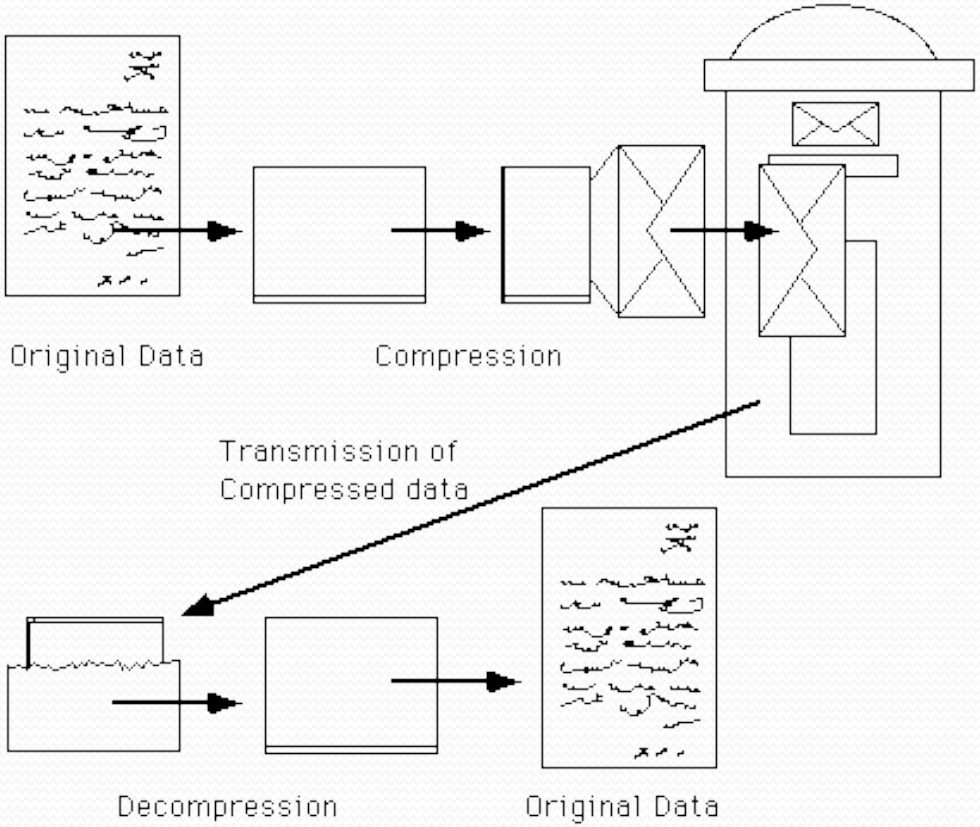
Image coding and compression

Computer Vision & Image Processing

Today

- Information and Data
- Redundancy
- Image Quality
- Coding
- Compression



Redundancy information and data

- Image coding และ Image compression เป็นการลดขนาดจำนวนข้อมูลของรูปภาพทำให้ไฟล์ภาพมีขนาดเล็กลง เพื่อสะดวกแก่การจัดเก็บข้อมูล หรือส่งข้อมูล แต่ขณะเดียวกันข้อมูลในภาพก็จะต้องไม่หายไป รายละเอียดต่าง ๆ ในภาพก็ยังคงถูกเก็บรักษาไว้เหมือนเดิม

- Image coding และ Image compression เป็นการเข้ารหัสของข้อมูลภาพ เพื่อให้ภาพมีขนาดในการจัดเก็บที่เล็กลง และยังสามารถนำข้อมูลของภาพคืนมาได้อย่างถูกต้อง หลังจากมีการถอดรหัสแล้ว

Definition

- n_1 = data
- n_2 = data – redundancy (data after compression)

