Basic concepts of image Compression technique: JPEG

Introduction

- To understand the need for compact image representation, consider the amount of data required to represent a 2 hour Standard Definition (SD) using 720 x 480 x 24 bit pixel arrays.
- A video is a sequence of video frames where each frame is a full color still image.
- Because video player must display the frames sequentially at rates near 30fps, SD video data must be accessed at

30fps x (720x480)ppf x 3bpp = 31,104,000 bps

fps – frames per second,
ppf – pixels per frame,
bpp – bytes per pixel & bps – bytes per second

Introduction

Thus a 2 hour movie consists of

31,104,000 bps x (60^2) sph x 2 hrs ≈ **2.24 x 10^{11} bytes.**

OR

224GB of data

sph = second per hour

- Twenty seven 8.5GB dual layer DVDs are needed to store it.
- To put a 2hr movie on a single DVD, each frame must be compressed by a factor of around 26.3.
- The compression must be even higher for HD, where image resolution reach 1920 x 1080 x 24 bits/image.

Data and Information

- *Data* is **not** the same thing as *information*.
- Data is the means with which information is expressed. The amount of data can be much larger than the amount of information.
- Data that provide no relevant information = *redundant data* or *redundancy*.

Image coding or compression has a goal to reduce the amount of data by reducing the amount of redundancy

Data Redundancy

Redundant data :

Representation that contain irrelevant or repeated information.

- $n1$ = data.
- $n2$ = data − redundancy (i.e., data after compression).
- **Compression ratio** = $CR = n1/n2$

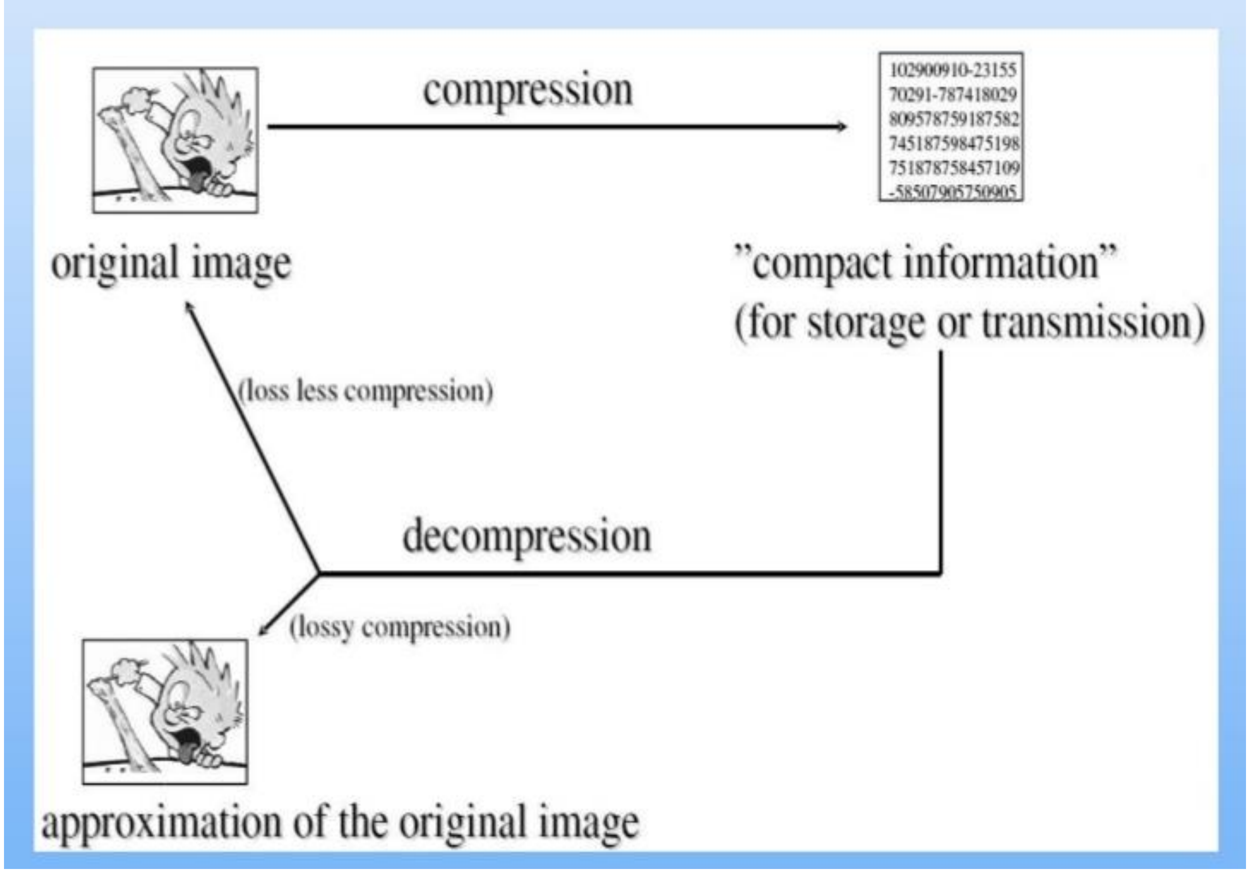
Relative redundancy = $RD = 1 - 1/CR$

What is image compression?

