University of Nottingham
UK | CHINA | MALAYSIA

COMP-2032

Introduction to Image Processing

Lecture 10

Image Compression

Learning Outcomes

IDENTIFY

1. Redundancy
2. Huffman Coding
3. Psychovisual Redundancy - GIF
4. A Compression System - JPEG

University of Nottingham
UK | CHINA | MALAYSIA

Redundancy

Image Compression

Individual image(s)

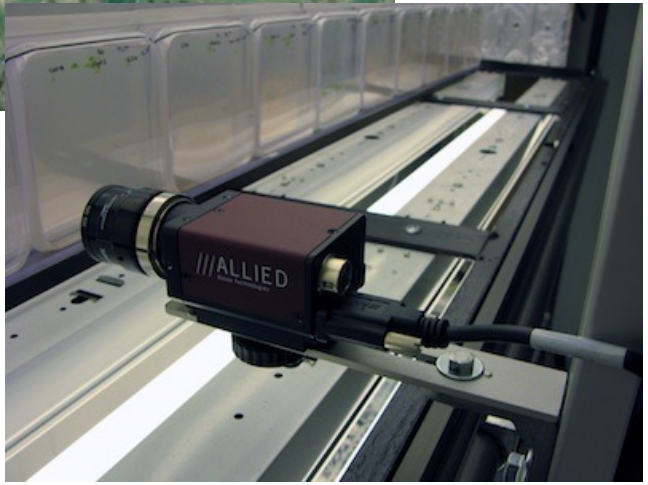
Can be large

Are easy to acquire, collections increases rapidly

In some applications, images are gathered automatically

Luckily, image data is redundant in several ways

- Coding redundancy
- Spatial redundancy
- Psychovisual redundancy



ACK: Prof. Tony Pridmore, UNUK

Coding Redundancy

The grey level histogram of an image gives the probability (frequency) of occurrence of grey level r_k

$$p(r_k) = \frac{n_k}{n} \qquad k = 0,2,\ldots,L-1$$

If the number of bits used to represent each value of r_k is $l(r_k)$, the average number of bits required to represent a pixel is

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

To code an MxN image requires MNL_{avg} bits

ACK: Prof. Tony Pridmore, UNUK

Coding Redundancy

If an m-bit natural binary code is used to represent grey level then

- All pixels take the same amount of space,
- $P(r_k)$ value sum to 1, so:

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

- And an image occupies MN_m bits

But some pixel values are more common than others…

ACK: Prof. Tony Pridmore, UNUK

Variable Length Encoding

Assigning fewer bits to the more probable grey levels than to less probable ones can achieve data compression, e.g.:

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

Build a codebook, replace 'true' pixel values with code

Lossless

The process can be reversed by inverting the codebook

ACK: Prof. Tony Pridmore, UNUK

Spatial Redundancy

- Sometimes called *Interpixel Redundancy*
- Neighbouring pixels often have similar values
- Compression based on spatial redundancy involves some element of pixel grouping or transformation
- Simplest is **Run-Length Encoding**

• Maps the pixels along each scan line into a sequence of pairs $(g_1, r_1), (g_2, r_2), \ldots,$
• Where g_j is the ith grey level, r_j is the run length of ith run

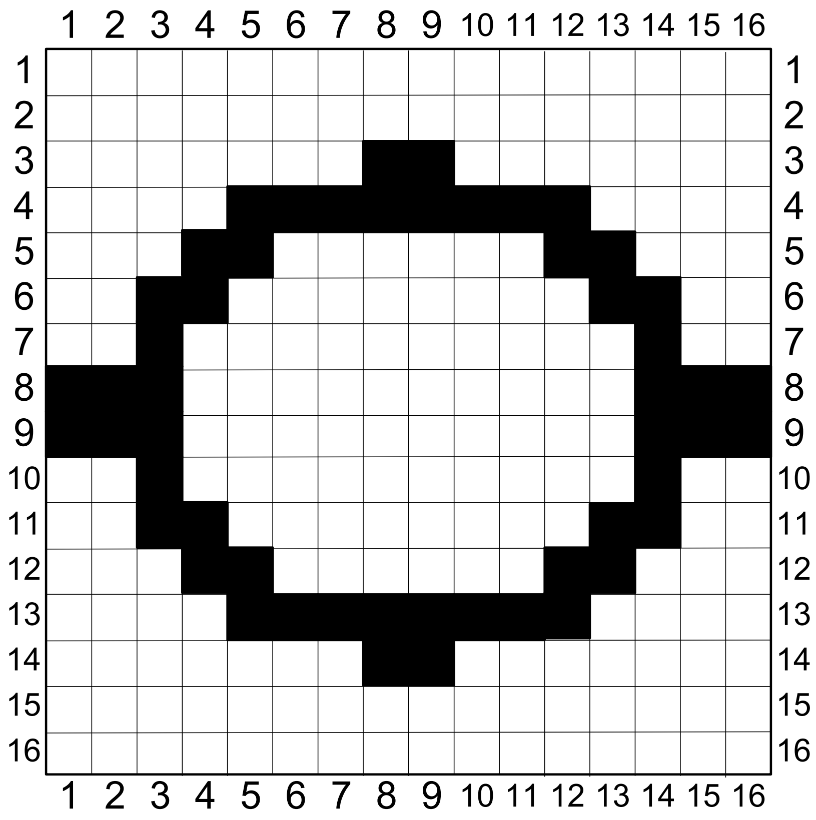
ACK: Prof. Tony Pridmore, UNUK

A Binary Example

Row 1: (0, 16)
Row 2: (0, 16)
Row 3: (0, 7) (1, 2) (0, 7)
Row 4: (0, 4), (1, 8) (0, 4)
Row 5: (0, 3) (1, 2) (0, 6) (1, 3) (0, 2)
Row 6: (0,2) (1, 2) (0,8) (1, 2) (0, 2)
Row 7: (0, 2) (1,1) (0, 10) (1,1) (0, 2)
Row 8: (1, 3) (0, 10) (1,3)
Row 9: (1, 3) (0, 10) (1, 3)
Row 10: (0,2) (1, 1) (0,10) (1, 1) (0, 2)
Row 11: (0, 2) (1, 2) (0, 8) (1, 2) (0, 2)
Row 12: (0, 3) (1, 2) (0, 6) (1, 3) (0, 2)
Row 13: (0, 4) (1,8) (0, 4)
Row 14: (0, 7) (1, 2) (0, 7)
Row 15: (0, 16)
Row 16: (0, 16)

encode

decode

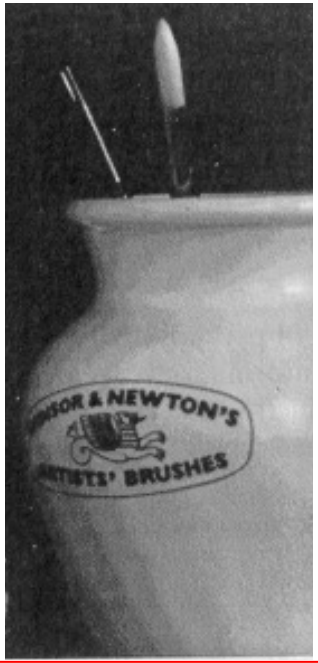


ACK: Prof. Tony Pridmore, UNUK

Psychovisual Redundancy

Some grey level and colour differences are imperceptible; goal is to compress without noticeable change to the image

