8/8

LZW Coding

dictionary created

e.g., for 256 gray levels, 1st 256 words are

0, 1, ..., 255

sequentially examine sequences, if not in dictionary

⇒ place them.

Suppose 9-bit dictionary (512 words)

Given 1×2 image : | 255 | 255 |

~~takes~~ 2 * 8 = 16 bits

using table :

outputs : 256

1 * 9 = 9 bits

0	0
⋮	⋮
255	255
256	255-255
257	—
⋮	⋮
511	—

LZ Coding

Construct dictionary:

0 - 255 gray level symbols

as scan ~~sequence~~ input, sequences not in dictionary are added, then next time that sequence occurs, ~~use~~ longest fit dictionary index to represent it.

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

39
output → 39
dictionary → 256 39-39

39 39
output → 39
dictionary → 257 39-126

39 39 126
output → 126
dictionary 258 126-126

39 39 126 126
output → 126
dictionary → 259 126-39

39 39 126 126
39

39 39 126 126
39 39
output → 256
dictionary → 260 39-39-126

39 39 126 126
39 39 126

8/10

Run-length Coding

give value + ~~number~~ length of run

BMP : encoded

byte 1	byte 2
# pixels	color index

absolute mode

1st byte is 0

2nd byte : 0 end of line

: 1 end of image

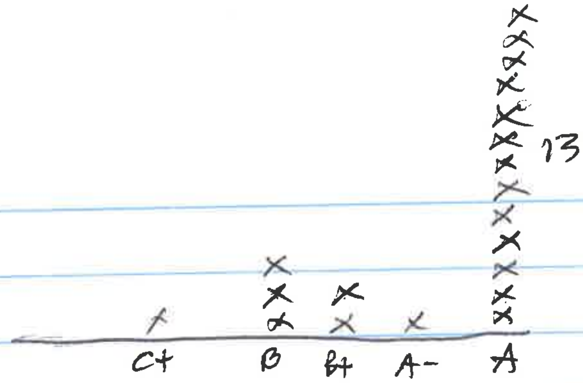
: 2 next 2 bytes contain unsigned horizontal + vertical offsets to next pixel

: 3 – 255 is number of uncompressed pixels that follow.

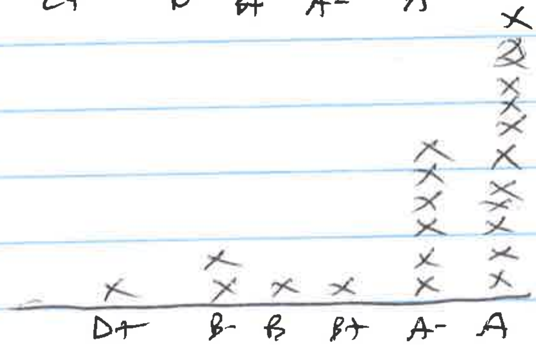
```
function ime = cs4640_RLE(im)

~~ime(1) =~~
im1 = im(:); len_im1 = length(im1);
first ~~start~~ = 1;
~~last = 0;~~

while first <= len_im1
   val = im1(first);
   ~~last = first;~~ last = first;
   ~~while im1(last+1~~
   if first == len_im1
      ime(end+1) = 1;
      ime(end+1) = val;
      return
   end
   while im1(last+1) == val & last < len_im1
      last = last + 1;
   end
   ime(end+1) = last - first + 1;
   ime(end+1) = val;
   first = last + 1;
end
```

week 8

* Quiz 5



* Assign 3



* no $\sqrt{\ }$ in power spectrum
* make sure images have something visible
  use log(1 + im) ?

