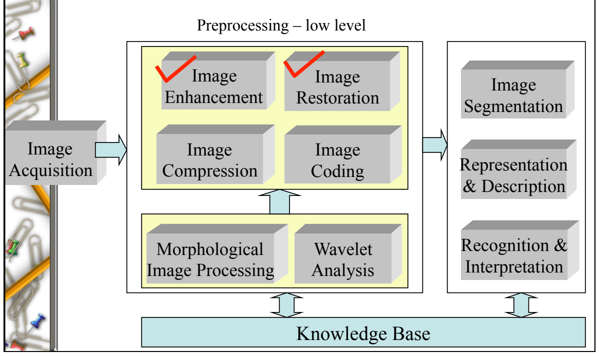
ECE472/572 - Lecture 11

Image Compression – Fundamentals and Lossless Compression Techniques

11/03/11

Roadmap



Roadmap

- Color image processing
  - Interpreting color
  - Pseudo-color IP
    - Intensity slicing
    - Transform in spatial domain
    - Transform in frequency domain
  - Full-color IP
    - Tonal correction
    - Color correction
    - Enhancement (I vs. RGB channels)
    - Restoration (I vs. RGB channels)
    - Color image edge detection
- Image compression
  - Data vs. information
  - Entropy
  - Data redundancy
    - Coding redundancy
    - Interpixel redundancy
    - Psycho-visual redundancy
  - Fidelity measurement
- Lossless compression
  - Variable length coding
    - Huffman coding
  - LZW
  - Bitplane coding
  - Binary image compression
    - Run length coding
  - Lossless predictive coding
- Lossy compression

Questions

- What is the difference between data and information?
- How to measure data redundancy?
- How many different types of data redundancy? Explain each.
- What is entropy? What is the first/second order estimate of entropy?
- Understand the two criteria that all coding mechanisms should satisfy.
- Understand Huffman coding
- How do you explain the average coding length from Huffman coding is always greater than the entropy?
- What image format uses which coding scheme?
- Understand RLC, differential coding

Data and Information

- Data and information
  - Different data set can represent the same kind of information
- Data redundancy
  - Relative data redundancy

$$R_D = 1 - \frac{1}{C_R} \qquad C_R = \frac{n_1}{n_2}$$

  - C_R: compression ratio
  - n_1, n_2: number of information carrying units in two data sets that represent the same information
- Data compression
  - Reducing the amount of data required to represent a given quantity of information

Data Redundancy

- Coding redundancy
- Interpixel redundancy
- Psychovisual redundancy

Coding Redundancy

- In general, coding redundancy is present when the codes assigned to a set of events (such as gray-level values) have not been selected to take full advantage of the **probabilities** of the events.
- In most images, certain gray levels are more probable than others

Coding Redundancy - Example

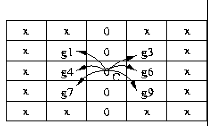
$$p_r(r_k) = \frac{n_k}{n}$$

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

r_k	$p_r(r_k)$	$Code1$	$l_1(r_k)$	$Code2$	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

Interpixel Redundancy

- Because the value of any pixel can be reasonably predicted from the value of its neighbors, much of the visual contribution of a single pixel to an image is redundant, it could have been guessed on the basis of its neighbors' values.
- Include spatial redundancy, geometric redundancy, interframe redundancy



Interpixel Redundancy - Example

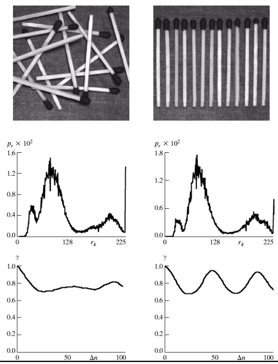
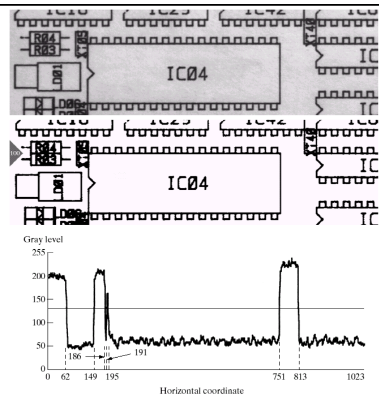


FIGURE 8.2 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.



Line 100: (1, 63) (0, 87) (1, 37) (0, 5) (1, 4) (0, 556) (1, 62) (0, 210)

FIGURE 8.3 Illustration of run-length coding: (a) original image. (b) Binary image with line 100 marked. (c) Line profile and binarization threshold. (d) Run-length code.

Psychovisual Redundancy

- The eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance than other information in normal visual processing
- In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings.

Psychvisual Redundancy

a b c

FIGURE 8.4 (a) Original image. (b) Uniform quantization to 16 levels. (c) IGS quantization to 16 levels.