- Compression ration = $C_R = \frac{n_1}{n_2}$

- Relative redundancy = $R_D = 1 - \frac{1}{C_R}$

Image coding & compression

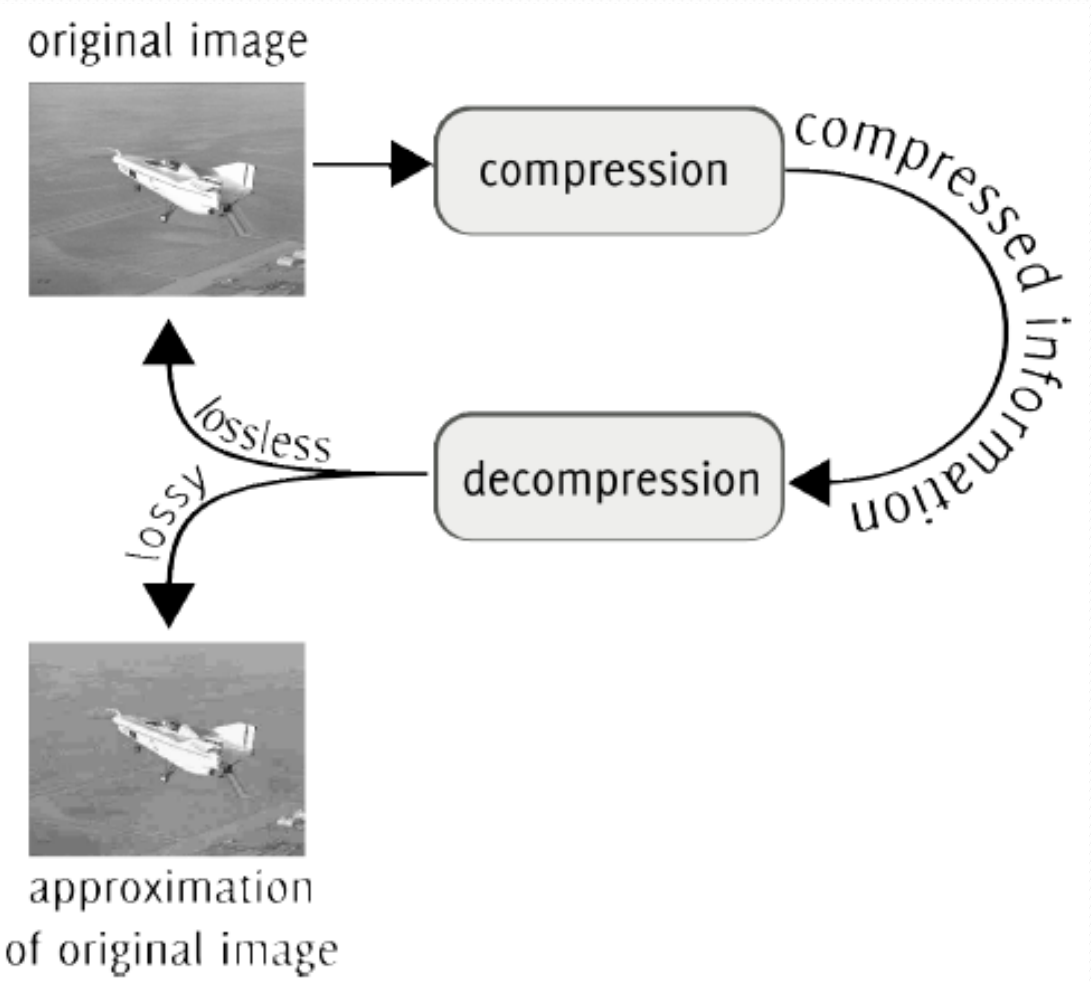


Image compression

- Lossless (Reversible) ไม่มีการสูญเสียข้อมูลในภาพ
  - ภาพที่ถูกบีบอัด(compression) เมื่อนำมาขยายให้เหมือนเดิมแล้ว (decompression) จะไม่มีการสูญเสียข้อมูลไป จำเป็นมากในขั้นตอนการทำการวิเคราะห์ภาพ
  - อัตราส่วนของการบีบอัดข้อมูล (compression ration) โดยทั่วไปจะมีค่า 2 ถึง 10 เท่า
- Lossy (Non reversible)มีการสูญเสียข้อมูลในภาพบางส่วน
  - ส่วนใหญ่ใช้ในการสื่อสารข้อมูล ,กล้อง compact, video, www etc.
  - อัตราส่วนของการบีบอัดข้อมูล (compression ration) โดยทั่วไปจะมีค่า 10 ถึง 30 เท่า

Image coding & compression

- Image coding
  - เป็นการแทนค่าข้อมูลภาพด้วย code ประเภทต่าง ๆ
- Image compression
  - ทำการลดขนาดของข้อมูลซึ่งมาได้จากวิธีการ Image coding
  - ทำให้การจัดเก็บ และการส่งข้อมูลภาพ มีประสิทธิภาพที่ดี