Consider: $\sum_{k=0}^{255} N^2 \cdot 8 \cdot p(k) = N^2 \cdot 8$ # bits

info? $H = -\sum_{k=0}^{255} p(k) \log_2(p(k))$

how does information change when compressed?

$$\begin{bmatrix} 1 & 2 & 3 & 3 \\ 2 & 2 & 4 & 3 \\ 2 & 2 & 3 & 4 \\ 1 & 2 & 3 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

a_i	$p(a_i)$
1	1/8
2	6/8
3	6/8
4	1/8

if preserve the prob's, then info content is constant

a_i	
0	1/2
1	1/2

$H = -\left(\cancel{2}\cdot\frac{1}{\cancel{8}4}\cdot\log_2\frac{1}{8} + \cancel{2}\cdot\frac{6}{\cancel{8}4}\cdot\log_2\frac{6}{8}\right)$

$\left(-\frac{3}{4} + -\frac{3}{2} \times 0.415\right)$

$= \frac{3}{4} + \frac{3\cdot 0.415}{2}$ 1.3726

$H = -\left(\frac{1}{2}(-2) + \frac{1}{2}(-2)\right)$

$= 2$

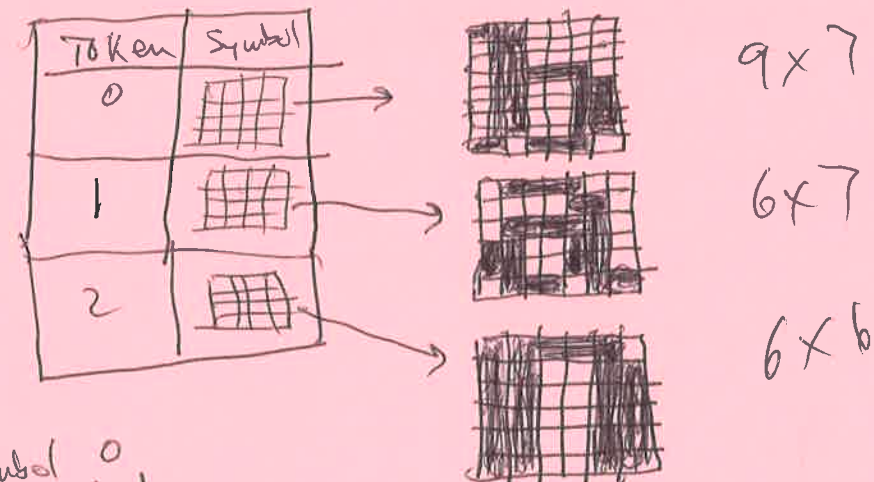
entropy went up

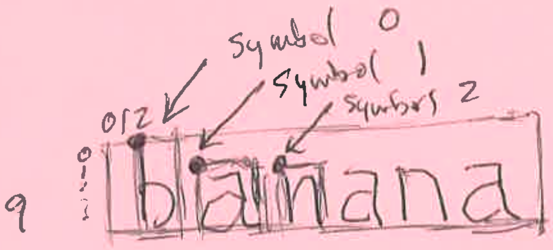
8/11

Symbol-based coding p. 628

Image → collection of subimages : symbols

symbol → symbol dictionary





51

459 bits

vs.

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

← 3 bytes

6x3x8 table

9x7

6x7

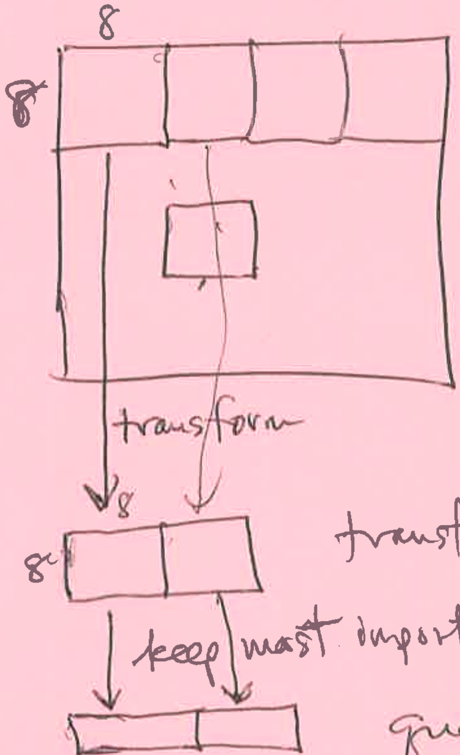
6x6

258 bits

C = 1.61

might use code books of sub images & match subwindows to best fit & use that (lossy)