- Image compression refers to the process of redundancy amount of data required to represent the given quantity of information for digital image. The basis of reduction process is removal of redundant data.

Why image compression?

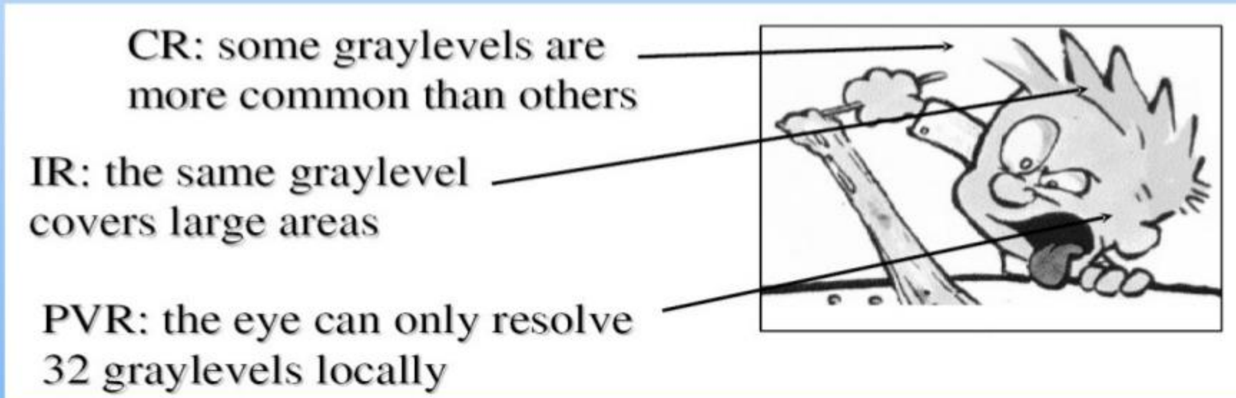
- Reducing the amount of data
- Reducing transmission time



Different types of Redundancy

- Three redundancies in 2-D arrays:
  - Coding redundancy
  - Inter-pixel redundancy
  - Irrelevant information

CR Coding Redundancy.
IR Interpixel Redundancy.
PVR Psycho-Visual Redundancy



Data Redundancy

TYPES OF DATA REDUNDANCY

- Three principal types of data redundancies that can be identified and exploited in digital images

1. Coding redundancy
2. Spatial or temporal (interpixel) redundancy
3. Psychovisual redundancy (irrelevant information)

- Data compression attempts to reduce one or more of these redundancy types.

Coding Redundancy

TYPES OF DATA REDUNDANCY

- **Coding redundancy**

The 8-bit codes that are used to represent the intensities in most 2-D intensity arrays contain more bits than are needed to represent the intensities.

- **Spatial or temporal (interpixel) redundancy**

Interpixel redundancy implies that pixel values are correlated (i.e., A pixel value can be reasonably predicted by its neighbors).

- **Psychovisual redundancy (irrelevant information)**

Most images contain information that is ignored by the human visual system and/or irrelevant to the intended use of the image. It is redundant in the sense that it is not used.

