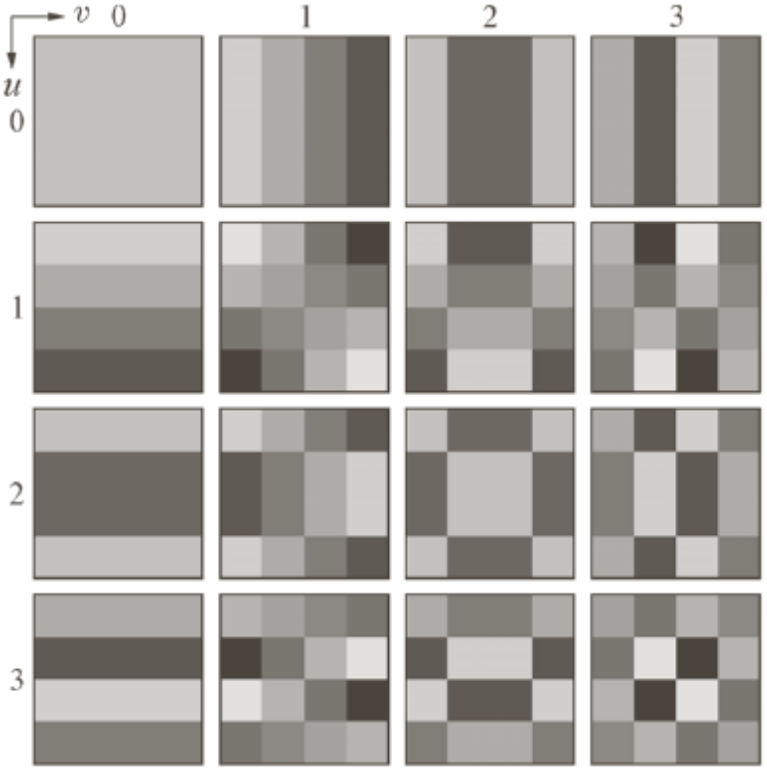
$$g(x,y) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} T(u,v)\alpha(u)\alpha(v)\cos\left(\frac{(2x+1)u\pi}{2n}\right)\cos\left(\frac{(2y+1)v\pi}{2n}\right)$$

Block-transform coding

4x4 subimages (4x4 basis images)