256 grey levels



16 grey levels

16 grey levels

A simple method:

Add a small random number to each pixel before quantization

ACK: Prof. Tony Pridmore, UNUK

Evaluating Compression

- **Fidelity Criteria**: success is judged by comparing original and compressed version
- Some measures are objective, e.g., root mean square error (e_{rms}) and signal to noise ratio (SNR)
- Let f(x,y) be the input image, f'(x,y) be reconstructed input image from compressed bit stream, then

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

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

ACK: Prof. Tony Pridmore, UNUK

Fidelity Criteria

e_{rms} = **6.93**	e_{rms} = 6.78
SNR_{rm} = 10.25	SNR_{rm} = **10.39**

ACK: Prof. Tony Pridmore, UNUK

Fidelity Criteria

- E_{rms} and SNR are convenient objective measures
- Most decompressed images are viewed by human beings
- Subjective evaluation of compressed image quality by human observers are often more appropriate

Rating scale of Television Allocations Study Organisation.

(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.

ACK: Prof. Tony Pridmore, UNUK

Image Compression Systems

- Transform input data in a way that facilitates reduction of interpixel redundancies
- Reversible

Mapper

- Transform input data in a way that facilitates reduction of psychovisual redundancies
- Not reversible

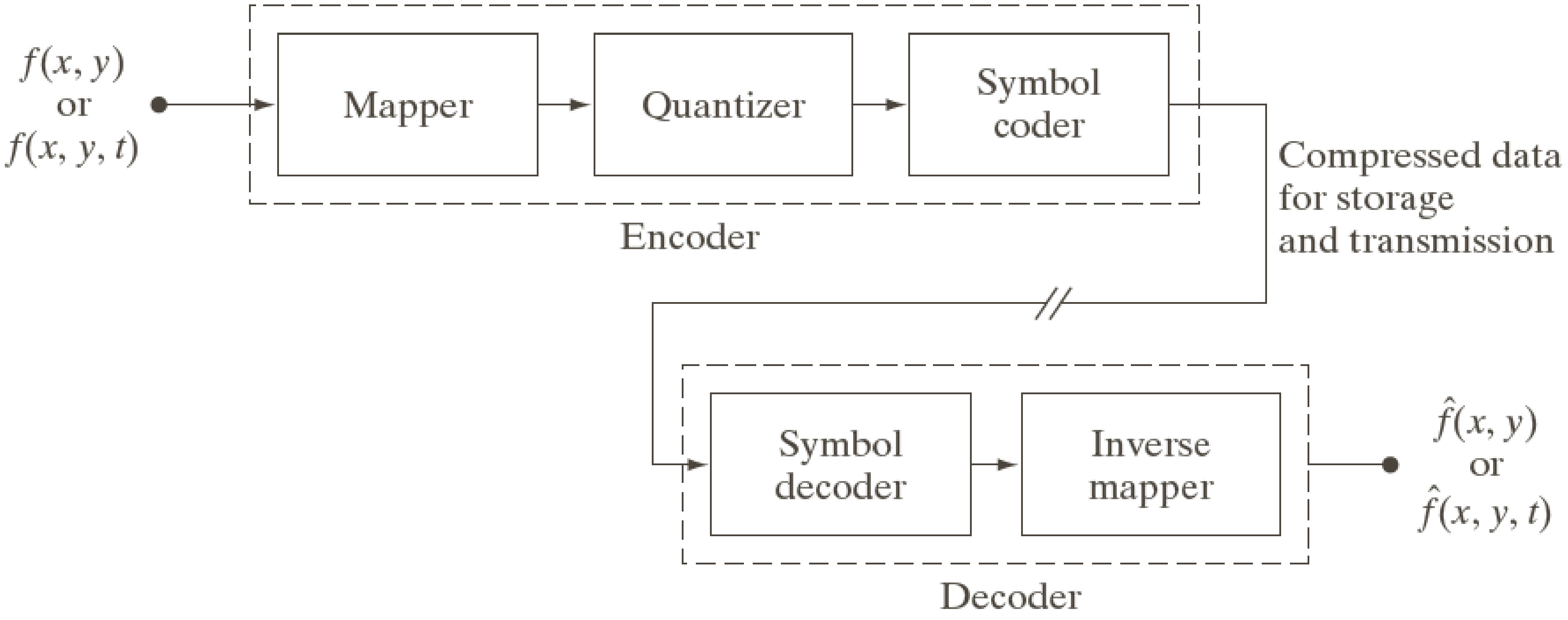
Quantiser

- Assigns the shortest code to the most frequently occurring output values
- Reversible

Symbol coder

ACK: Prof. Tony Pridmore, UNUK

Image Compression Systems



Functional block diagram of a general image compression system

ACK: Prof. Tony Pridmore, UNUK

University of Nottingham
UK | CHINA | MALAYSIA

Huffman Coding

Exploiting Coding Redundancy

Try to maintain a high level of information in compressed images

- These methods are derived from information theory
- Not limited to images, are applicable to any digital information

- Speak of symbols instead of pixel values and sources instead of images
- Exploit nonuniform probabilities of symbols (nonuniform histogram) and use a variable-length code

Evaluation requires a measure of the information content of a source:

Entropy

ACK: Prof. Tony Pridmore, UNUK

Entropy

- **The idea**: associate information with probability
- A random event *E* with probability *P(E)* contains:

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

units of information

Suppose that grey level values are generated by a random variable, then r_k contains:

$$I(r_k) = -log(P(r_k))$$

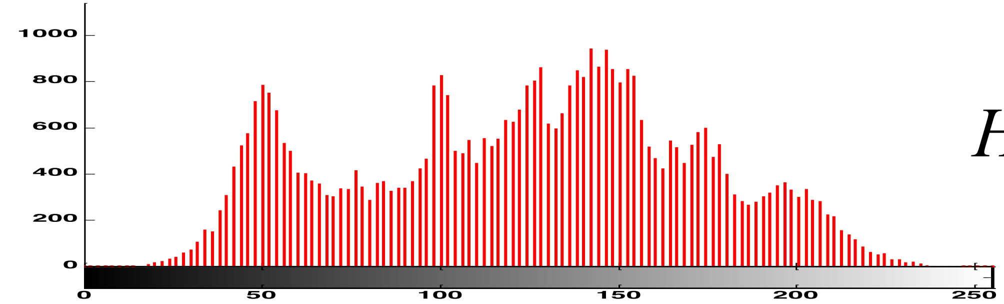
units of information

Note I(E) = 0 when P(E) = 1

ACK: Prof. Tony Pridmore, UNUK

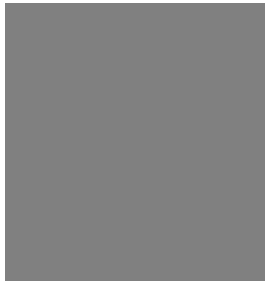
Entropy

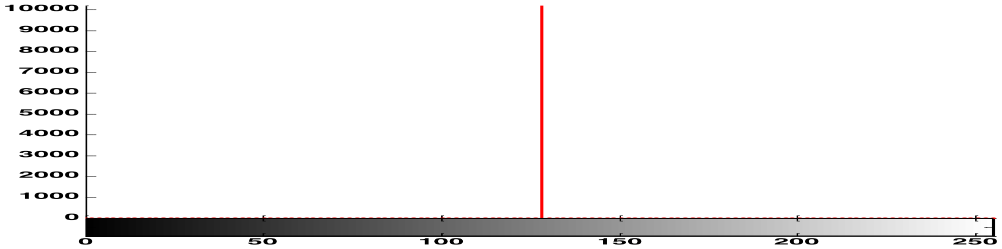
Entropy is the average information content of an image, a measure of histogram dispersion



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

entropy = 7.4635





entropy = 0

Can't compress to less than H bits/pixel without losing information

ACK: Prof. Tony Pridmore, UNUK

Huffman Coding

- Compute probabilities of each symbol by ***histogramming*** the source
- Process probabilities to pre-compute codebook: code(i)

Codebook is static (fixed)

- Encode source symbol-by-symbol:

symbol(i) -> code (i)

- Transmit coded signal and codebook
- The need to pre-process (histogram) the source before encoding begins is a disadvantage

ACK: Prof. Tony Pridmore, UNUK

Huffman Coding

Builds a binary tree in which symbols to be coded are nodes

- Create a list of nodes, one per for symbol, sorted in order of symbol frequency (or probability)
- REPEAT (until only one node left)
  - Pick the two nodes with the lowest frequencies/probabilities and create a parent of them
  - **Randomly** assign the codes 0, 1 to the two new branches of the tree and delete the children from the list
  - Assign the sum of the children's probabilities to their parent and insert it in the list

Algorithm

Path from root to node gives code for corresponding symbol

ACK: Prof. Tony Pridmore, UNUK

Huffman Coding

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

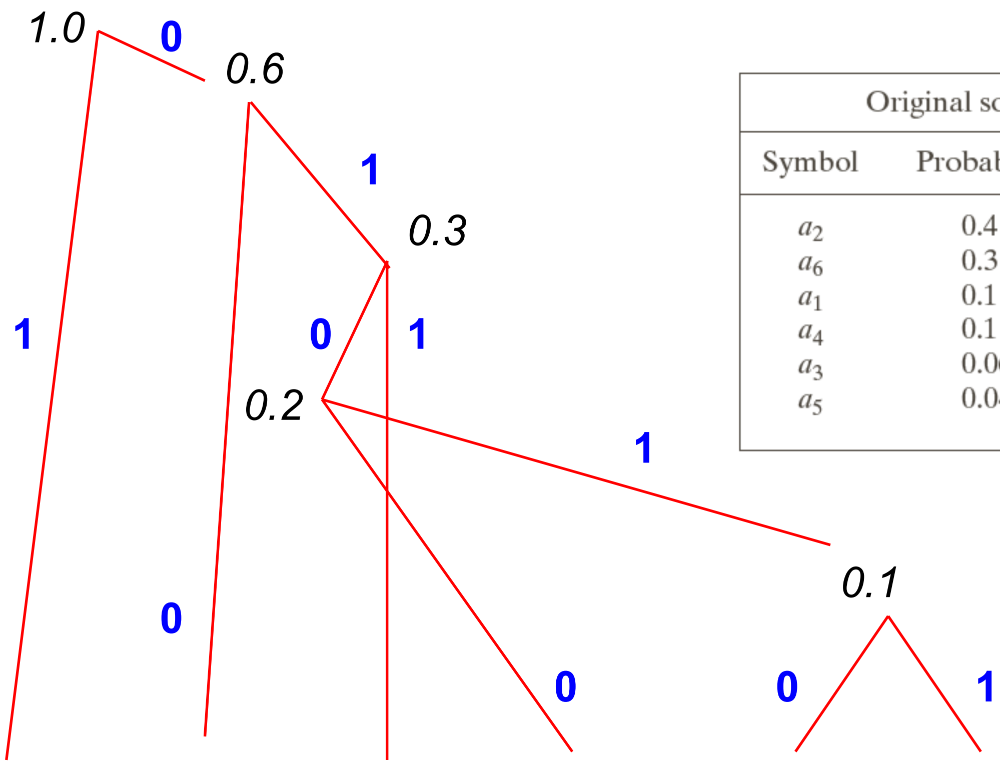
New parent is higher up list (0.2) than a_1 (0.1)

Create parent node of a_3 and a_5

Create parent of new node and a_4

ACK: Prof. Tony Pridmore, UNUK

Huffman Coding



Prob$_i$	0.4	0.3	0.1	0.1	0.06	0.04
Symbol$_i$	a_2	a_6	a_1	a_4	a_3	a_5
Code$_i$	**1**	**00**	**011**	**0100**	**01010**	**01011**

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

ACK: Prof. Tony Pridmore, UNUK

Huffman Coding

- The algorithm systematically places nodes representing high probability symbols further up the tree: *their paths (and so codes) are shorter*
- No code is prefix to any other – don't need to mark boundaries between nodes
  - e.g., 01101010 must be a_1a_3

In this example

- Average length of the code is 2.2 bits/symbol
- The entropy of the source is 2.14 bits/symbol

In general

- Break image into small (e.g., 8 x 8) blocks
- Each block is a symbol to be encoded