Coding Redundancy

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Inter-pixel Redundancy

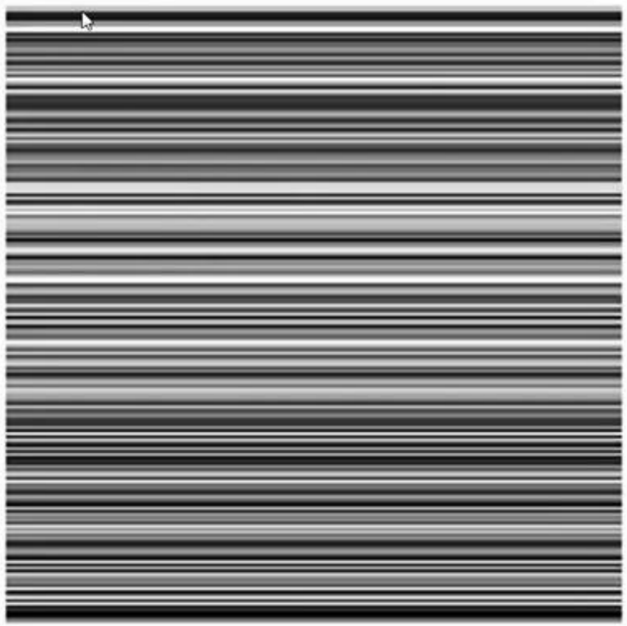
SPATIAL AND TEMPORAL REDUNDANCY

CONSIDER THE FOLLOWING IMAGE OF SIZE 256 × 256.
IN THE CORRESPONDING 2-D IMAGE:

256, 255

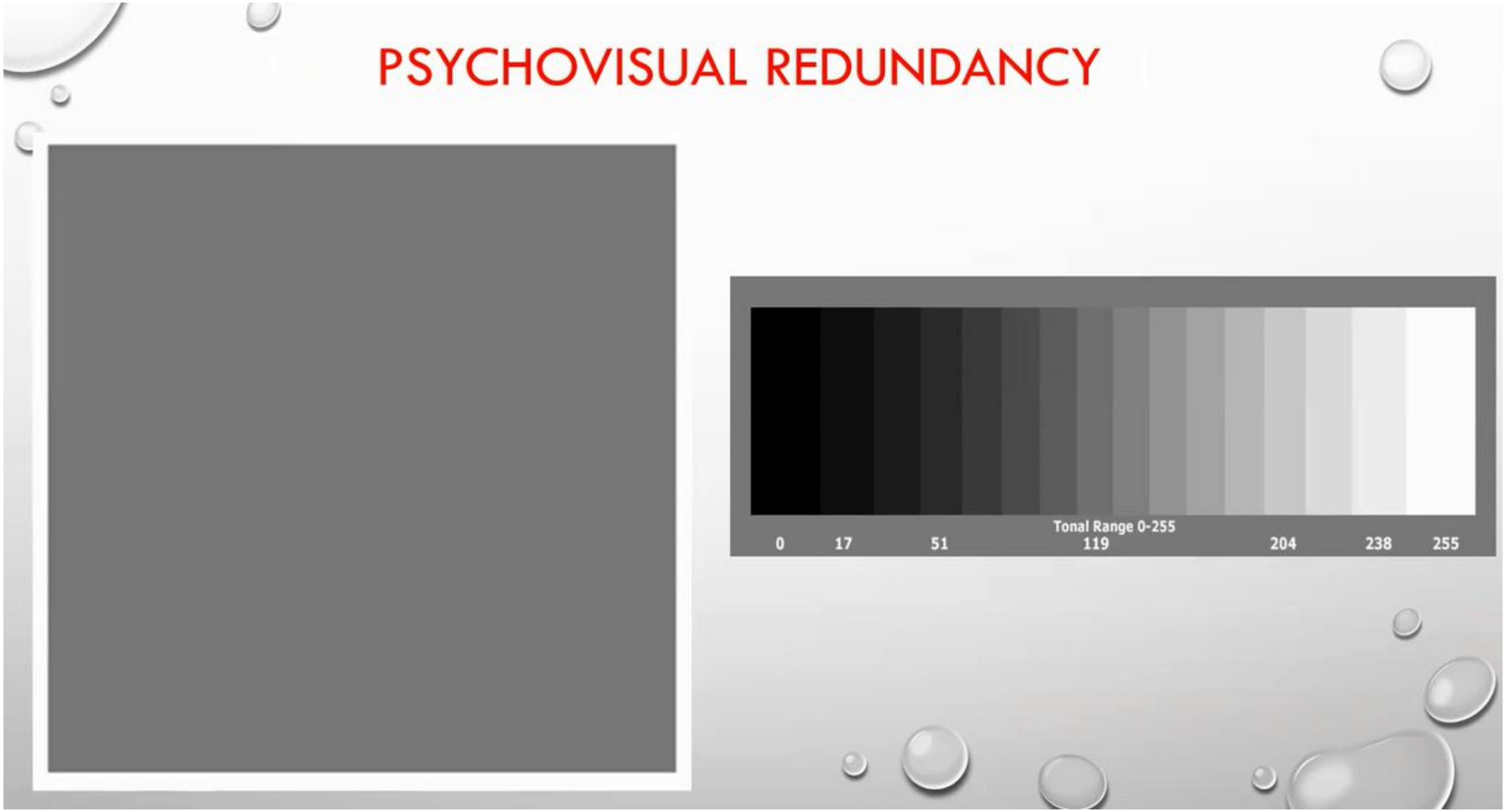
256, 0

256, 1



Irrelevant information

- *Information that not used by the human visual systems*



Coding Redundancy

CODING - DEFINITIONS

- **Code:** a list of symbols (letters, numbers, bits etc.)
- **Code word:** a sequence of symbols used to represent some information (e.g., Gray levels).
- **Code word length:** number of symbols in a code word.