Walsh-Hadamard transform



Discrete cosine transform

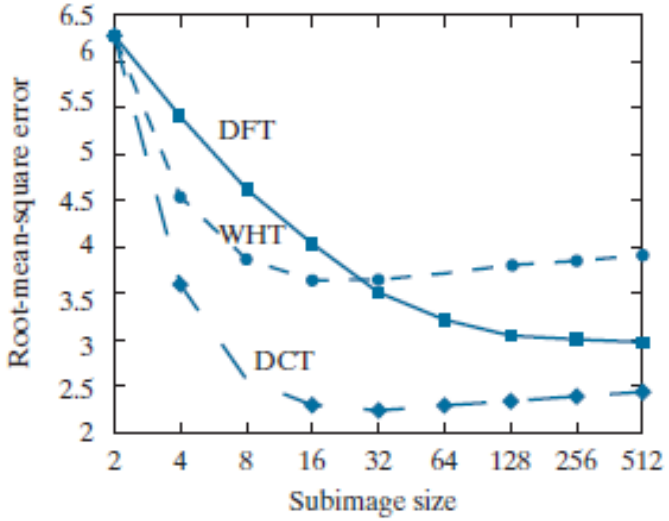
Block-transform coding

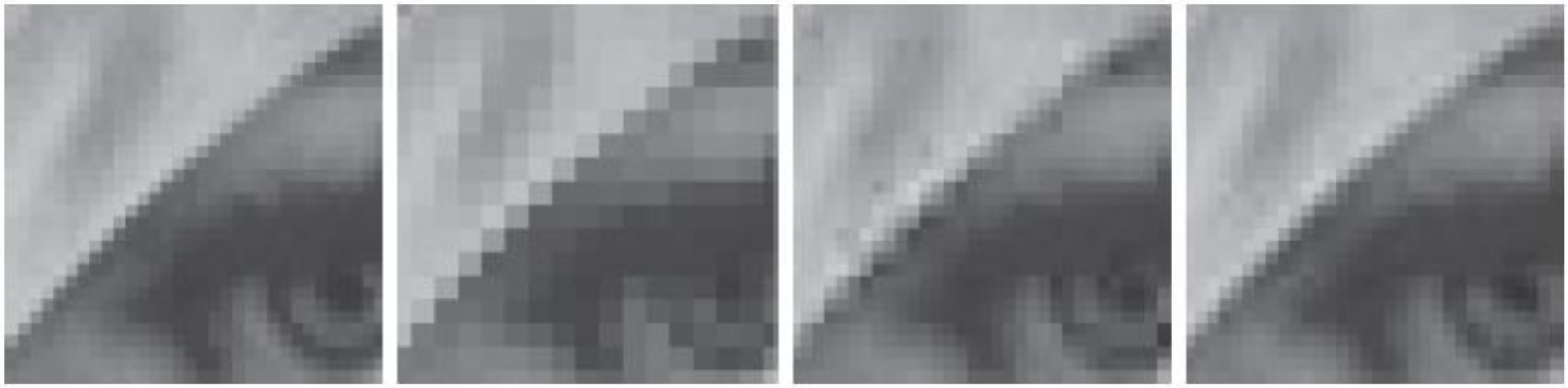
8x8 subimages	Fourier transform	Walsh-Hadamard transform	cosine transform