ACK: Prof. Tony Pridmore, UNUK

Break

University of Nottingham
UK | CHINA | MALAYSIA

Psychovisual Redundancy GIF

Using Psychovisual Redundancy

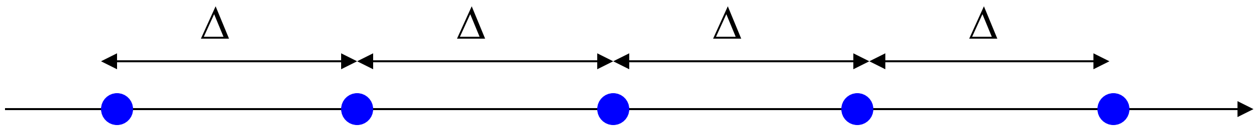
- Represent areas of grey level/colour space with fewer bits
- Lossy: cannot be inverted
- Find the best tradeoff between

Maximal compression ←→ minimal distortion

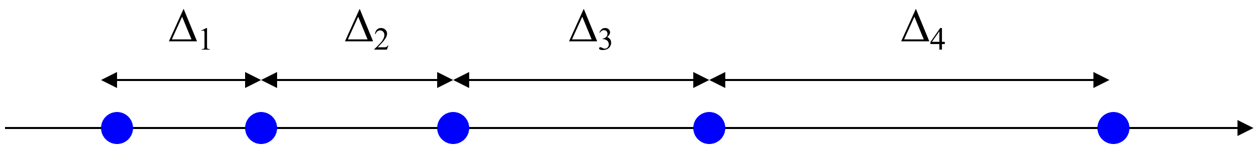
Quantisation is widely used

Scalar Quantisation (i.e., quantising scalar values)

Uniform scalar quantisation:



Non-uniform scalar quantisation:



ACK: Prof. Tony Pridmore, UNUK

Vector Quantisation

Palettised image (gif)

- Map vector values (R,G,B) onto scalar values
- Multiple vectors map to each scalar

Vector quantisation

True colour R,G,B
8 bits each
1677216 possible colours

gif
8 bits per pixel
256 possible colours

ACK: Prof. Tony Pridmore, UNUK

Vector Quantisation

(r, g, b)

For each pixel in the original image

Find the closest colour in the Colour Table

r_0	g_0	b_0
r_1	g_1	b_1
⋮	⋮	⋮
r_{255}	g_{255}	b_{255}

Colour Table

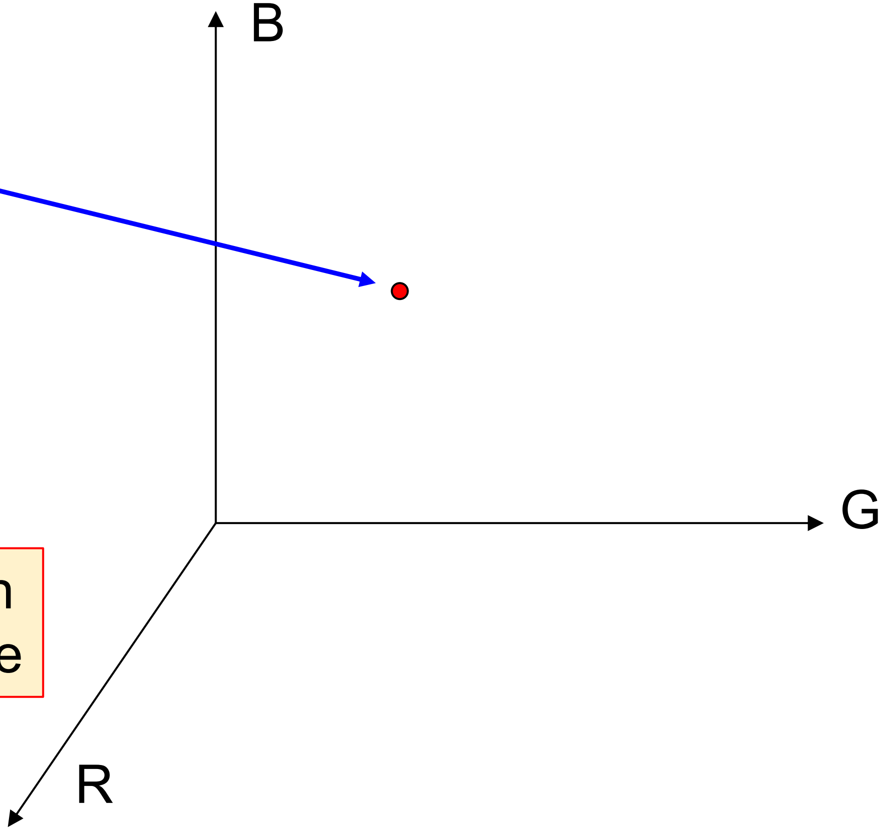
Record the index of that colour (for storage or transmission)

To reconstruct the image, place the indexed colour from the Colour Table at the corresponding spatial location

ACK: Prof. Tony Pridmore, UNUK

Building a Palette

(r, g, b)

