University of Reading

# SE3IA11/SEMIP12 Image Analysis

# Image Compression - The Fundamentals

Lecturer:
Prof. James Ferryman  Computational Vision Group
Email: j.m.ferryman@reading.ac.uk

University of Reading

# Image Compression

- **The problem**:

  Consider the storage and transmission of a 60min full-motion colour movie (images are 8-bit and of size 512x512):

  Storage: 512x512x3x25x3600 ~ 71GB!

  or 120 CDROMs or 16 DVDs …

  Transmission: downloading at 10M/s ~ about 2 hours!

  On top of the above is the cost associated with audio data!

- **The solution**:

  Compress the images by throwing out redundant and insignificant data

University of Reading

# Another Example

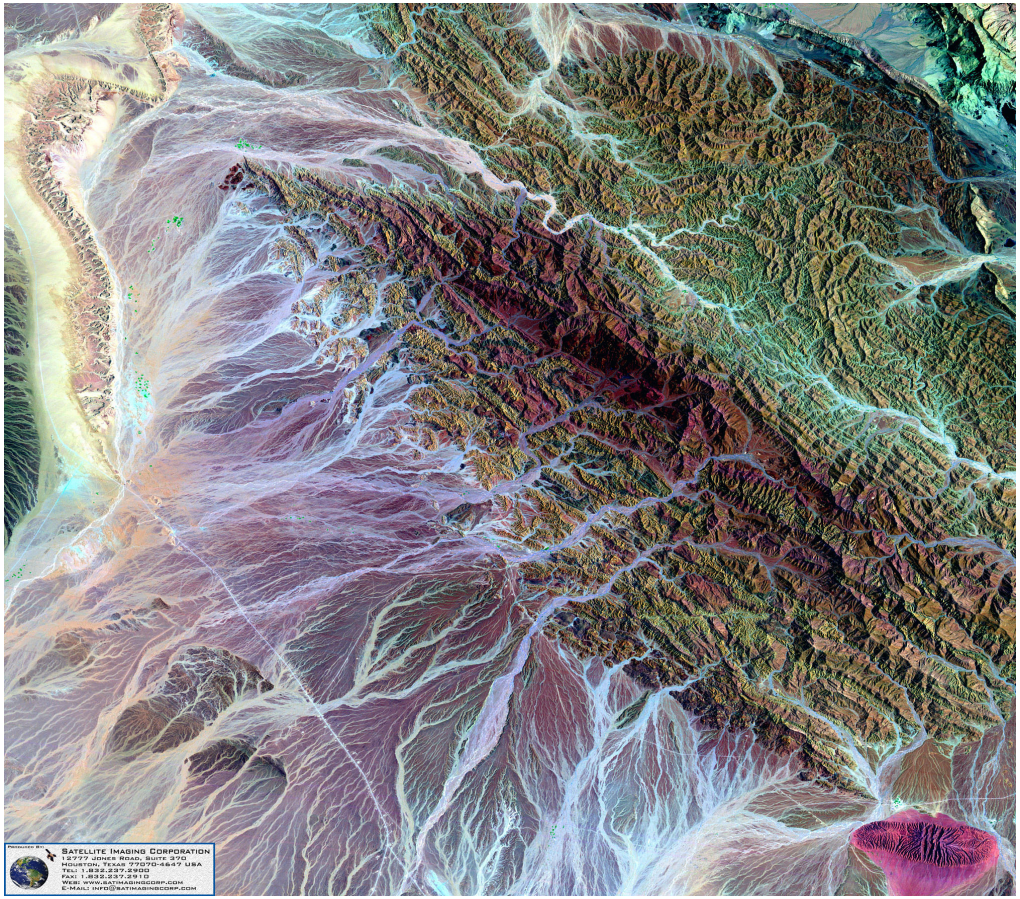
- A Landsat D satellite broadcasts 85 x $10^6$ bits / s
- A typical image from one pass contains 6100 x 6100 pixels in 7 spectral bands ~ 260 Mb image data

- Japanese Advanced Earth Observing Satellite (ADEOS) has spatial resolution of 8 metres for the polychromatic band and 16 metres for the multispectral bands has a transmitted data rate of 120 Mbps

University of Reading

# Satellite Imagery

- How would you compress the following?



University of Reading

# Image Compression

- Compressing an image is significantly different from compressing raw binary data
  - Naturally, you may use a general purpose compression program but the result is less than optimal
- Images contain certain statistical properties which can be exploited by encoders specifically designed for them
- Additionally, some of the finer image detail can be sacrificed for the sake of saving more bandwidth/ storage space

University of Reading

# Data Redundancy

- Three types of redundancies associated with the digital coding of intensity values, intensity correlations among neighbouring pixels, and the psychovisually insignificant image data

  Coding redundancy

  In a digital 8-bit image, all 256 intensity values are represented using 8 bits, i.e. the digital codes for all values are of the same length (8), So for value 0, you have 00000000; value 1 you have 00000001; etc.

  The above is based on an important assumption that all intensity values are equally probable in the image

  This is a VERY unrealistic assumption (look at the intensity histogram of any natural image you come across to see whether it' s really flat).

University of Reading

# Coding Redundancy

- A better code design method is to let the length of the codes be linked to the probability of the intensity values
- For example, shorter codes for more probable values and longer codes for less frequent ones
- Consider the following:

| | | |
|---|---|---|
| 0 | 2 | 2 |
| 1 | 2 | 3 |

Intensity values

| | | |
|---|---|---|
| 01 | 10 | 10 |
| 01 | 10 | 11 |

Fixed-length codes
(no. of bits: 12)

| | | |
|---|---|---|
| 10 | 0 | 0 |
| 10 | 0 | 110 |

Variable-length codes
(no. of bits: 10)

- Note this does not involve any change of image content. It merely changes the representation (the codes) of the intensity values in the computer

# Spatial Redundancy

University of Reading

- Images often include large areas of similar intensity values
- Intensities of surfaces of real-world objects usually change fairly slowly
- The intensity of a pixel can be predicted rather accurately based on intensity values of neighbouring pixels – spatial redundancy

University of Reading

# Psychovisual Redundancy

- The eye is a strange optical device. It does not respond to all visual information with equal sensitivity
- Some visual information is simply unnoticeable to the naked human eyes
- If the ultimate information receiver is the human eyes (such as watching a movie), then there is no point to process such *psychovisually redundant* information
- Psychovisual redundancy is fundamentally different from coding and spatial redundancy as its removal always incurs the loss of information (though psychovisually insignificant)