Objective Measures of Image Quality

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

- Total Error $$e_{tot} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\hat{f}(x,y) - f(x,y))$$

- Root-Mean-Square $$e_{RMS} = \frac{1}{MN} \sqrt{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\hat{f}(x,y) - f(x,y))^2}$$

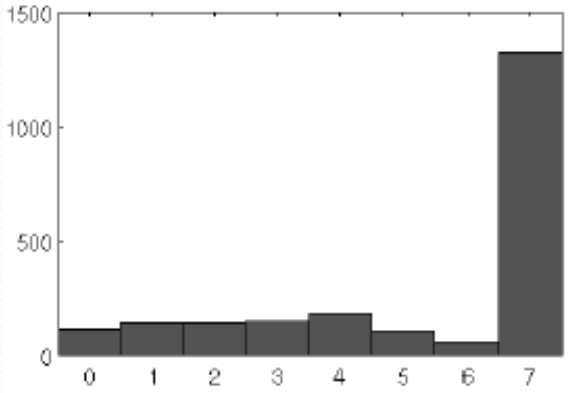
Measure the amount of data

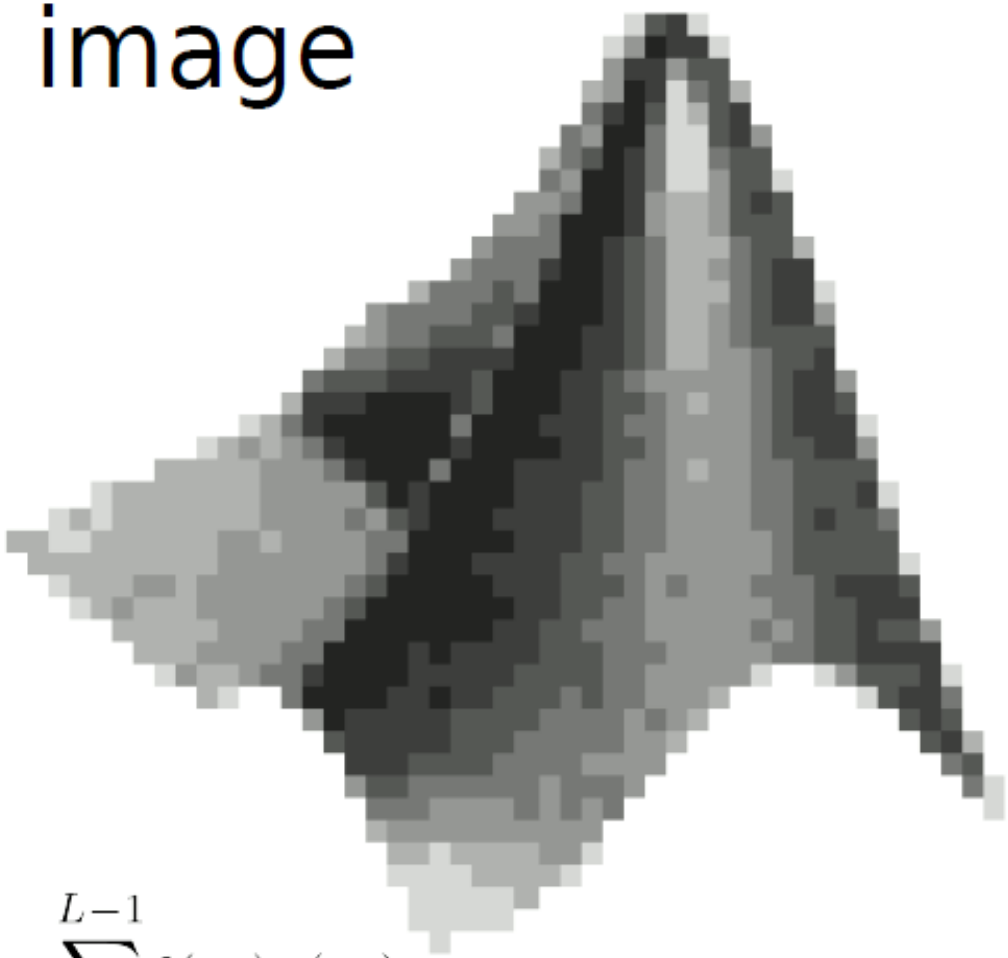
- ค่าเฉลี่ยของจำนวน bits แทนแต่ละ pixel ในภาพขนาด MxN นั้น มีจำนวน gray level (ระดับสีเทา) ซึ่งจะแทนด้วยตัวแปร L_{avg} ดังนั้น