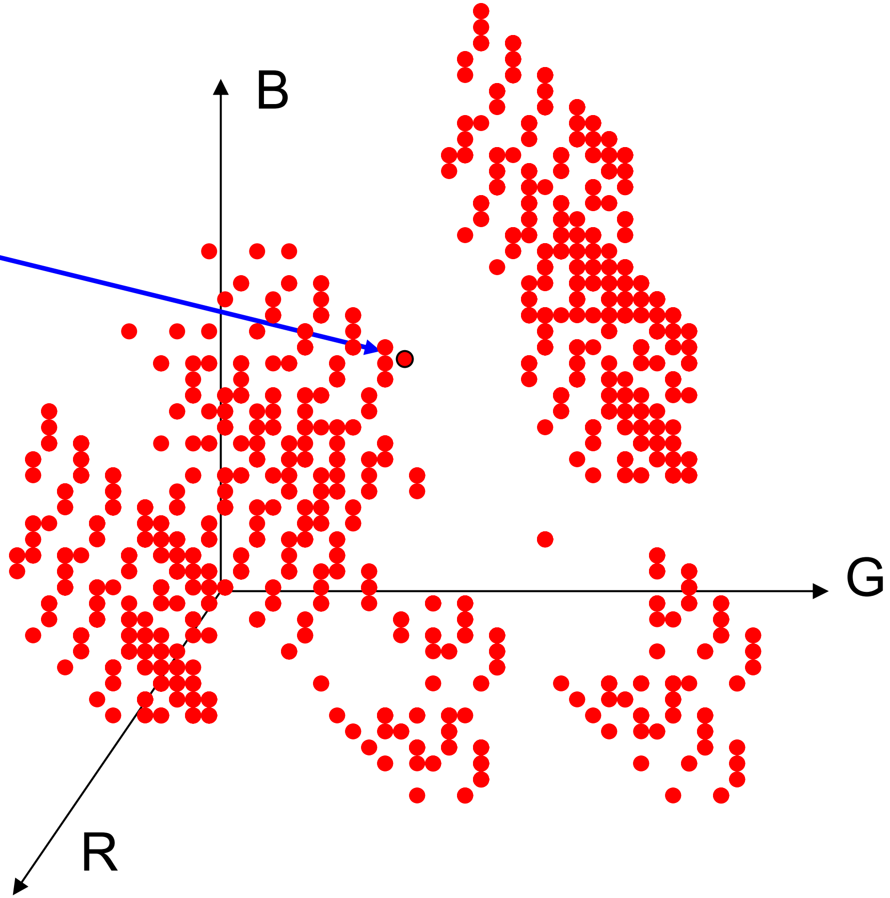


A pixel corresponds to a point in the 3 dimensional R, G, B space

ACK: Prof. Tony Pridmore, UNUK

Building a Palette

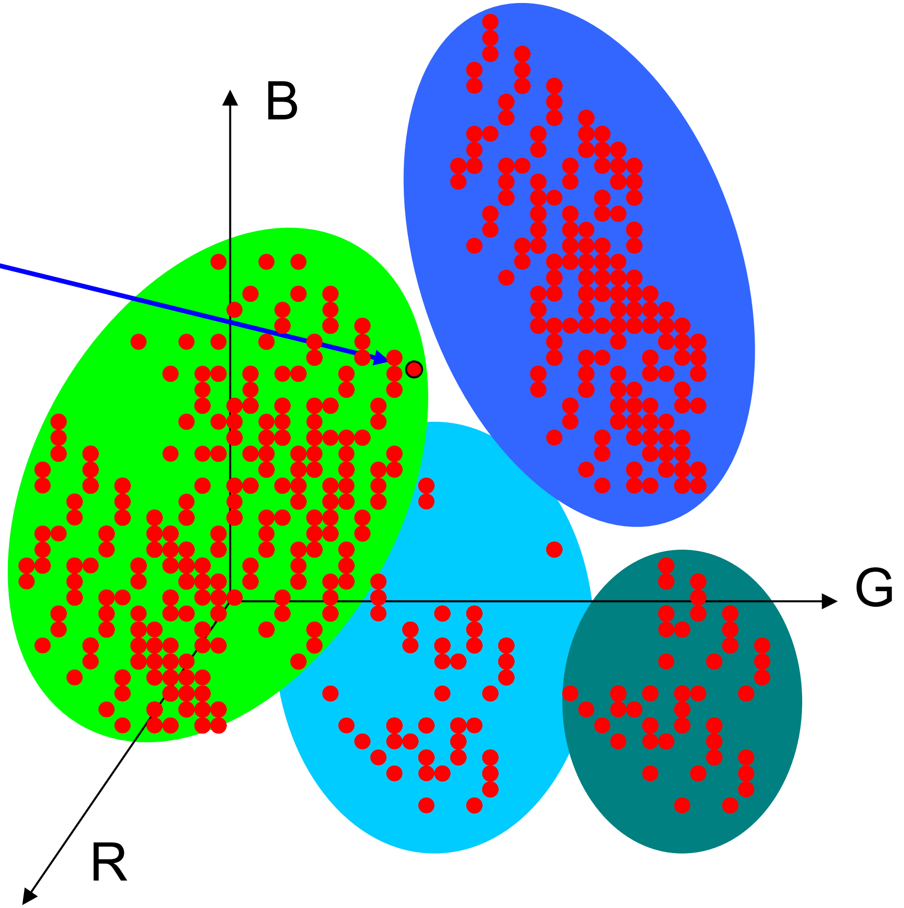
(r, g, b)