Improved Gray Scale Quantization (IGS)

Evaluation Metrics

- Fidelity criteria
- Measure information
  - Entropy

Fidelity Criteria

- Objective fidelity criteria
  - Root-mean-square error (e_{rms})

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

  - Mean-square signal-to-noise ratio (SNR_{ms})

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

  - Root-mean-square SNR_{ms} (SNR_{rms})

Measure Information

- A random event E that occurs with probability $P(E)$ is said to contain

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

  - $I(E)$: **self-information** of E
  - The amount of self-information attributed to event E is inversely related to the probability of E
  - If base 2 is used, the unit of information is called **a bit**
  - Example: if $P(E) = ½$, then $I(E) = -\log_2 P(E) = 1$.

Entropy

- Average information per source output, or uncertainty, or **entropy** of the source

$$H(z) = -\sum_{j=1}^{J} P(a_j) \log P(a_j)$$

- How to interpret the increase of entropy?
- What happens if the events are equally probable?

Example

- Estimate the information content (entropy) of the following image (8 bit image):

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

Gray – level	Count	Probability
21	12	3/8
95	4	1/8
169	4	1/8
243	12	3/8

- **First-order estimate of the entropy (bits/ pixel)**

$$-3/8*\log 2(3/8) - 1/8*\log 2(1/8) - 1/8*\log 2(1/8) - 3/8*\log 2(3/8) = 1.81$$

Gray – level – pair	Count	Probability
(21,21)	8	2/7
(21,95)	4	1/7
(95,169)	4	1/7
(169,243)	4	1/7
(243,243)	8	2/7

- **Second-order estimate of the entropy (bits every two pixels)**

$$-2/7*\log 2(2/7)*2 - 1/7*\log 2(1/7)*3 = 2.2$$

What do these two numbers tell you?

Compression Approaches

- Error-free compression or lossless compression
  - Variable-length coding
  - Bit-plane coding
  - Lossless predictive coding
- Lossy compression
  - Lossy predictive coding
  - Transform coding

Variable-length Coding

- Only reduce code redundancy
- Assign the shortest possible code words to the most probable gray levels
- Huffman coding
  - Can obtain the optimal coding scheme

Revisit Example

r_k	$p_r(r_k)$	$Code1$	$l_1(r_k)$	$Code2$
$r_0 = 0$	0.19	000	3	00
$r_1 = 1/7$	0.25	001	3	1
$r_2 = 2/7$	0.21	010	3	0
$r_3 = 3/7$	0.16	011	3	01
$r_4 = 4/7$	0.08	100	3	10
$r_5 = 5/7$	0.06	101	3	11
$r_6 = 6/7$	0.03	110	3	000
$r_7 = 1$	0.02	111	3	001

Can we do this?

Huffman Coding

- **Uniquely decodable**
- **Instantaneous coding**
- Source reduction
  - Order the probabilities of symbols
  - Combine the lowest probability symbols into a single symbol that replaces them in the next source reduction
- Code each reduced source, starting with the smallest source and working back to the original source

Example

R1	0.25 (01)	0.25 (01)	0.25 (01)	0.25 (01)	**0.35** (00)	**0.4** (1)	**0.6** (0)
R2	0.21 (10)	0.21 (10)	0.21 (10)	0.21 (10)	0.25 (01)	0.35 (00)	0.4 (1)
R0	0.19 (11)	0.19 (11)	0.19 (11)	0.19 (11)	0.21 (10)	0.25 (01)	
R3	0.16 (001)	0.16 (001)	0.16 (001)	**0.19** (000)	0.19 (11)		
R4	0.08 (0001)	0.08 (0001)	**0.11** (0000)	0.16 (001)			
R5	0.06 (00000)	0.06 (00000)	0.08 (0001)				
R6	0.03 (000010)	**0.05** (00001)					
R7	0.02 (000011)						

Entropy of the source:

$-0.25*\log_2(0.25) - 0.21*\log_2(0.21) - 0.19*\log_2(0.19) - 0.16*\log_2(0.16)$
$-0.08*\log_2(0.08) - 0.06*\log_2(0.06) - 0.03*\log_2(0.03) - 0.02*\log_2(0.02)$
$= 2.6508$ bits/pixel

Average length of Huffman code:

$2*0.25 + 2*0.21 + 2*0.19 + 3*0.16 + 4*0.08 + 5*0.06 + 6*0.03 + 6*0.02$
$= 2.7$ bits

LZW

- Lempel-Ziv-Welch
- Attack interpixel redundancy - spatial redundancy
- Assigns fixed-length code words to variable length sequences of source symbols
- Requires no a-priori knowledge of the probability of occurrence of the symbols
- Used in GIF, TIFF, PDF
- Coding book is created while the data are being encoded

Example

39	39	126	126
39	39	126	126
39	39	126	126
39	39	126	126

TABLE 8.7 LZW coding example.