University of Reading

# Psychovisual Redundancy

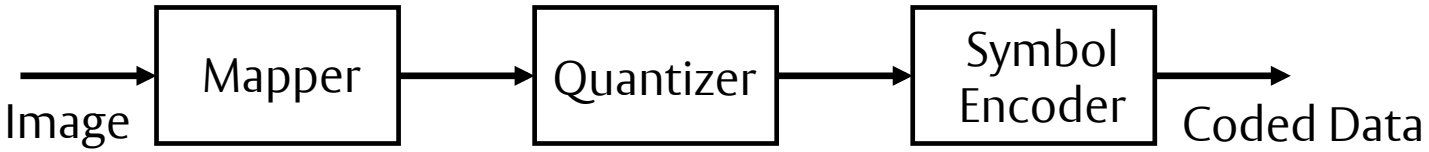

Image represented in 256 greylevels


Image quantized to 16 greylevels

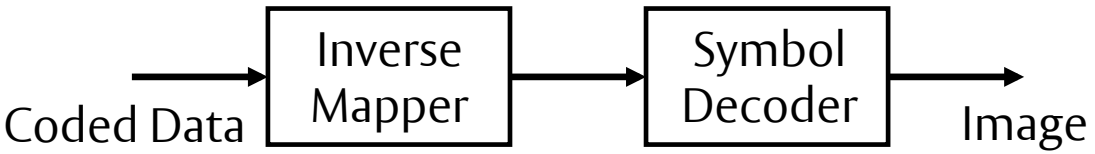
University of Reading

# General Encoder/Decoder Models

- Compression techniques attempt to remove one or more of the above redundancies according to the following general model:




**Encoder Model**




**Decoder Model**

University of Reading

# General Encoder/Decoder Models

- The *mapper* maps the image from the spatial domain representation to a representation which makes spatial redundancy more accessible. The mapping (e.g. DCT) is usually invertible so no information is lost
- The *quantizer* represents mapper outputs with a reduced accuracy, especially those outputs which correspond to psychovisual redundancy. This is an often irreversible operation so it must not be included if error-free compression is required
- The *symbol encoder* assigns codes to the quantized output values. It removes coding redundancy by allocating short codes to more frequent quantized values and long ones to less frequent values
- The *decoder model* is the opposite of the encoder model but has no de-quantizer (since quantization is irreversible)

University of Reading

# Lossless vs. Lossy

- *Lossless compression* – incurs no information loss
  - Involves compressing data, which when decompressed, will be an exact replica of the original data
  - Appropriate for data that has been expensive to capture (e.g. triband satellite images)
  - Need to be exactly reproduced when decompressed
- *Lossy compression* – some information is lost and cannot be recovered
  - *Approximation of the original image is often sufficient for most purposes*
  - *Error between original and compressed image must be tolerable*

University of Reading

# Compression

- The usual steps involved in compressing an image are:
  1. Specify the *rate* (bits available) and *distortion* (tolerable error) parameters for the target image
  2. Dividing the image data into various classes, based on their importance
  3. Dividing the available bit budget among the classes, such that the distortion is a minimum
  4. Quantize each class separately using the bit allocation information derived in Step 3
  5. Encode each class separately using an entropy coder and write to file

University of Reading

# Decompression

- Reconstructing an image from compressed data is usually a faster process than compression
- The steps involved are:
  1. Read in the quantized data from the file, using an entropy decoder (reverse of Step 5 on previous slide)
  2. Dequantize the data (reverse of Step 4 on previous slide)
  3. Rebuild the image (reverse of Step 2 on previous slide)

# Codec Performance Measures

University of Reading

- Three important factors to consider when choosing a compression technique

1. Fidelity measures the quality of the compressed-then-decompressed image. Objective measures include the *root-mean-square error* $E_{rms}$ between original image $f(x,y)$ and decompressed image $\hat{f}(x,y)$

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

and the *mean-square-to-noise ratio* $SNR_{ms}$

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

University of Reading

# Codec Performance Measures

- Mean Square Error (MSE) represents cumulative squared error between compressed and original image
  - Lower value represents less error
- Signal Noise Ratio (SNR)
  - High value represents less error

University of Reading

# Codec Performance Measures

- Subjective measures include absolute ratings (excellent, fine, passable, marginal, inferior, unusable), and relative ratings (much worse, worse, slightly worse, the same, slightly better, better, much better)

2. Compression ratio

   CR = number of bits of original image divided by that of compressed image

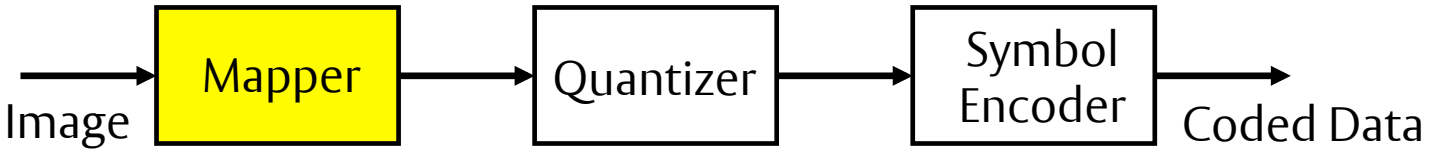
3. Computational cost

   Computation required to compress and decompress an image

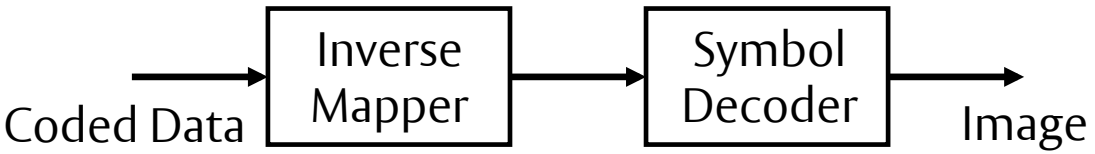
University of Reading

# Encoder/Decoder Model

- Compression techniques attempt to remove one or more of the above redundancies according to the following general model:




**Encoder Model**

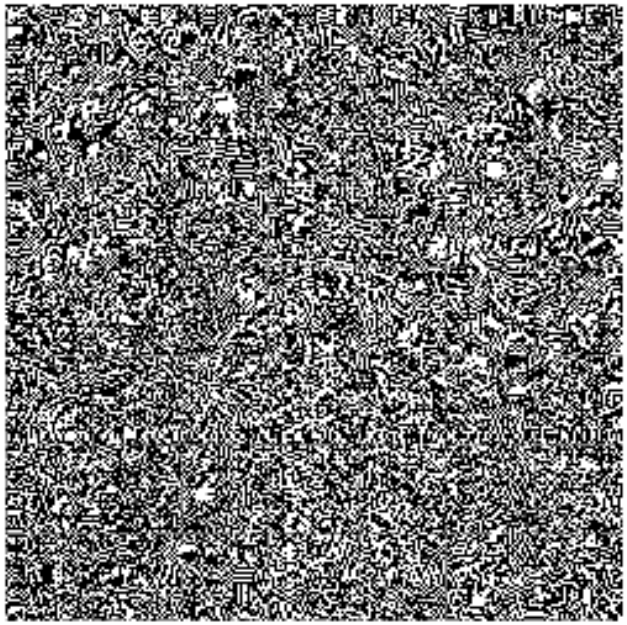



**Decoder Model**

University of Reading

# Mapper

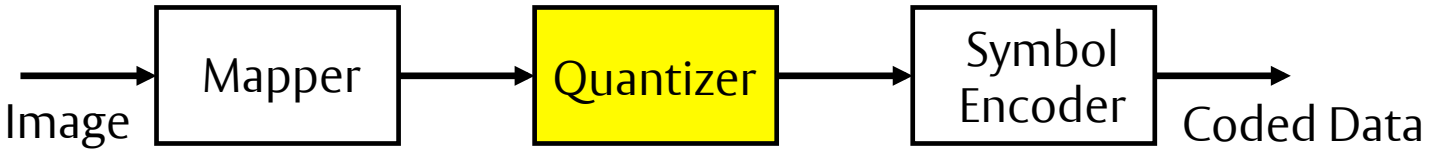
Each 8x8 block processed with DCT to obtain DCT coefficients



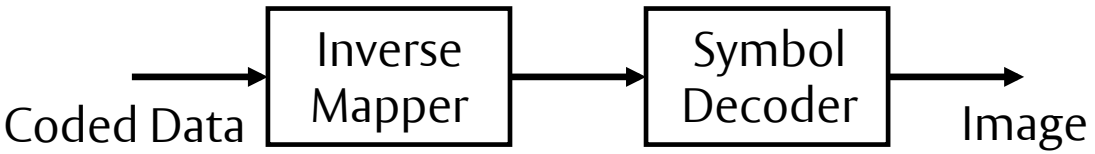
University of Reading

# Encoder/Decoder Model

- Compression techniques attempt to remove one or more of the above redundancies according to the following general model:




**Encoder Model**




**Decoder Model**

University of Reading

# Quantization

- Refers to process of approximating continuous set of image data values with finite (preferably small) set of values
- Input to quantizer is original data, output is always one among a finite number of levels
- Quantizer is a function whose set of output values are discrete, and usually finite
- Essentially a process of approximation
- A good quantizer is one which represents original signal with minimum loss of information

University of Reading

# Quantization

- There are two types of quantization
  - Scalar Quantization
  - Vector Quantization

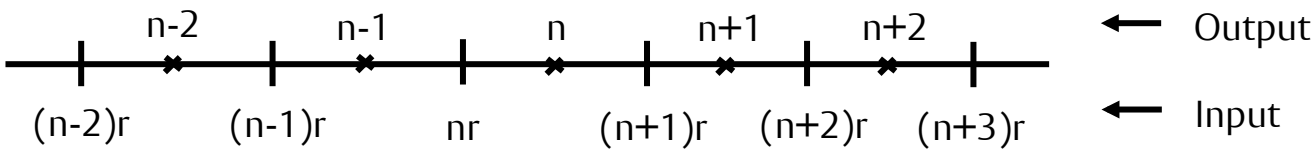
- In Scalar Quantization, each input symbol is treated separately in producing the output

- In Vector Quantization, the input symbols are grouped together into “vectors”, and processed to give output

# Scalar Quantization

University of Reading

- A quantizer can be specified by its input partitions and output levels (also called *reproduction points)*
- If the input range is divided into levels of equal spacing, then quantizer is termed *Uniform Quantizer*, otherwise a *Non-Uniform Quantizer*
- A Uniform Quantizer can be easily specified by its lower bound and the step size, and is also easier to implement




A Uniform Quantizer

- If the input falls between n*r and (n+1)*r, the quantizer outputs the symbol n

University of Reading

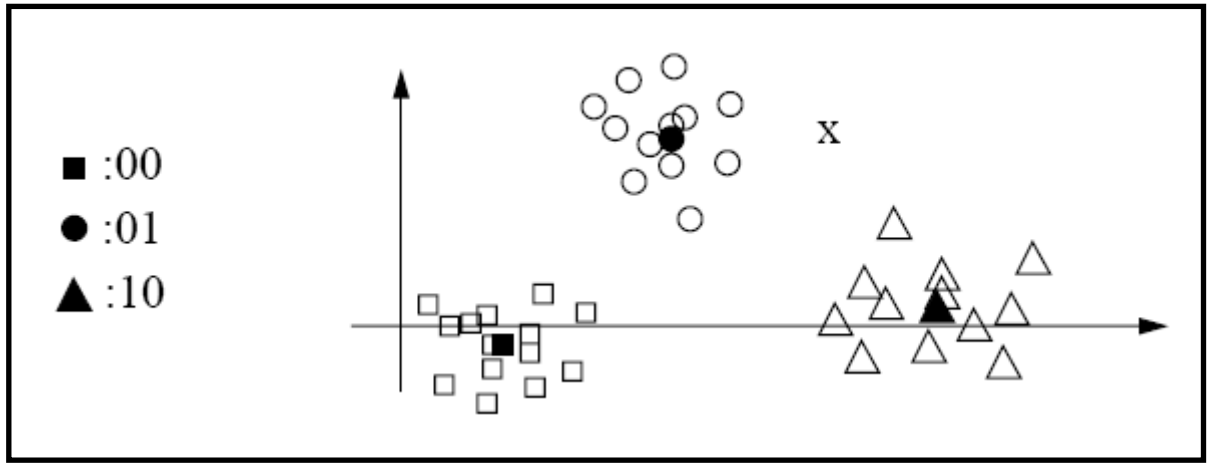
# Vector Quantization

- Divide image into small blocks (e.g. nxn) and regard the $n^2$ intensity values as forming a $n^2$-dimension vector (so each block is a point in the $n^2$-dimension space)
- Describe each cluster of such vectors/points by a representative prototype vector
  - e.g. a vector located at the centre of the cluster

  and represent the prototype vector using a pre-defined codeword
- Use of prototype increases optimality of vector quantizer, but at expense of increased computational complexity