Map all pixels into R,G,B space,
Clouds of pixels are formed

ACK: Prof. Tony Pridmore, UNUK

Building a Palette

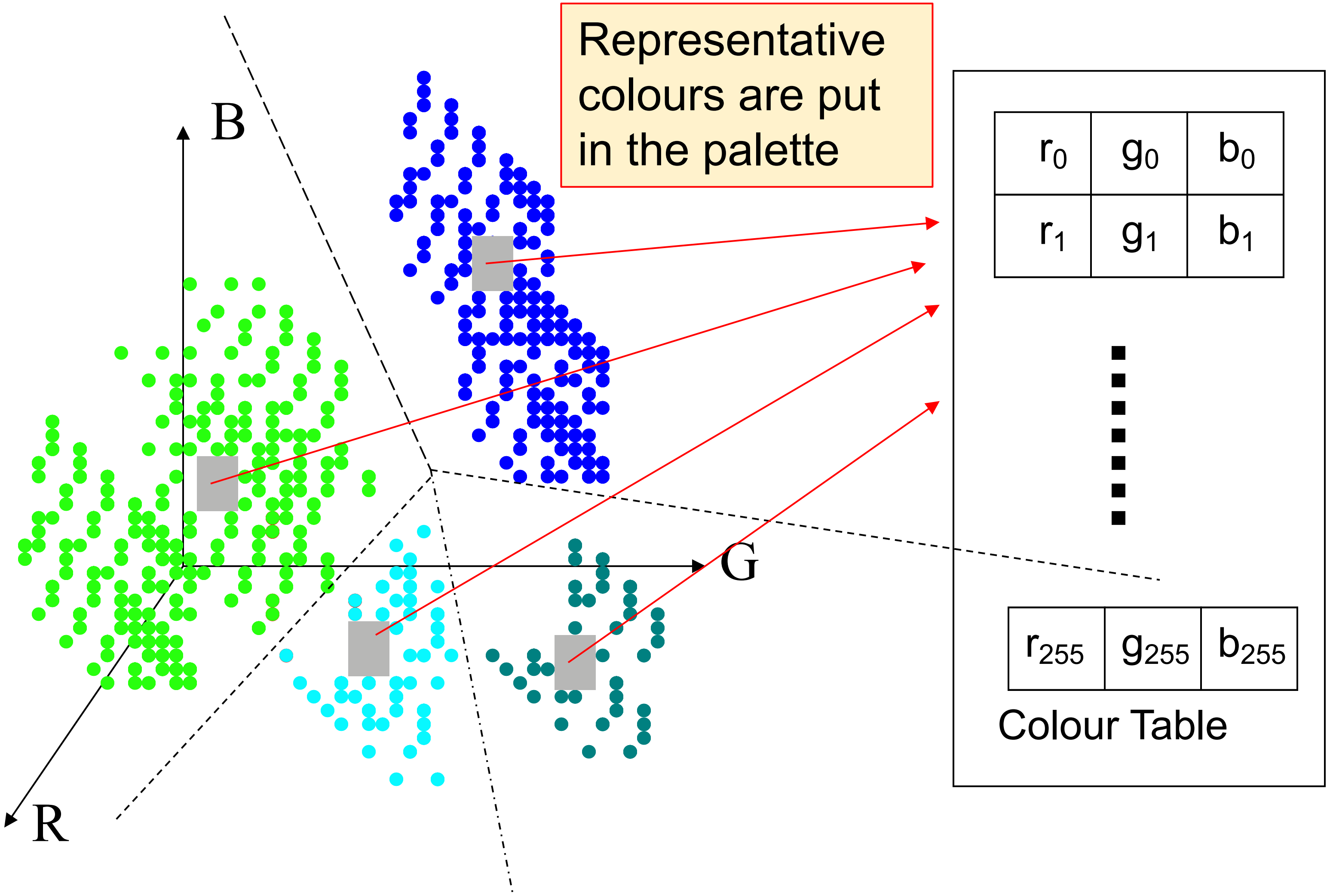


Group pixels that are close to each other, and replace them by one single colour

Close to each other means of a similar colour

ACK: Prof. Tony Pridmore, UNUK

Building a Palette



ACK: Prof. Tony Pridmore, UNUK

Building a Palette

Many clustering algorithms exists

- Supervised
- unsupervised

- We know how many clusters we need: *one per palette entry*
- We need clusters that are spread across the colour space
- A unsupervised method…

K-Means Clustering

- Start with estimates of the mean of each cluster ☞ $\mu_1, \mu_2, \ldots, \mu_k$
- Assign each point, p, to the cluster where ☞ $|p - \mu_i|$ is smallest
- Recompute the means
- Repeat until no changes are made to the clusters

ACK: Prof. Tony Pridmore, UNUK

University of Nottingham
UK | CHINA | MALAYSIA

A Compression System
JPEG

Spatial Redundancy

- Run-length encoding needs adjacent pixels to be **equal**
- Pixels are more often **highly correlated** (dependent)
  - Not equal, but can predict the image pixels to be coded from those already coded

- Each pixel value (except at the boundaries) is predicted based on its neighbours (e.g., linear combination) to get a predicted image
- The difference between the original and predicted images yields a differential or residual image <u>with a reduced set of values</u>
- The differential image is encoded using Huffman coding, or similar

ACK: Prof. Tony Pridmore, UNUK

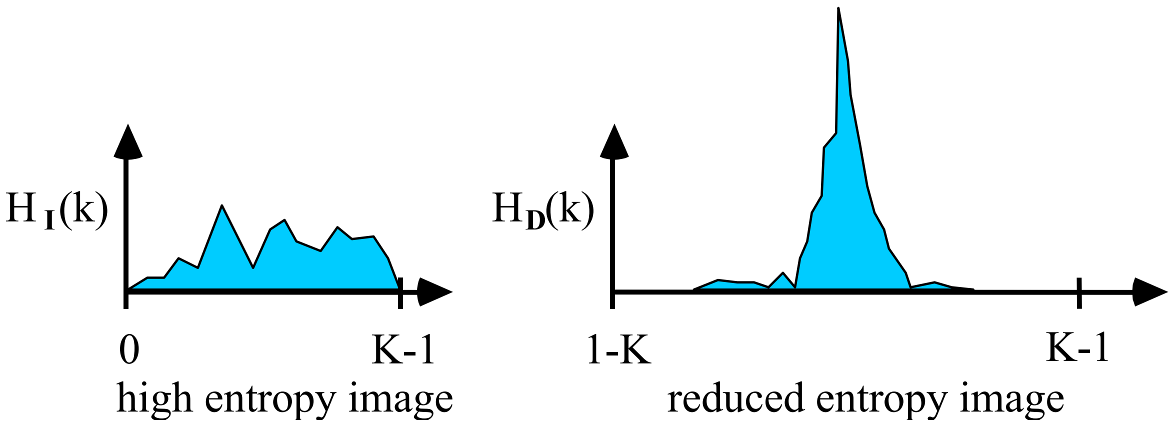
Differential Pulse-code Modulation

Code the difference between adjacent pixels

Original pixels:
82, 83, 86, 88, 56, 55, 56, 60, 58, 55, 50, ……

DPCM:
82, 1, 3, 2, -32, -1, 1, 4, -2, -3, -5, ……

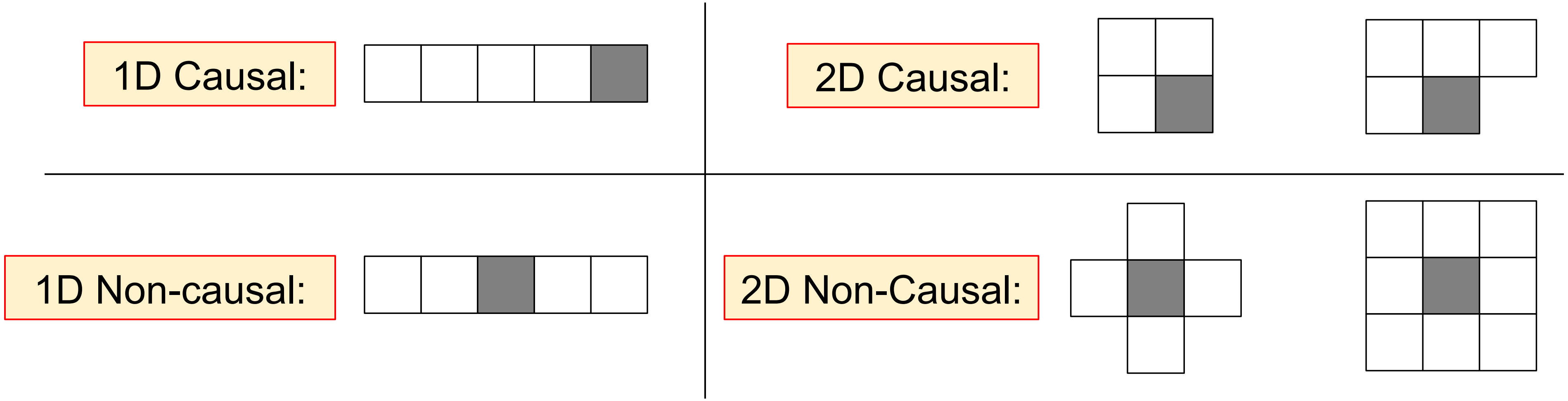
- Prediction is that the next pixel value – current one
- Need the first value to provide a point of reference
- Invertible (lossless) and lower entropy