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

- เมื่อ $l(r_k)$ คือจำนวน bit ของภาพในระดับ gray level นั้น ๆ
- และ $p(r_k)$ คือความน่าจะเป็น (probability) ของ gray level ที่เกิดขึ้นในภาพ
- จำนวน bits ที่ใช้สำหรับภาพขนาด MxN จะเท่ากับ MN * L_{avg}

Example 3-bit image





r_k	count	$p(r_k)$	source
0	113	0.051	000
1	139	0.063	001
2	142	0.064	010
3	145	0.066	011
4	181	0.082	100
5	105	0.047	101
6	52	0.023	110
7	1323	0.601	111

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

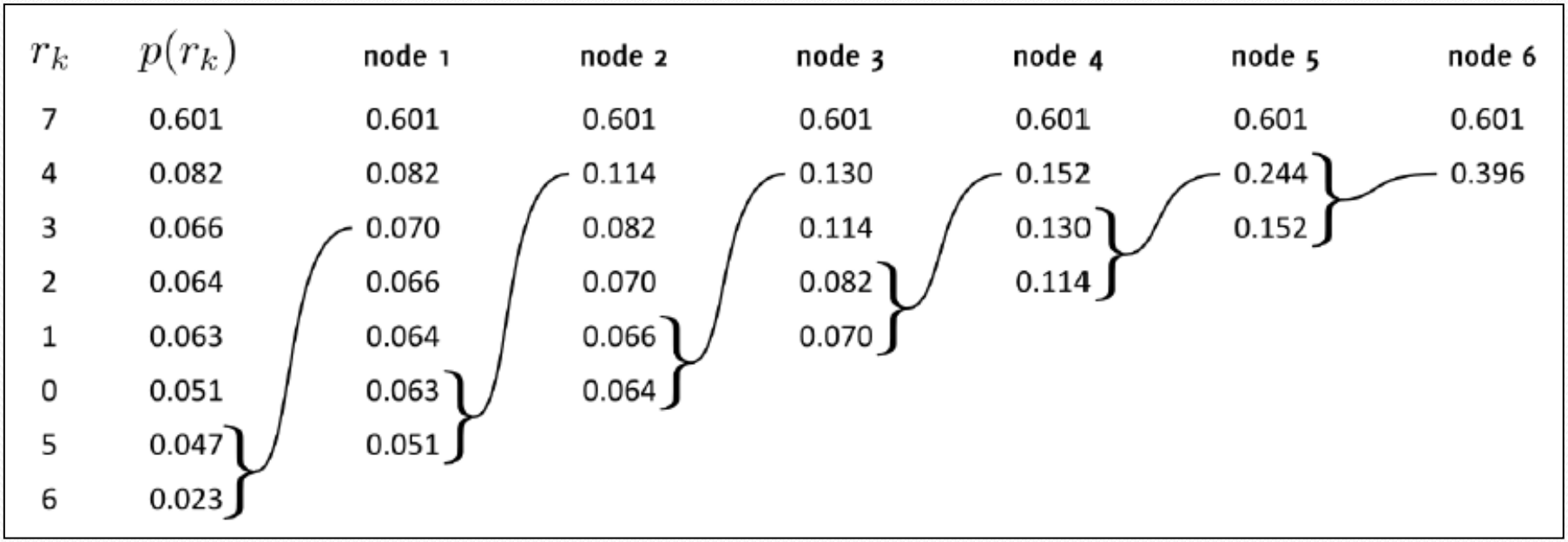
$$L_{avg} = 3$$

Huffman Coding

- First
  1. จัดเรียง probability ของ gray level จาก มากไปน้อย
  2. ทำการบวก probability ค่าที่น้อยที่สุด 2 ค่าเข้าด้วยกัน
  3. ทำการจัดเรียงค่าใหม่
  4. ทำซ้ำขั้นตอนที่ 1 ถึง 3 จนกระทั่งเหลือค่า probability แค่ 2 ค่า
- Second
  1. แทน code 0 สำหรับค่า probability สูงสุด และ code 1 สำหรับ probability ต่ำสุด ภายในคู่บวกใด ๆ
  2. ทำจากปลายทางไปต้นทาง และทำซ้ำตามข้อ 1 จนกระทั่งค่า สุดที่ต้นทางแล้วหยุดทำ