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126

Bit-plane Coding

- Attack inter-pixel redundancy
- First decompose the original image into bit-plane
- Binary image compression approach
  - run-length coding (RLC)

Bit-plane Decomposition

- Bit-plane slicing
- Problem: small changes in gray level can have a significant impact on the complexity of the bit plane
  - 127 vs. 128 ➔ 0111 1111 vs. 1000 0000
- Solution:

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le m-2$$

$$g_{m-1} = a_{m-1}$$

- Example:
  - 127 ➔ 0111 1111 ➔ 0100 0000
  - 128 ➔ 1000 0000 ➔ 1100 0000

Binary Image Compression - RLC

- Developed in 1950s
- Standard compression approach in FAX coding
- Approach
  - Code each contiguous group of 0's or 1's encountered in a left to right scan of a row by its length
  - Establish a convention for determining the value of the run
  - Code black and white run lengths separately using variable-length coding

A Complete Example

Original image	2 127 128 3 32 33								
Binary code	0000 0010 0111 1111 1000 0000 0000 0011 0010 0000 0010 0001								
XOR binary	0000 0011 0100 0000 1100 0000 0000 0010 0011 0000 0011 0001	Run-length coding							
8 bit planes	001 011 000 000 000 000 100 100 000 000 011 011 000 000 100 001	2 1 3	1 2 3	3 1 2	3 1 2	3 3	3 3	0 1 2 0 1 2	0 1 2 2 1
Huffman coding	1 0.34 (1) 0.66 (0) 2 0.33 (00) 0.34 (1) 3 0.33 (01)	3 1.00 (0)	0 0.34 (1) 1 0.33 (00) 2 0.33 (01)	1 2 0.4 (1) 0.6 (0) 2 2 0.4 (00) 0.4 (1) 0 1 0.2 (01)					
Final code	001 01	100 01	01 100	01 100	0 0	0 0	10001 10001	01100 001	

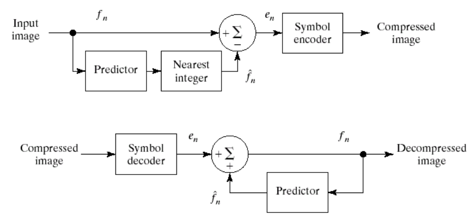
Lossless Predictive Coding

- Do not need to decompose image into bit planes
- Eliminate interpixel redundancy
- Code only the new information in each pixel
- The new information is the difference between the actual and predicted value of that pixel

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

$$\hat{f}(x,y) = round\left[\sum_{i=1}^{m} \alpha_i f(x, y-i)\right]$$

FIGURE 8.19 A lossless predictive coding model: (a) encoder; (b) decoder.



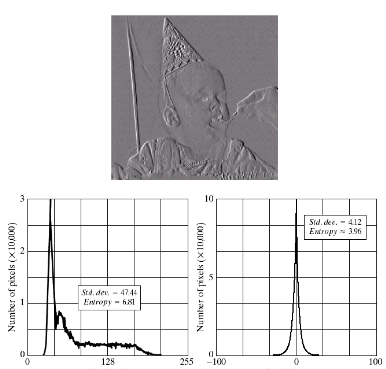
Previous Pixel Coding (differential coding)

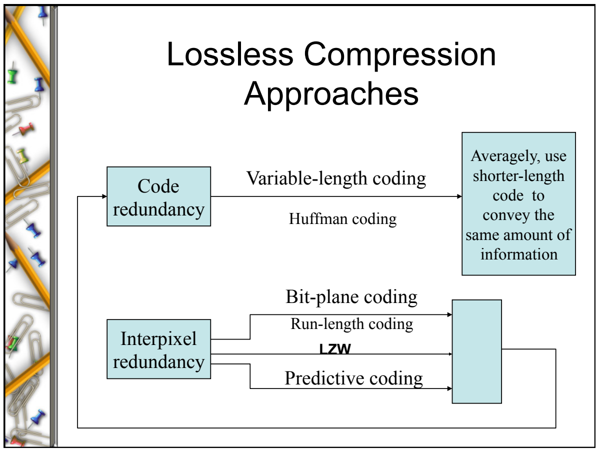
$$\hat{f}(x,y) = round\left[\alpha f(x, y-1)\right]$$

- First-order linear predictor / previous pixel predictor uses differential coding

126	127	128
126	1	1

FIGURE 8.20 (a) The prediction error image resulting from Eq. (8.4-9). (b) Gray-level histogram of the original image. (c) Histogram of the prediction error.




Lossless Compression
Approaches
Code
redundancy
Variable-length coding
Huffman coding
Averagely, use shorter-length code to convey the same amount of information
Interpixel
redundancy
Bit-plane coding
Run-length coding
LZW
Predictive coding