Retain 32 largest coefficients			
Error image			
rms error	2.32	1.78	1.13

Lower is better

Block-transform coding

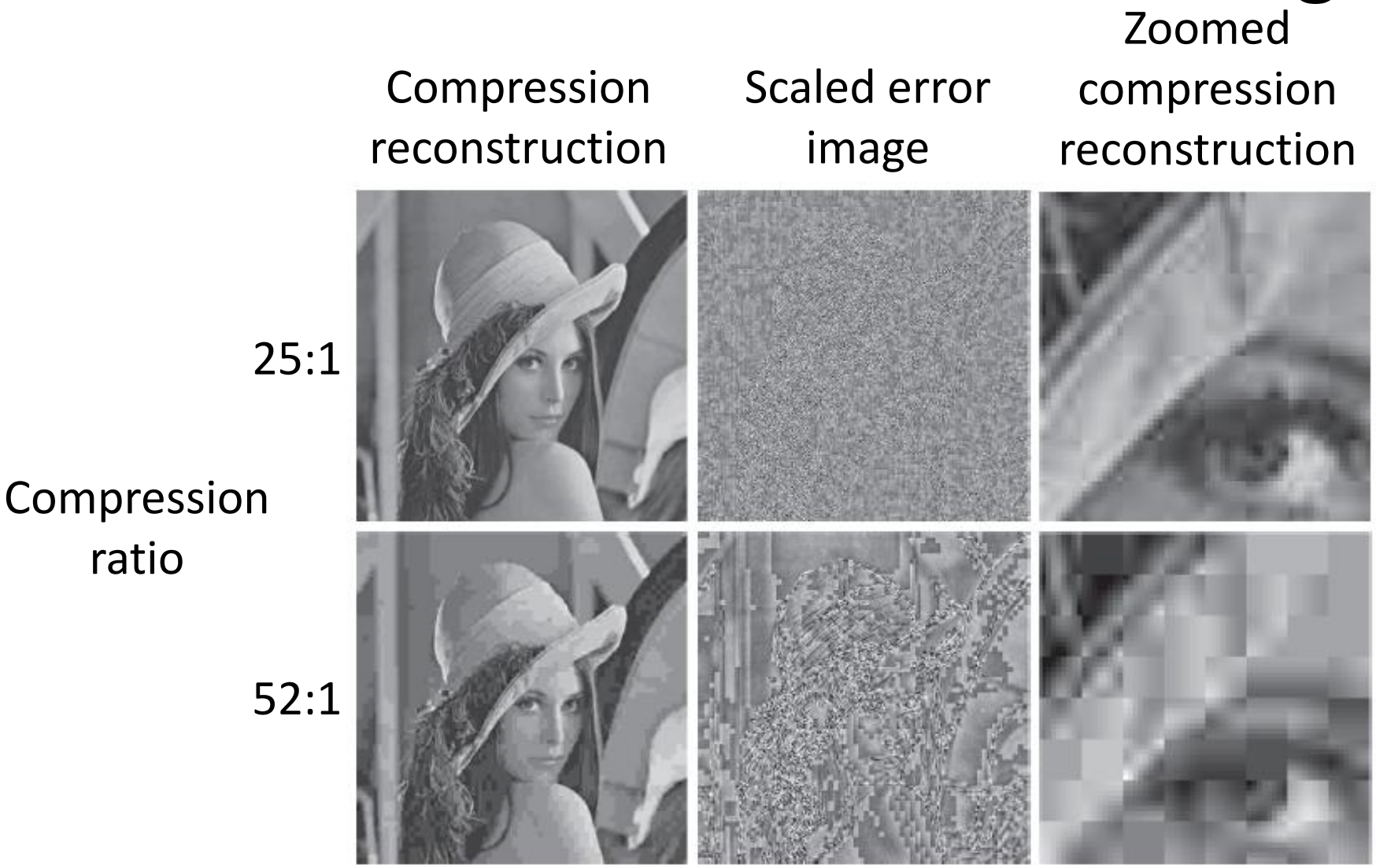
Reconstruction error versus subimage size