Example of Huffman coding

First (ขั้นตอนแรก)



r_k	$p(r_k)$	node 1	node 2	node 3	node 4	node 5	node 6
7	0.601	0.601	0.601	0.601	0.601	0.601	0.601
4	0.082	0.082	0.114	0.130	0.152	0.244	0.396
3	0.066	0.070	0.082	0.114	0.130	0.152	
2	0.064	0.066	0.070	0.082	0.114		
1	0.063	0.064	0.066	0.070			
0	0.051	0.063	0.064				
5	0.047	0.051					
6	0.023						

Huffman Coding

- First
  1. จัดเรียง probability ของ gray level จาก มากไปน้อย
  2. ทำการบวก probability ค่าที่น้อยที่สุด 2 ค่าเข้าด้วยกัน
  3. ทำการจัดเรียงค่าใหม่
  4. ทำซ้ำขั้นตอนที่ 1 ถึง 3 จนกระทั่งเหลือค่า probability แค่ 2 ค่า
- Second
  1. แทน code 0 สำหรับค่า probability สูงสุด และ code 1 สำหรับ probability ต่ำสุด ภายในคู่บวกใด ๆ
  2. ทำจากปลายทางไปต้นทาง และทำซ้ำตามข้อ 1 จนกระทั่งค่า สุดที่ต้นทางแล้วหยุดทำ

Example of Huffman coding
Assigning codes

Second (ขั้นตอนที่สอง)

r_k	$p(r_k)$	code	node 1		node 2		node 3		node 4		node 5		node 6	
7	0.601	0	0.601	0	0.601	0	0.601	0	0.601	0	0.601	0	0.601	0
4	0.082	110	0.082	110	0.114	101	0.130	100	0.152	11	0.244	10	0.396	1
3	0.066	1000	0.070	111	0.082	110	0.114	101	0.130	100	0.152	11		
2	0.064	1001	0.066	1000	0.070	111	0.082	110	0.114	101				
1	0.063	1010	0.064	1001	0.066	1000	0.070	111						
0	0.051	1011	0.063	1010	0.064	1001								
5	0.047	1110	0.051	1011										
6	0.023	1111												

Example of Huffman coding

r_k	$p(r_k)$	code
7	0.601	0
4	0.082	110
3	0.066	1000
2	0.064	1001
1	0.063	1010
0	0.051	1011
5	0.047	1110
6	0.023	1111

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k)p(r_k) = 2.10$$

(Before Huffman coding $L_{avg} = 3$)

$$C_R = \frac{n_1}{n_2} = \frac{3}{2.10} = 1.43$$

$$R_D = 1 - \frac{1}{C_R} = \frac{n_1 - n_2}{n_1} = \frac{3 - 2.10}{3} = 0.3$$

MN*L_{avg} (ก่อน huffman) = 2200*3 = 6600 bits

MN*L_{avg} (หลัง huffman) = 2200*2.1 = 4620 bits

Huffman Coding

- วิธีการ Huffman coding เมื่อมีการ Decompression เพื่อนำรูปภาพมาแสดง จะไม่มีการสูญหายของข้อมูล ภายในรูปภาพ (lossless)

- ตารางการแปลงของ Huffman coding จะต้องถูกเก็บควบคู่ไปกับ code ของไฟล์ภาพด้วย เพื่อทำการแปลงค่ากลับให้ถูกต้อง

Homework

rk	count
0	250
1	1200
2	900
3	2100
4	355
5	450
6	120
7	780

- จาก ค่า gray level และจำนวนของ gray level ที่ให้ จงแสดงการวิธีการทำ Huffman coding และหาค่า L_{avg} (ก่อนทำ) ,L_{avg} (หลังทำ) , CR, RD , $MN*L_{avg}$ (ก่อน huffman) , $MN*L_{avg}$ (หลัง huffman)