University of Reading

# Vector Quantization




- Compression is achieved by representing each vector using the codeword of the nearest prototype vector
- Decompression is performed by look-up of a codeword table

# Dequantization

University of Reading

- Receives output levels of a quantizer and converts them into normal data
  - Translates each level into a *reproduction point* in the actual range of data

- The optimum quantizer (encoder) and optimum dequantizer (decoder) must satisfy the following conditions:
  - Given the output levels or partitions of the encoder, the best decoder is one that puts the reproduction points x' on the centres of mass of the partitions. Known as the *centroid condition*
  - Given the reproduction points of the decoder, the best encoder is one that puts the partition boundaries exactly in the middle of the reproduction points, i.e. each x is translated to nearest reproduction point. Known as *nearest neighbour condition*

- The quantization error (x – x' ) is used a measure of the optimality of the quantizer and dequantizer

University of Reading

# Bit Allocation

- First step in compressing image is segregation of data into different classes
- Depending on importance of data it contains, each class is allocated a portion of total bit budget, such that compressed image has minimum possible distortion
- This procedure is called *Bit Allocation*
- The Rate-Distortion theory (RDT) – Shannon - is often used for solving problem of allocating bits to set of classes
- Theory aims at reducing distortion for given target bitrate, by optimally allocating bits to various classes of data

University of Reading

# Optimal Bit Allocation using RDT

1. Initially, all classes are allocated a maximum number of bits
2. For each class, one bit is reduced from its quota of allocated bits, and the distortion due to the reduction of that 1 bit is calculated
3. Of all classes, the class with minimum distortion for a reduction of 1 bit is noted, and 1 bit is reduced from its quota of bits
4. The total distortion for all classes D is calculated
5. The total rate for all classes is calculated as $R = p(i) * B(i)$, where p is the probability and B is the bit allocation for each class
6. Compare the target rate and distortion specifications with the values obtained above. If not optimal, go to Step 2.

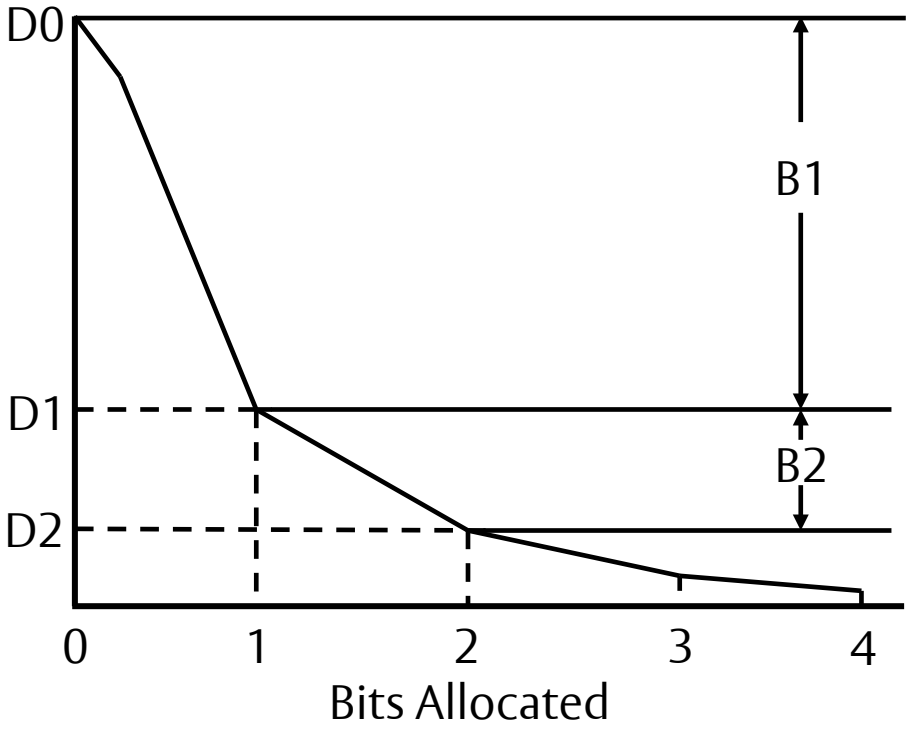
University of Reading

# Optimal Bit Allocation using RDT

- In the approach, one bit at a time is reduced until optimality is achieved either in distortion or target rate, or both
- An alternate approach is to initially start with zero bits allocated for all classes, and to find the class which is most *benefitted* by obtaining an additional bit
- The *benefit* of a class is defined as the decrease in distortion for that class

University of Reading

# Optimal Bit Allocation using RDT

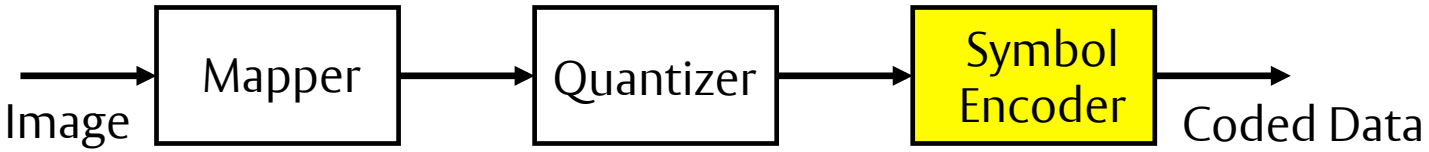



- A benefit of a bit is a decreasing function of the number of bits allocated previously to the same class

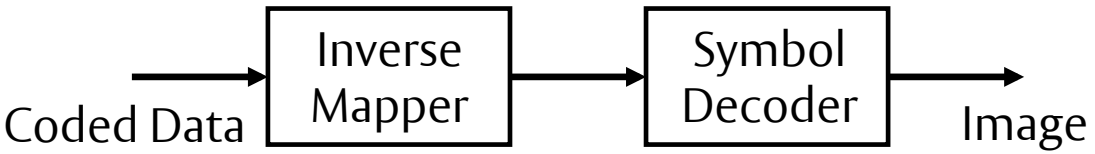
University of Reading

# Encoder/Decoder Model

- Compression techniques attempt to remove one or more of the above redundancies according to the following general model:




**Encoder Model**




**Decoder Model**

University of Reading

# Symbol (Entropy) Coding

- After the data has been quantized into a finite set of values, it can be encoded using an Entropy Coder to provide additional compression
- Entropy refers to the amount of information present in the data
- If an image has G greylevels, and the probability of greylevel k is P(k), then Entropy H_e, not considering correlation of greylevels, is defined as

$$H_e = -\sum_{k=0}^{G-1} P(k)\log_2(P(k))$$

- A good estimate of entropy is usually not available
- Image entropy can however be estimated from a greylevel histogram

University of Reading

# Entropy

- Let h(k) be the frequency of greylevel k in an image f, with 0 <= k <= 2^b-1, and let the image size be M x N
- The probability of occurrence of greyelevel k can be estimated as

$$\tilde{\tilde{P}}(k) = \frac{h(k)}{MN}$$

and the entropy can be estimated as

$$H_e = -\sum_{k=0}^{2^b-1} \tilde{\tilde{P}}(k) \log_2 (\tilde{\tilde{P}}(k))$$

University of Reading

# Entropy Coding

- An entropy encoder encodes the given set of symbols with the minimum number of bits required to represent them
- Two of the most popular entropy coding schemes are:
  - Huffman Coding
  - Arithmetic Coding

University of Reading

# Huffman Coding

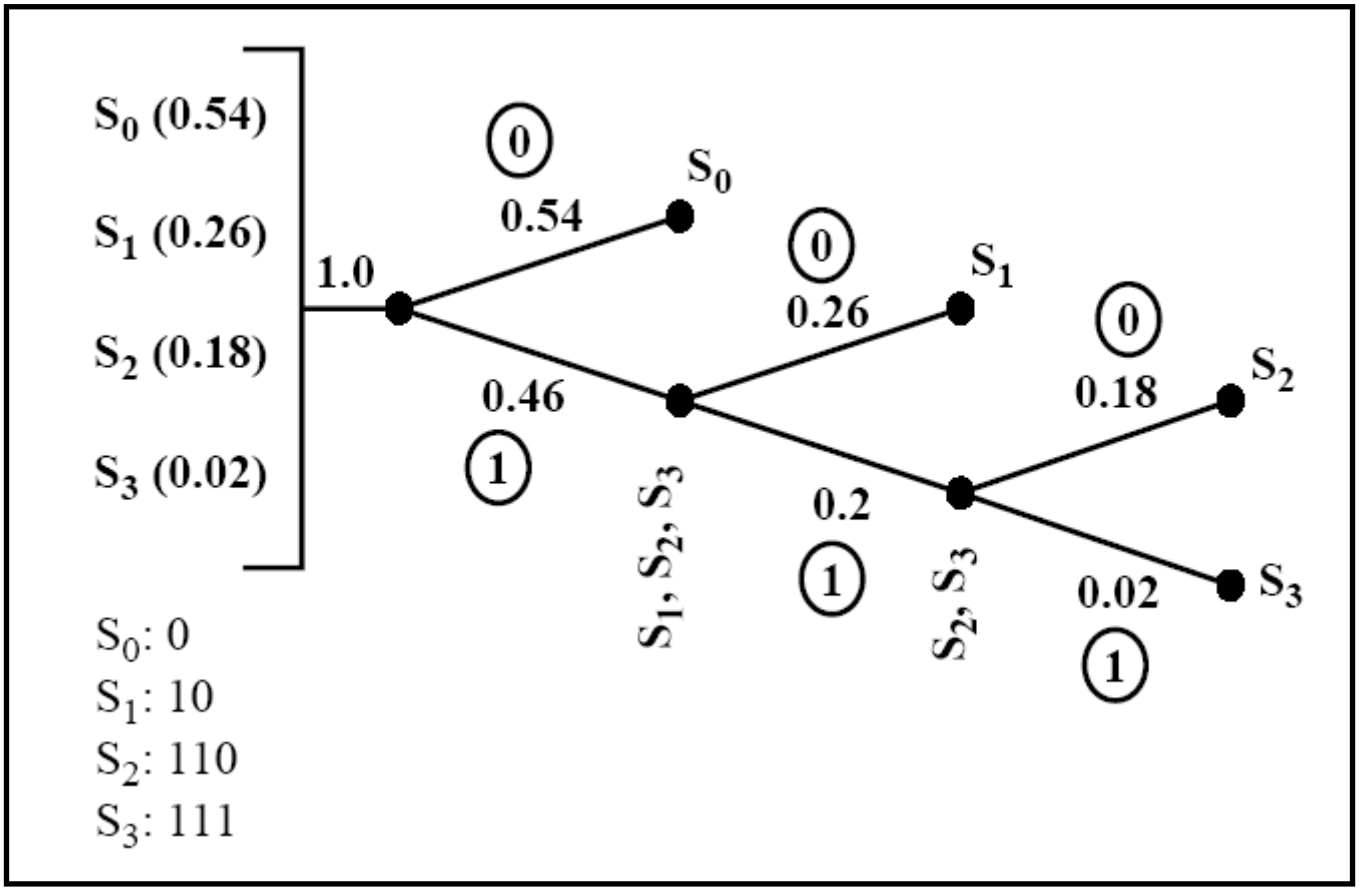
- A widely used technique for lossless compression (also called entropy coding); simply by removing coding redundancy (i.e. shorter codes for more frequent symbols, long ones for less frequent symbols)

  Generating Huffman Codes

  - Recursively divide symbols into two groups, one containing only the most frequent symbol and the other the remaining
  - At each level of the tree, assign 0 to the more frequent group and 1 to the other
  - Huffman codes are obtained by traversing the tree from root to symbols by recording 0s and 1s encountered

University of Reading

# Huffman Coding



University of Reading

# Huffman Coding

- Huffman codes are uniquely decodable by means of lookup tables.
- A string of Huffman codes can only have one valid interpretation
- In the example on the previous slide, the string 110100111111 can only be decoded as the symbol sequence $S_2S_1S_0S_3S_3$.