Block Transform Coding p.632



Divide image into blocks

transform coefficients

keep most important components (high magnitude power)

quantized + coded

transforms → FT, DCT

choice depends on error tolerance

Recall that: f is $N \times N$

$$\underbrace{f(x,y) = \frac{1}{MN} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{N} + \frac{vy}{N}\right)}}$$

$$s(x,y,u,v) = e^{j2\pi\left(\frac{ux}{N} + \frac{vy}{N}\right)}$$

over the whole image, this is:

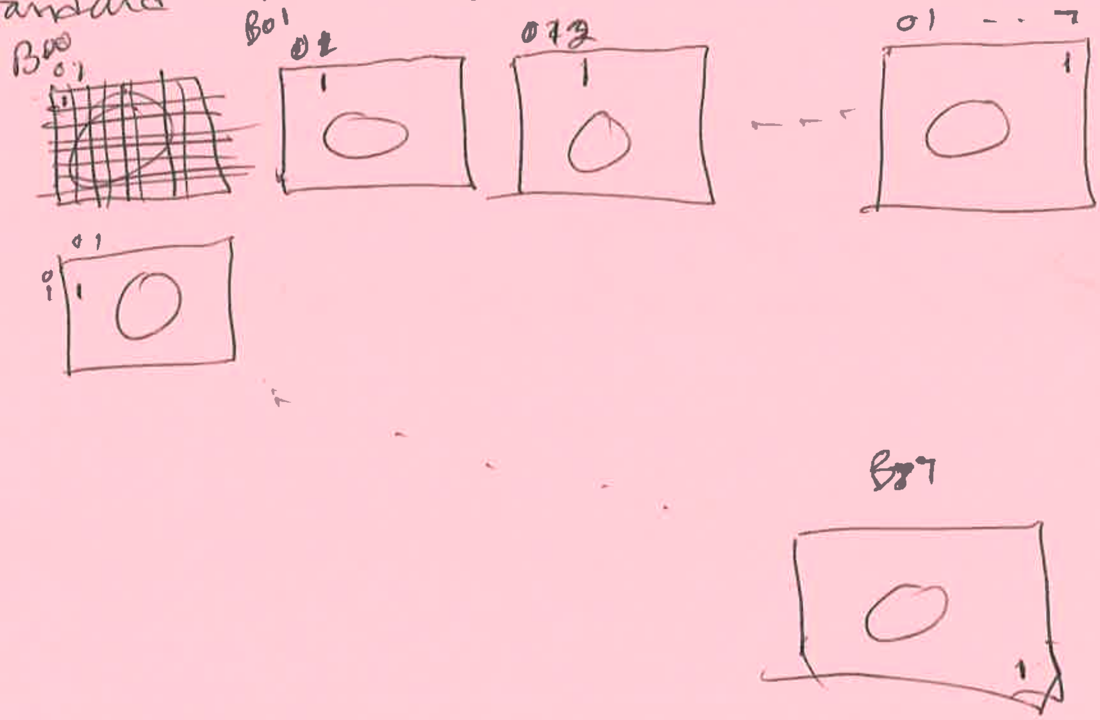
$$\underset{N \times N}{G} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \underset{\text{Transform coefficient}}{T(u,v)}\, \underset{N \times N}{S_{uv}}$$

(lenna g(1:8,1:8))