ACK: Prof. Tony Pridmore, UNUK

Predictive Coding

- Higher order pattern prediction
- Use both 1D and 2D patterns (*to predict shaded pixel*)



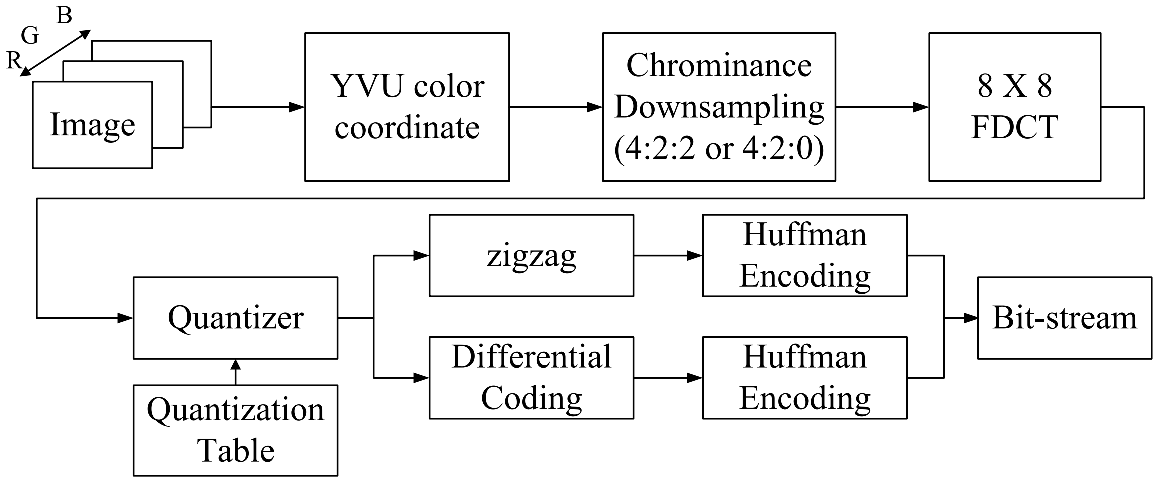
ACK: Prof. Tony Pridmore, UNUK

A Complete System: JPEG

A set of methods with a common *baseline* system

- Discrete Cosine Transform
- Quantisation
- Variable length encoding

A JPEG-compatible product must only include support for the baseline



ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

Conversion RGB to YUV is optional, but common

Y is grey level, U,V are colour (Lecture 2)

Y is kept at full resolution, U and V are downsampled

Y

U

V

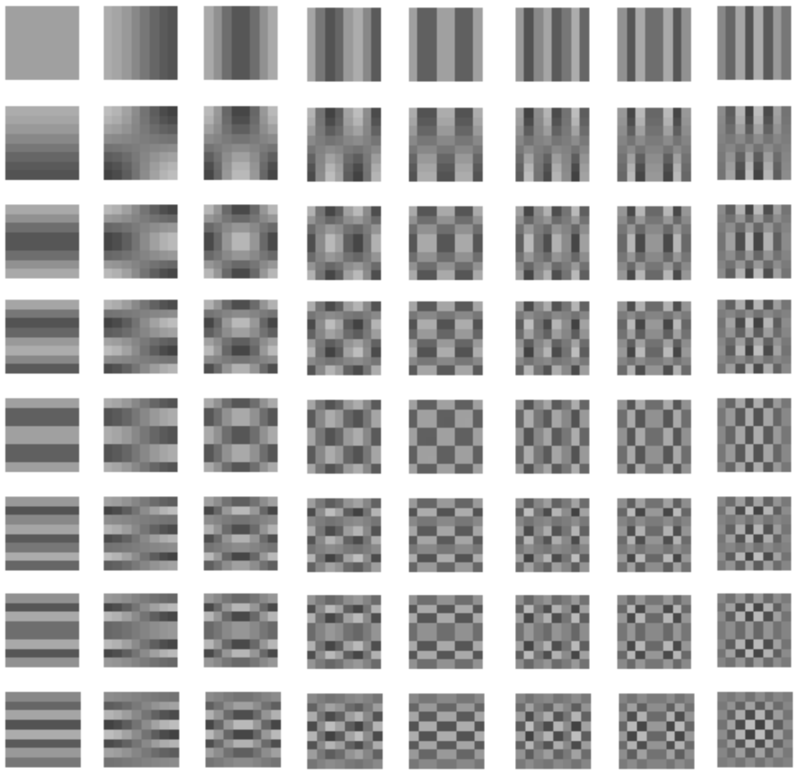
Human vision is more sensitive to grey-level variations than colours

The rest of the process is applied (separately) to each field

ACK: Prof. Tony Pridmore, UNUK

Image Transforms

- Like FFT, basis functions are different
- Top left is DC level
- All other are ”AC”
- Frequency of basis functions increases with distance from origin
- Basis patterns vary in 2D



$$X(u,v) = \frac{4C(u)C(v)}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cos\left(\frac{(2m+1)u\pi}{2N}\right) \cos\left(\frac{(2n+1)v\pi}{2N}\right)$$

ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

- The image is broken in 8 x 8 pixel blocks, which are processed sequentially, top left to bottom right
- First subtract half maximum intensity value so values are distributed about 0

$$
\begin{bmatrix}
52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\
63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\
62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\
63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\
67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\
79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\
85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\
87 & 79 & 69 & 68 & 65 & 76 & 78 & 94
\end{bmatrix}
\quad
\begin{bmatrix}
-76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\
-65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\
-66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\
-65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\
-61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\
-49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\
-43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\
-41 & -49 & -59 & -60 & -63 & -52 & -50 & -34
\end{bmatrix}
$$

Image block

Normalised image block

ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Normalised image block

2D-DCT

high frequency

DC component

high frequency

$$\begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.13 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.88 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}$$