Example: (binary code, symbols: 0,1, length: 3)

CODE	CODE WORD		
0:	000	4:	100
1:	001	5:	101
2:	010	6:	110
3:	011	7:	111

CODING REDUNDANCY

- $r_k \rightarrow$ Input Intensity Value e.g. 0 – 255 for grayscale image
- $l(r_k) \rightarrow$ No. of bits used to represent r_k
- Then average no of bits required to represent each pixel is

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

Say, uniform width str. (all pixels are using 8 bits to represent intensity)

→ equal length code (fixed length code)

→ m-bit fixed length code (8 bit fixed length code in our case)

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

CODING REDUNDANCY

- In this case, $l(r_k) = 8$ [each rq. 8 bits in fixed length]

$$L_{avg} = 8\sum_{k=0}^{L-1} p_r(r_k) = 8 * 1 = 8$$

- Coding redundancy tries to reduce L_{avg}
- Thus, total no. of bits required to represent an $M \times N$ image is MNL_{avg}
- The code can be an equal length code, or variable length code.

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Cont.

CODING REDUNDANCY

Code 1 → equal length code
Code 2 → variable length code
(obt by Huffman Coding)
Min bits → higher probability

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Cont.

CODING REDUNDANCY

- For code 1, $L_{avg} = 8$
- On the other hand, using code 2, the average length of the encoded pixels is

$$L_{avg} = 0.25(2) + 0.47(1) + 0.25(3) + 0.03(3) = 1.81 \text{ bits}$$

- The resulting compression and corresponding relative redundancy are

$$C = \frac{256 \times 256 \times 8}{256 \times 256 \times 1.81} \approx 4.42$$

$$R = 1 - \frac{1}{4.42} = 0.774$$

Thus, 77.4% of the data in the original 8-bit 2-D intensity array is redundant.

Compression Method

HUFFMAN CODING

- A measure to reduce coding redundancy
- Most popular coding redundancy technique
- Variable length code
- Min length code is assigned to one with highest probability

Cont.

HUFFMAN CODING

Say, Image size: 10 x 10 (5 bit image)
Frequency:
a2 = 40 a6 = 30 a1 = 10 a4 = 10 a3 = 6 a5 = 4
P(a2) = 40/100 = 0.4

Symbols (like intensity levels)	Probabilities (sorted)	Source Reduction (do till two values are left) (Maintain in sorted order here as well)			
		1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Cont.

Symbols (like intensity levels)	Probabilities (sorted)	Source Reduction (do till two values are left) (Maintain in sorted order here as well)			
		1	2	3	4
a2	0.4 1	0.4	0.4	0.4	0.6 0
a6	0.3 00	0.3	0.3	0.3 00	0.4 1
a1	0.1 011	0.1	0.2 010	0.3 01	
a4	0.1 0100	0.1 0100	0.1 011		
a3	0.06 01010	0.1 0101			
a5	0.04 01011				

Encoded String: 010100111100

Decoding : a3 a1a2 a2 a6

Parameters:

1. Average length of code
   Lavg = 0.4 * 1 + 0.3 * 2 + 0.1 * 3 + 0.1 * 4 + 0.06 * 5 + 0.04 * 5 = 2.2 bits/symbol
2. Total no. of bits to be transmitted
   10 * 10 * 2.2 = 220 bits
3. Entropy = 2.1396
4. How much you saved = $\dfrac{10 * 10 * 5 - 10 * 10 * 2.2}{10 * 10 * 5}$ = 0.56 = 56%

Reference

- Digital Image Processing, Rafael C.Gonzalez, Richard E.Woods, 3rd Edition.
- Digital Image Processing: Part II, Huiyu, Jiahua Wu, Jianguo Zha ng.
- Jpeg Image Compression Using Discrete Cosine Transform - A S urvey,A.M.Raid, W.M.Khedr, M. A. El-dosuky and Wesam Ahm ed, International Journal of Computer Science & Engineering Sur vey (IJCSES) Vol.5, No.2, April 2014.
- Tarek Ouni and Mohamed Abid , International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 5, No. 3, September, 2012 , Scan Methods and Their Application in Imag e Compression .