Consider an 8x8:

$$W = \begin{matrix} 135 & 134 & 133 & 131 & 130 & 128 & 127 & 126 \\ 134 & & & & & & & \\ 134 & & & & & & & \\ 133 & & & \cdots & & & & \\ 132 & & & & & & & \\ 131 & & & & & & & \\ 130 & & & & & & & 121 \\ 130 & & & & & & & \end{matrix}$$

standard basis images: (8x8)



$$W = \sum_{x=0}^{7} \sum_{y=0}^{7} gl(x,y)\, B_{xy}$$

$$= 135\, B_{00} + 134\, B_{01} + \dots + 121\, B_{77}$$

other basis image sets:

FT

DCT

8/15

FT $\rightarrow$ CS4640_Suv_FT

CS4640_week8 76 – 88

DC

$A(x,y,u,v) = \alpha(u)\,\alpha(v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$

CS4640_week8 90 – 103

---

and these ~~may~~ be separable, e.g., $\underline{FT}$

let $\bar{S}_u = \begin{array}{c} S(x,u) \\ \begin{bmatrix} S(0,u) \\ S(1,u) \\ \vdots \\ S(N-1,u) \end{bmatrix} \end{array}$ $\quad \bar{S}_v = \begin{array}{c} S(y,v) \\ \begin{bmatrix} S(0,v) \\ S(1,v) \\ \vdots \\ S(N-1,v) \end{bmatrix} \end{array}$

$\bar{S}_u(x) : S(x,u) = \frac{1}{\sqrt{N}} e^{j2\pi \frac{ux}{N}}$

$S_v(y) : S(y,v) = \frac{1}{\sqrt{N}} e^{j2\pi \frac{vy}{N}}$

$S_{u,v} = \bar{S}_u \bar{S}_v^T$

$= N \begin{bmatrix} \\ \\ \end{bmatrix}^{1} \times {}^{1}\begin{bmatrix} & N & \end{bmatrix} = \begin{bmatrix} S(0,0,u,v) & S(0,1,u,v) \dots & S(0,N-1,u,v) \\ S(1,0,u,v) & & \\ \vdots & & \\ S(N-1,u,v) & & S(N-1,N-1,u,v) \end{bmatrix}$

$= N \times N$

8/16

DCT

$$S(x,u) = \alpha(u) \cos\left(\frac{(2x+1)u\pi}{N}\right)$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u > 0 \end{cases}$$

DCT_analysis

CS4640-week8 105-113

JPEG

based on DCT   but has lossless option

input/output data: 8 bits

DCT values : 11 bits

2,6

$$\bar{s}_u = \begin{array}{c} x, u \\ \begin{bmatrix} s(0,u) \\ s(1,u) \\ \vdots \\ s(N-1,u) \end{bmatrix} \end{array} \qquad \frac{1}{\sqrt{N}} e^{j 2\pi \left(\frac{ux}{\cancel{N}}\right)}$$

$$s_v = \begin{array}{c} y, v \\ \begin{bmatrix} s(0,v) \\ s(1,v) \\ \vdots \\ s(N-1,v) \end{bmatrix} \end{array} \qquad \frac{1}{\sqrt{N}} e^{j 2\pi \left(\frac{vy}{N}\right)}$$

$$s(x,u) = \frac{1}{\sqrt{N}} e^{\frac{j 2\pi u x}{N}}$$

$$S_{u,v} = s_u s_v^T$$

$$S_{u,v} = \begin{bmatrix} s(0,0,u,v) & s(0,1,u,v) & \cdots & s(0,N-1,u,v) \\ s(1,0,u,v) & & & \\ \vdots & & & \\ s(N-1,u,v & & & \end{bmatrix}$$

FT $s(x,y,u,v) = \frac{1}{N} e^{j 2\pi \left(\frac{ux}{N} + \frac{vy}{N}\right)}$

DCT $s(x,u) = \alpha(u) \cos\left(\frac{(2x+1)u\pi}{2N}\right)$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u=0 \\ \sqrt{\frac{2}{N}} & u= \\ & 1 \cdots N- \end{cases}$$

$$S_u^{*T} S_v = \begin{cases} 0 \\ 1 \end{cases}$$