DCT block

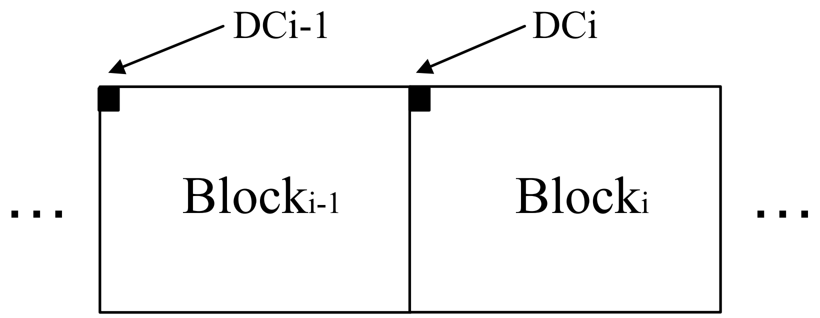
ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

- AC and DC components are processed separately
- DC components summarise patch intensity, so should vary smoothly over patches,

Use different coding (DCPM)

$$\text{Diff}_i = \text{DC}_i - \text{DC}_{i-1}$$



ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

AC components are quantised

- Divide the DCT block entries by values in a quantisation table
- Different tables for luminance (Y) and chrominance (UV) blocks

$$Q_Y = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix} \quad Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

$$X'(u,v) = Round\left(\frac{X(u,v)}{Q(u,v)}\right)$$

X(u,v): original DCT coefficient
X'(u,v): DCT coefficient after quantization
Q(u,v): quantization value

ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

Why quantise ?

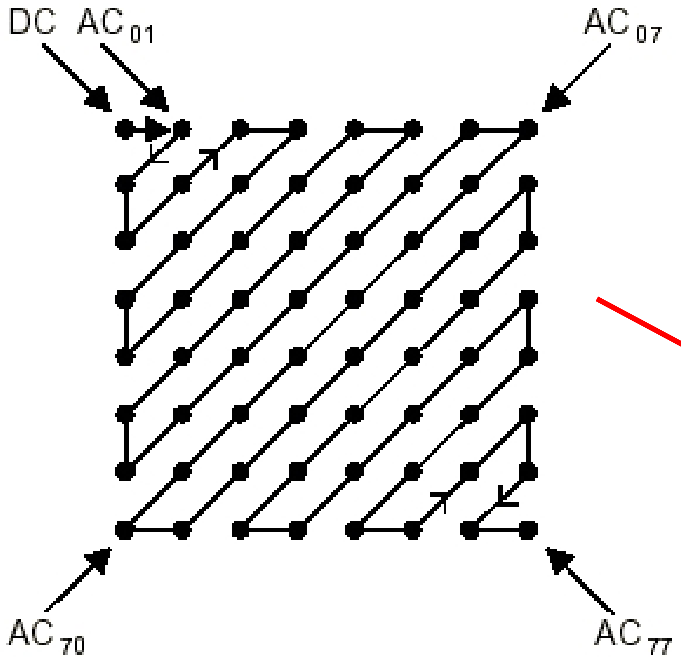
- Further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality
- Generally, the “high frequency coefficients” have larger quantisation values
- Quantisation makes most coefficients zero, it makes JPEG compression efficient, but ”lossy”

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

Zigzag coding orders elements of quantised DCT block (roughly) on frequency



Ordering produces long sequences of zeros

Sequences of AC values can be represented using:

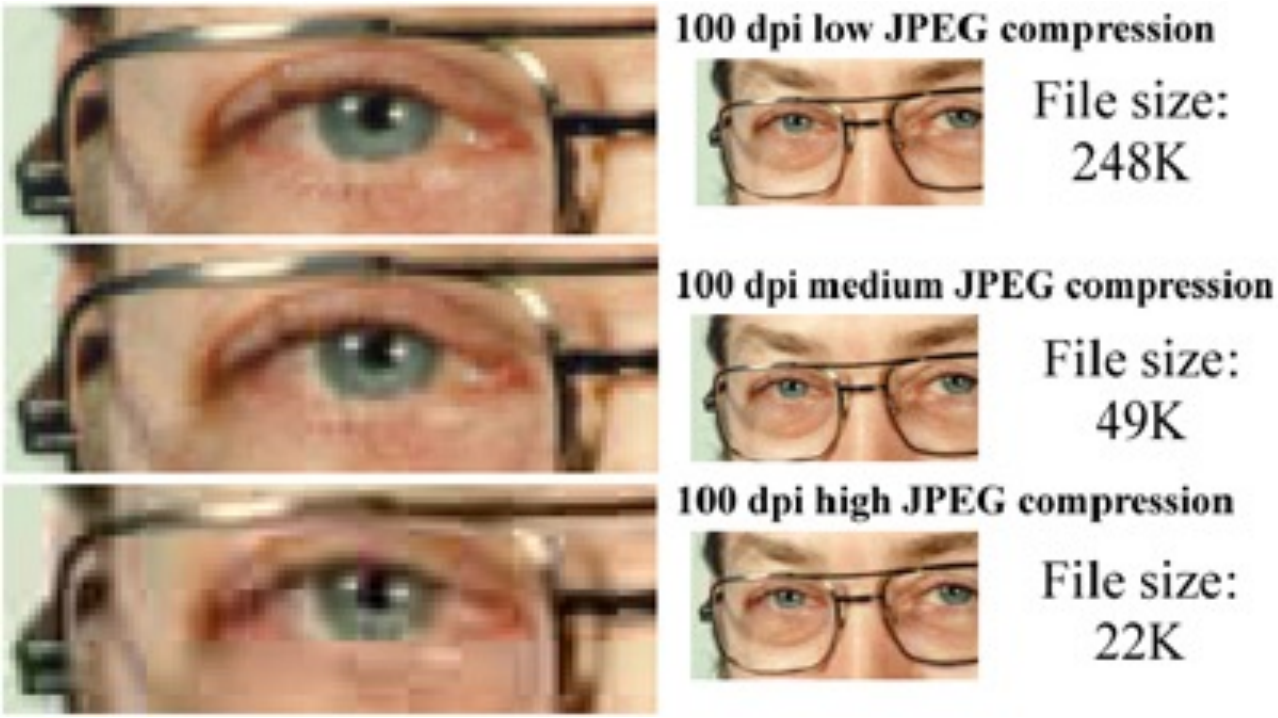
- Run-length coding
- Huffman coding

−26 −3 0 −3 −2 −6 2 −4 1 −3….

ACK: Prof. Tony Pridmore, UNUK

JPEG Compression

Increasing the amount of quantisation reduces file size but introduces artefacts: **blocks become visible**



ACK: Prof. Tony Pridmore, UNUK

Summary

1. Redundancy
2. Huffman Coding
3. Psychovisual Redundancy - GIF
4. A Compression System - JPEG

University of Nottingham
UK | CHINA | MALAYSIA

Questions

University of Nottingham
UK | CHINA | MALAYSIA

NEXT:

Interactive Methods & Compositing