University of Reading

# Truncated Huffman Coding

- If the number of symbols is large, generating the Huffman codes can be a time-consuming process and codes for some symbols can be very long
- A solution to this is to represent only the most frequent symbols using Huffman codes and the others using the usual fixed-length codes prefixed by a suitable Huffman code – Truncated Huffman Coding

University of Reading

# Problems with Huffman Coding

- Huffman codes have to be an integral number of bits long
  - For example, if the probability of symbol is 1/3, the optimum number of bits to code that symbol is around 1.6
  - The Huffman coding scheme has to assign either 1 or 2 bits
  - Either choice leads to a longer compressed image than is theoretically possible
- In non-adaptive data compression, a single pass over data is required to collect statistics; data is then encoded using statistics; decoder needs a copy of the statistics, which generates an overhead
- In adaptive data compression, both encoder and decoder start with statistical model in same state; models updated after each symbol processed
- Problem with combining adaptive modelling with Huffman coding is that rebuilding Huffman tree is a very expensive process

University of Reading

# Arithmetic Coding

- Respectable candidate to replace Huffman coding
- Completely bypasses idea of replacing input symbol with specific code
- Takes a stream of input symbols and replaces it with a single floating point output number
- The longer (and more complex) the data, the more bits are needed in the output number
- Coding best achieved with 16 bit or 32 bit integer math
- No floating point math is required; incremental transmission scheme is used

University of Reading

# Further Representation/ Coding Methods

University of Reading

# Run Length Coding

- Widely used technique for lossless data compression
- In greylevel images, a run is a sequence of consecutive pixels (say in a row) having constant intensity
- In binary images, a run is a string of consecutive 1s or 0s
- Each run is described by a pair of numbers showing the length and intensity of the run
- Example: 333222333322233300111

  RL codes: (4,3)(3,2)(4,3)(3,2)(4,3)(2,0)(3,1) or in symbols $S_0S_1S_0S_1S_0S_2S_3$
- Coding redundancy in RL codes can be removed by Huffman coding:

  Fixed-length codes: 00 01 00 01 00 10 11

  Huffman codes: 1 01 1 01 1 001 000

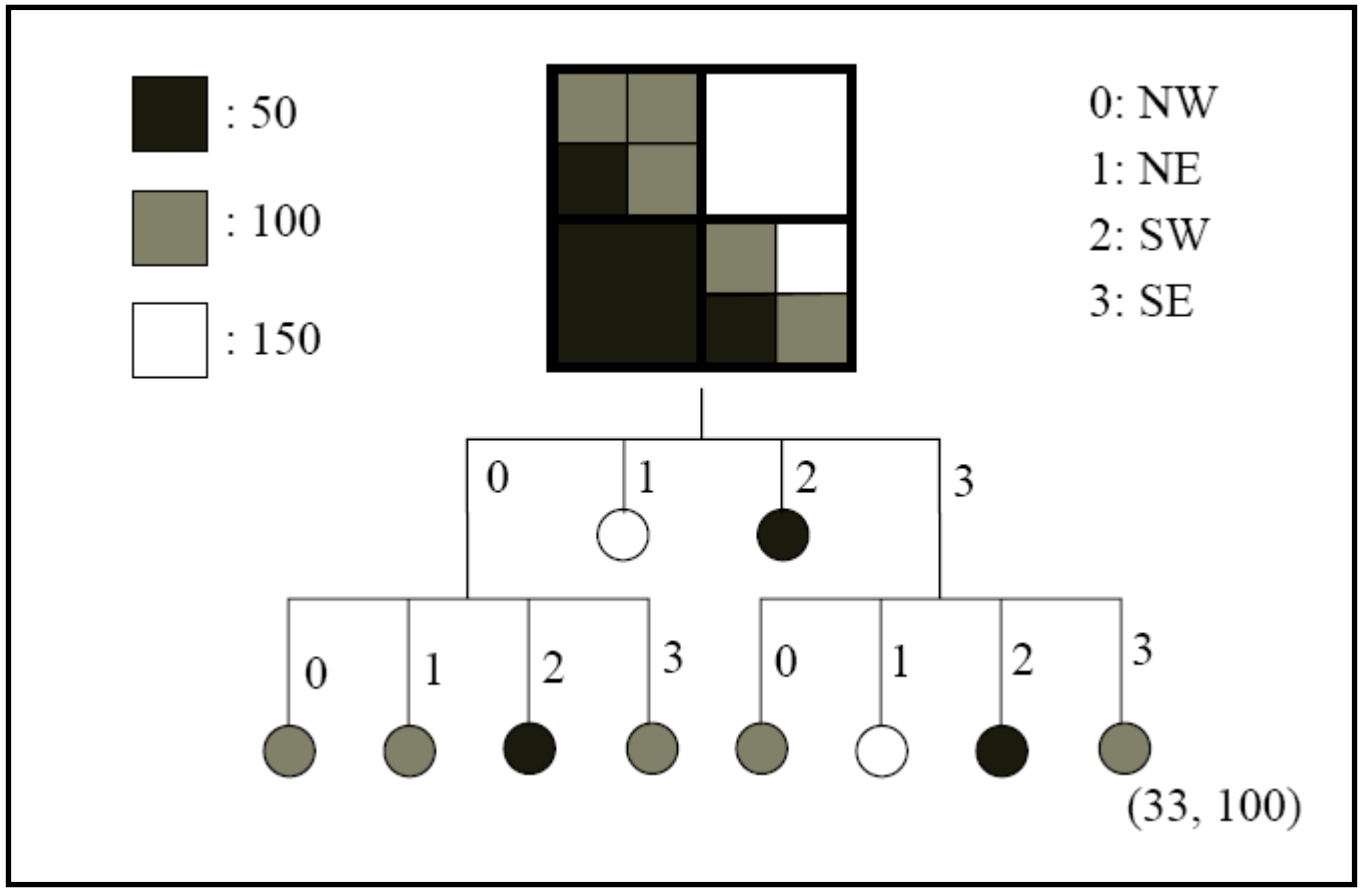
University of Reading

# Quad Tree Representation

- A popular data structure for digital images obtained by recursively splitting image (quadrant) into 4 quadrants. This generates a tree (called quad-tree) where each node has 4 children nodes
- The division of a quadrant is terminated if all pixels in the quadrant have identical intensities
- Compression is achieved by representing each leaf node (i.e. a node with no children) using a pair (leaf code, intensity)