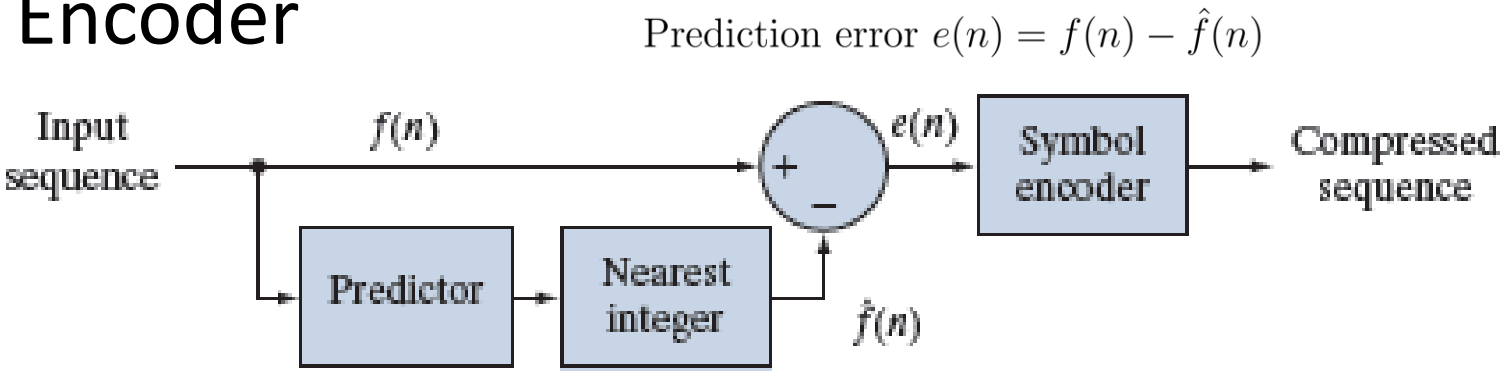
DCT subimage size: 2x2 4x4 8x8

JPEG uses block DCT-based coding

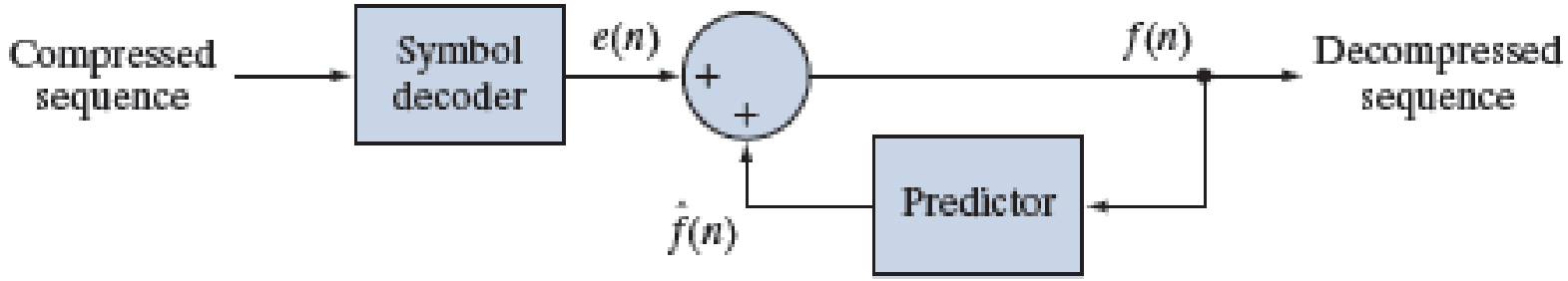


Predictive coding model

- Encoder



- Decoder



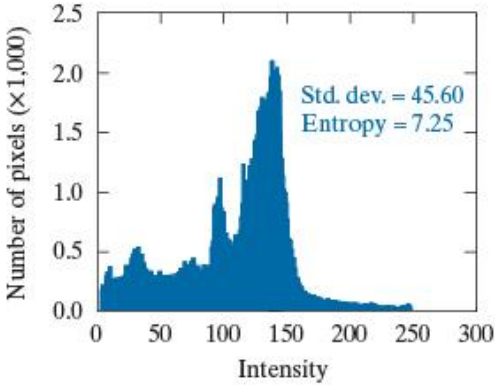
Predictive coding

Example: previous pixel coding $\hat{f}(x,y) = \text{round}[\alpha f(x, y-1)]$