University of Reading

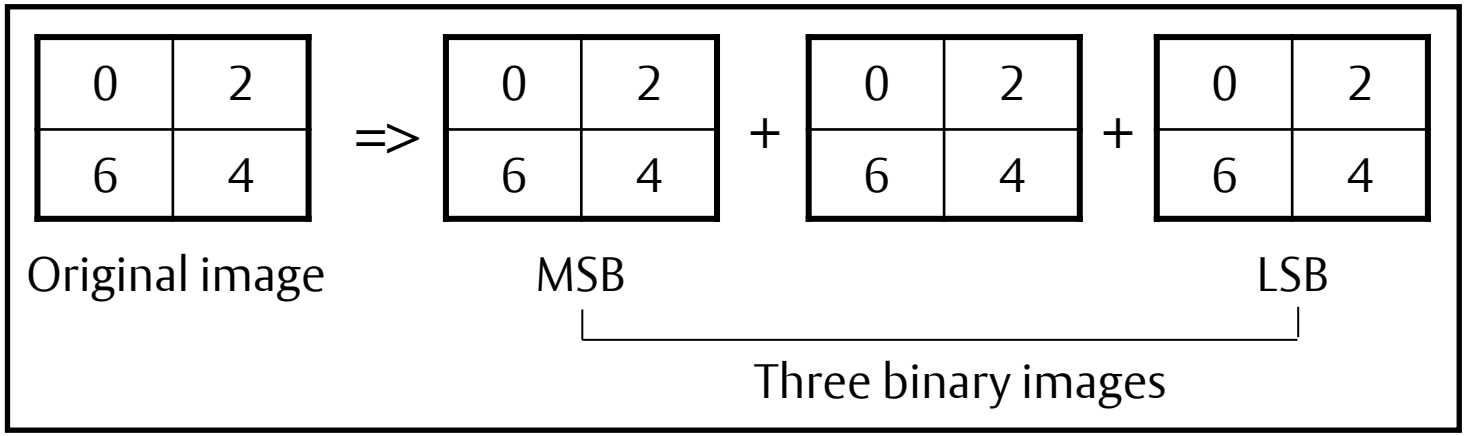
# Quad Tree Representation



University of Reading

# Bit Plane Coding

- An N-bit greyscale image may be regarded as a set of N binary images each of which corresponds to one bit (i.e. the MSB)




- Each binary image can the be coded using RLC or quad-tree.

University of Reading

# Predictive Coding

- The intensity of a pixel is predicted using some formula based on intensities of neighbouring pixels

| | | | |
|---|---|---|---|
| 0 | A | | |
| B | X | | |

$X' = (A+B)/2$

$D_X = X - X'$

- Difference ($D_X$) between predicted ($X'$) and actual ($X$) intensity value is then coded using RLC and/or Huffman coding
- Many prediction methods exist that may be used to determine the predicted value $X'$. The above shows a simple example.

University of Reading

# Lossy Predictive Coding

- Differences between predicted and actual intensities are quantized to pre-defined values
- Quantized differences are then coded using Huffman Coding
- The compression is lossy since quantization is involved

University of Reading

# Lossy Quad Trees

- Identical to lossy quad tree representations but stop dividing a quadrant if the pixels in the quadrant have *similar* intensities
  - i.e. the intensity variance is less than a threshold
- The average intensity of the quadrant is then assigned to all pixels in the quadrant
- We may also terminate the quadrant division if the intensity surface of the quadrant can be well approximated by a pre-defined surface
  - e.g. a sloped place; mathematically $Z=aX+bY+c$

University of Reading

# Block Truncation Coding

- Divide image into small blocks and for each block compute its average intensity $m_1$
- For each pixel in the block, if its intensity is less than $m_1$, set its intensity to A; otherwise to B
- The two greylevels A and B are chosen in such a way that the first and second moments of the intensity data in the block are preserved

# Block Truncation Coding

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_1$ | $a_2$ | $a_3$ | $a_4$ | | B | B | A | A | | 1 | 1 | 0 | 0 |
| $a_5$ | $a_6$ | $a_7$ | $a_8$ | | A | B | B | B | | 0 | 1 | 1 | 1 |
| $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | | A | A | A | B | | 0 | 0 | 0 | 1 |
| $a_{13}$ | $a_{14}$ | $a_{15}$ | $a_{16}$ | | A | A | A | B | | 0 | 0 | 0 | 1 |
| Original image | | | | | Bilevel image | | | | | bit-plane | | | |

$$m_1 = \frac{1}{N}\sum_{i=1}^{N} a_i \qquad A = m_1 - \sqrt{\frac{q(m_2 - m_1^2)}{N-q}}$$

$$m_2 = \frac{1}{N}\sum_{i=1}^{N} a_i^2 \qquad B = m_1 + \sqrt{\frac{(N-q)(m_2 - m_1^2)}{q}}$$

$m_1$: first moment; $m_2$: second moment; q: no. of B pixels
N: no. of pixels in block

- Each block is then described by the one-bit plane (which may be run-length coded) together with the 2 greylevels

University of Reading

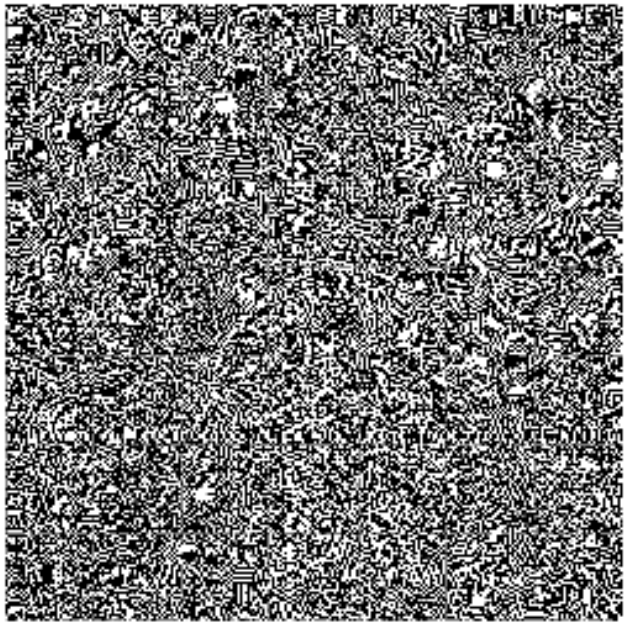
# Review and DCT Example

- Most of the image transforms we discussed earlier are capable of packing information into a much smaller number of items – *transform coefficients*
- Transform coding achieves compression by only keeping the small number of significant information-carrying components
- Among all image transforms DCT is by far the most widely used transform in image compression (see last lecture)
- In practice it's implemented by
  - Dividing image into small blocks (e.g. 8x8)
  - Perform DCT on each block
  - Quantize DCT coefficients
  - Code quantized coefficients using RLC and/or Huffman coding
- DCT decompression is performed by taking the opposite of the above operations in the reverse order (no dequantization, of course)