Prediction coefficient, nominally = 1

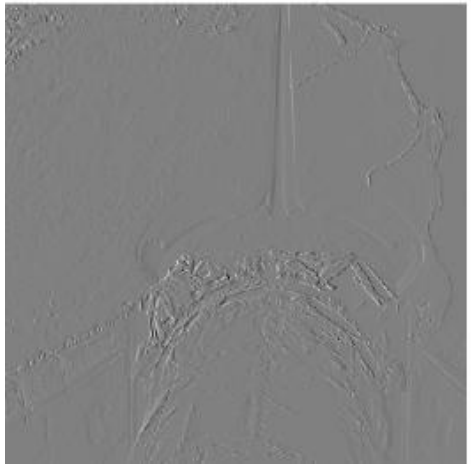
Input image

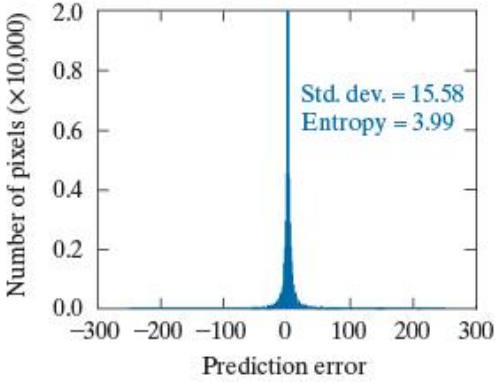




Histograms

Prediction error image





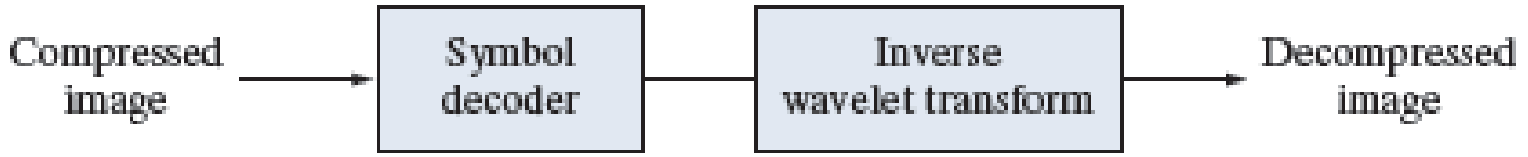
$e(x,y) = f(x,y) - \hat{f}(x,y)$

Wavelet coding

- Encoder



- Decoder



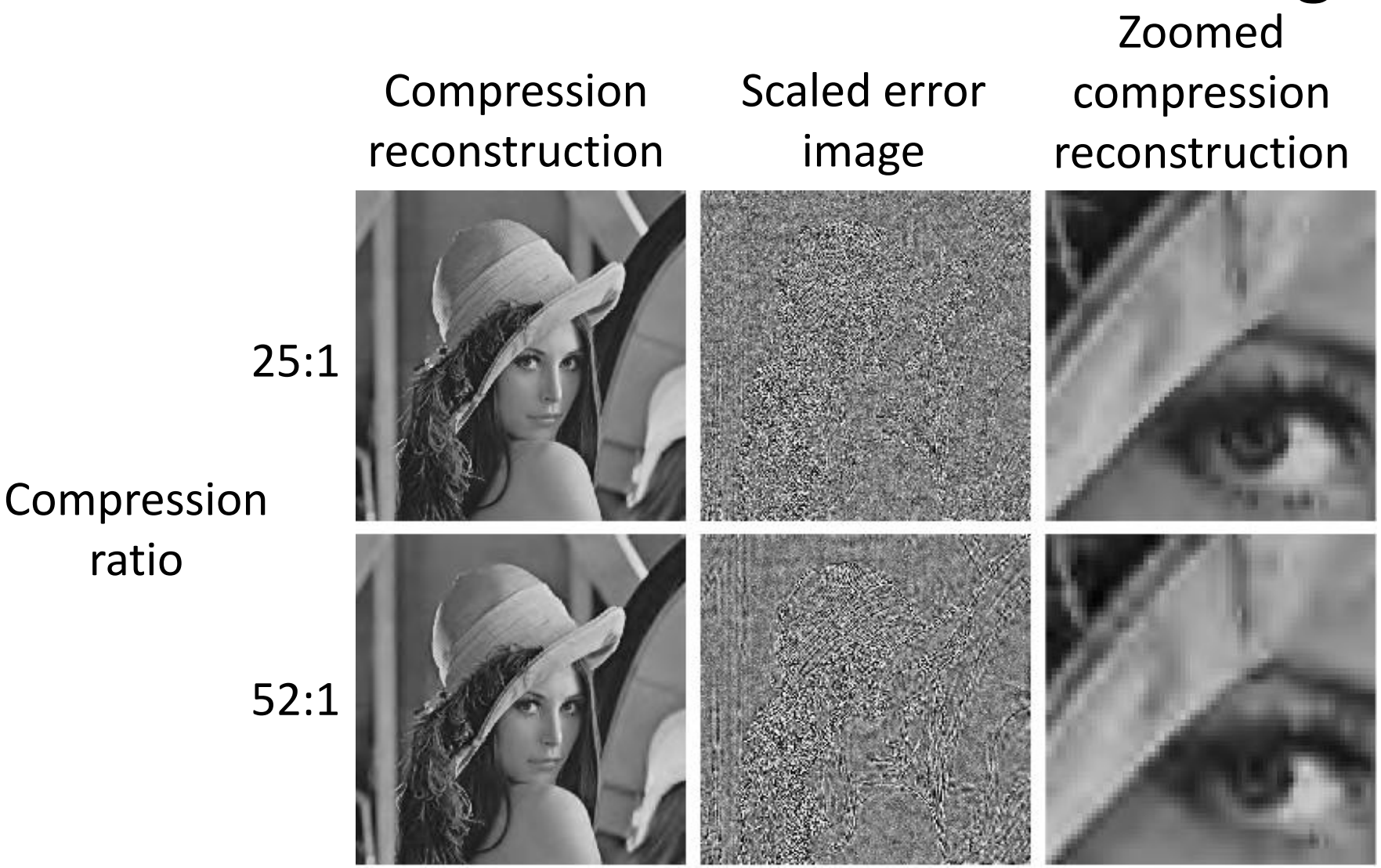
Wavelet coding

Detail coefficients below 25 are truncated to zero

TABLE 8.15 Decomposition level impact on wavelet coding the 512×512 image of Fig. 8.9(a).

Decomposition Level (Scales or Filter Bank Iterations)	Approximation Coefficient Image	Truncated Coefficients (%)	Reconstruction Error (rms)
1	256×256	74.7%	3.27
2	128×128	91.7%	4.23
3	64×64	95.1%	4.54
4	32×32	95.6%	4.61
5	16×16	95.5%	4.63

JPEG-2000 uses wavelet-based coding



JPEG-2000 uses wavelet-based coding

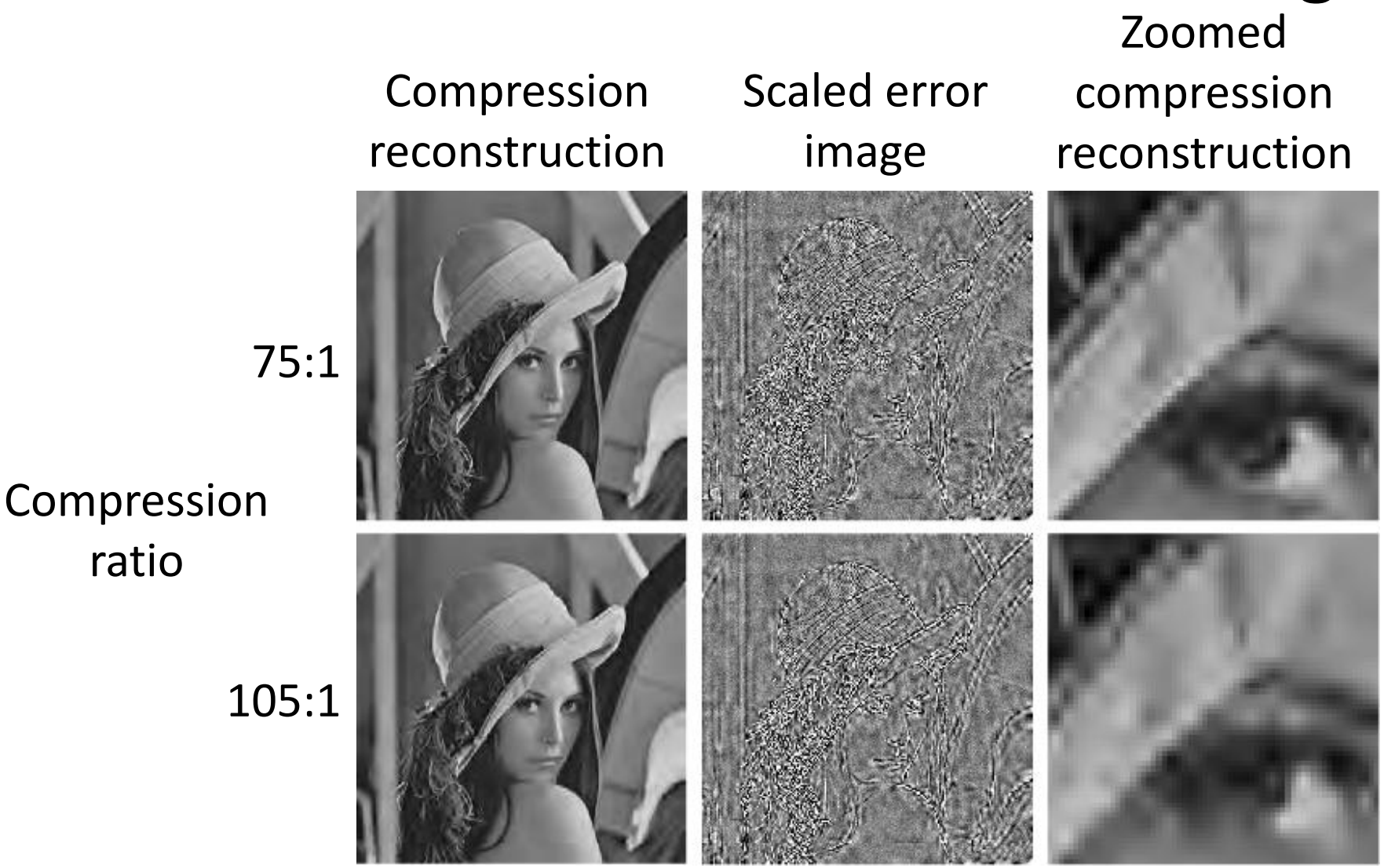


Image watermarking

- Visible watermarks
- Invisible watermarks

Visible watermark



Watermark



Watermarked image



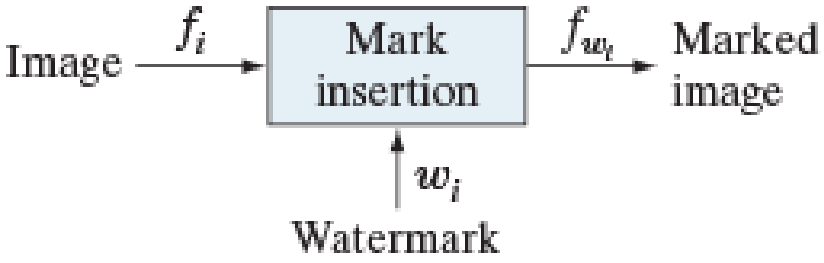
Original image minus watermark

Alpha blending

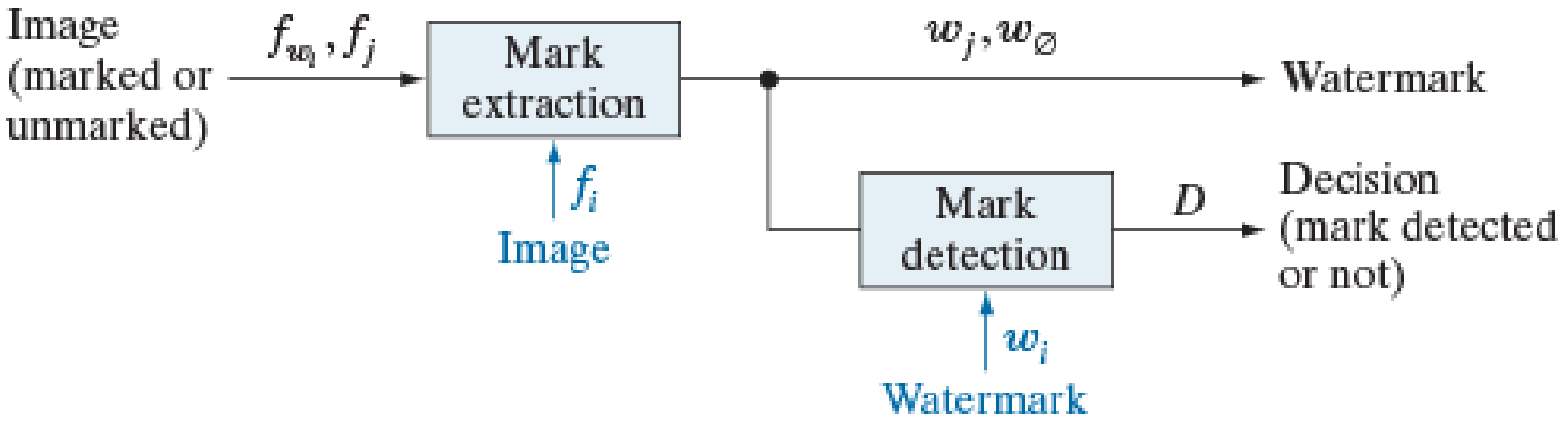
$f_w = (1 - \alpha) f + \alpha w$

Invisible image watermarking system

- Encoder



- Decoder



Invisible watermark

Example: watermarking using two least significant bits

Original image



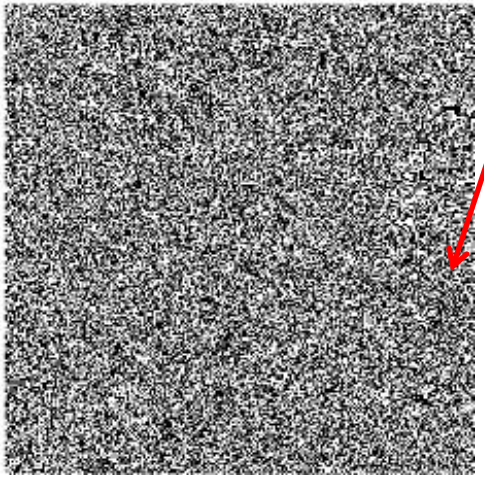
Extracted watermark



Two least significant bits

JPEG compressed





Fragile invisible watermark

Invisible watermark

- Encoder

1. Compute 2D DCT of image
2. Locate K largest magnitude coefficients $c_1, c_2, \ldots, c_K$ $\quad c_i \geq c_{i+1}$
3. Generate pseudo-random sequence of numbers $\omega_1, \omega_2, \ldots, \omega_K$ from Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$
4. Embed watermark

$$c_i' = c_i(1 + \alpha\omega_i) \qquad 1 \leq i \leq K$$

   where α is a positive constant that specifies how much c_i is altered. Replace c_i with c_i'.
5. Compute inverse DCT

Must store K largest magnitude coefficients $\mathbf{c}$, K-length pseudo-random sequence of numbers $\boldsymbol{\omega}$ (i.e., watermark), and α

Invisible watermark

- Decoder

1. Compute 2D DCT of image
2. Locate K largest magnitude coefficients $\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_K \qquad \hat{c}_i \geq \hat{c}_{i+1}$
3. Compute watermark

$$\hat{\omega}_i = \frac{\hat{c}_i - c_i}{\alpha c_i} \qquad 1 \leq i \leq K$$

4. Measure similarity using correlation coefficient

$$\gamma = \frac{\sum_{i=1}^{K} (\hat{\omega}_i - \bar{\hat{\omega}})(\omega_i - \bar{\omega})}{\sqrt{\left(\sum_{i=1}^{K} (\hat{\omega}_i - \bar{\hat{\omega}})^2\right)\left(\sum_{i=1}^{K} (\omega_i - \bar{\omega})^2\right)}} \qquad 1 \leq i \leq k$$

   where $\gamma \in [-1, 1]$ and is scale invariant

5. Decision: if $\gamma \geq$ threshold T, then watermark is present

Invisible watermark

Example: DCT-based watermarking

Watermarked images (different watermarks)



Extracted robust invisible watermark

Next Lecture

- Morphological image processing
- Reading
  - Chapter 9: Morphological image processing
    - Sections 9.1, 9.2, 9.3, and 9.5 (through subsection connected components)