University of Reading

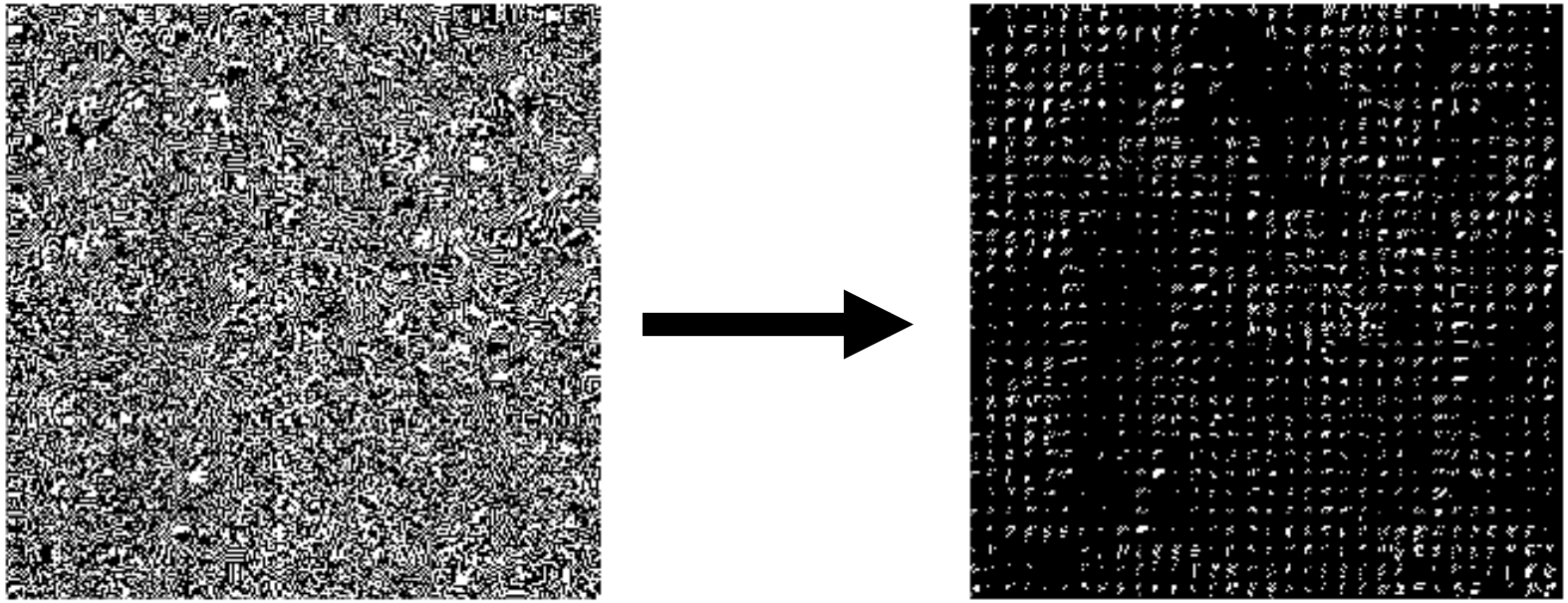
# Pepper Example

Each 8x8 block processed with DCT



University of Reading

# Pepper Example



Each element in each block of the image is quantized using a quantization matrix of quality level 50. Many of the elements become zeroed out, and the image takes up much less space to store

University of Reading

# Pepper Example

The image can now be decompressed using the inverse DCT. At quality level 50 there is almost no visible loss in this image, but there is high compression. At lower quality levels, the quality reduces significantly, but the compression increases moderately

Original Image

Quality 50:
84% Zeros

University of Reading

# Pepper Example

Quality 20:
91% Zeros

Quality 10:
94% Zeros

University of Reading

# Further Example

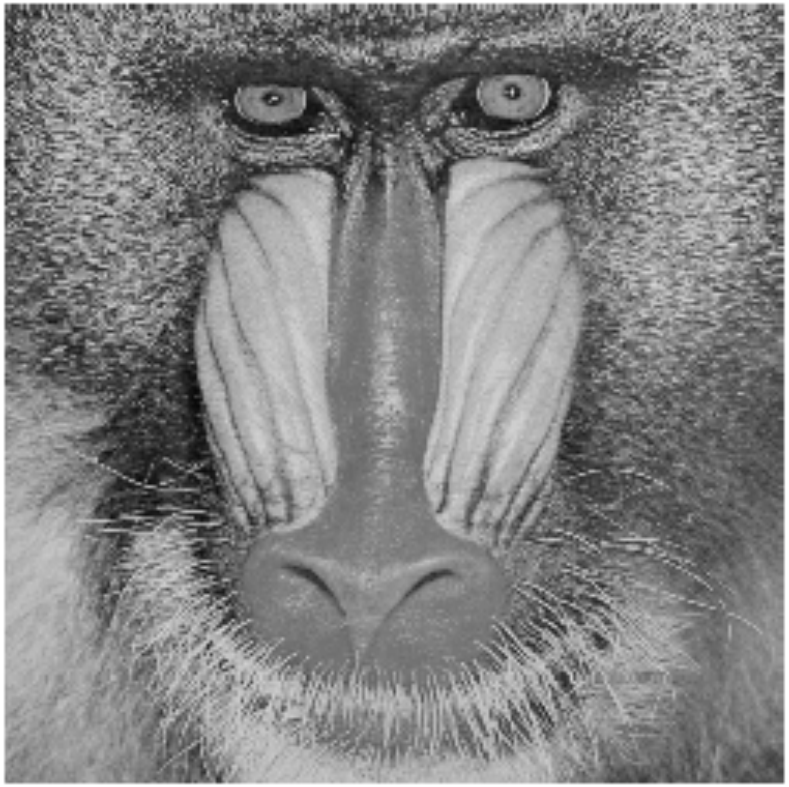
Original Image

Quality 15:
90% Zeros

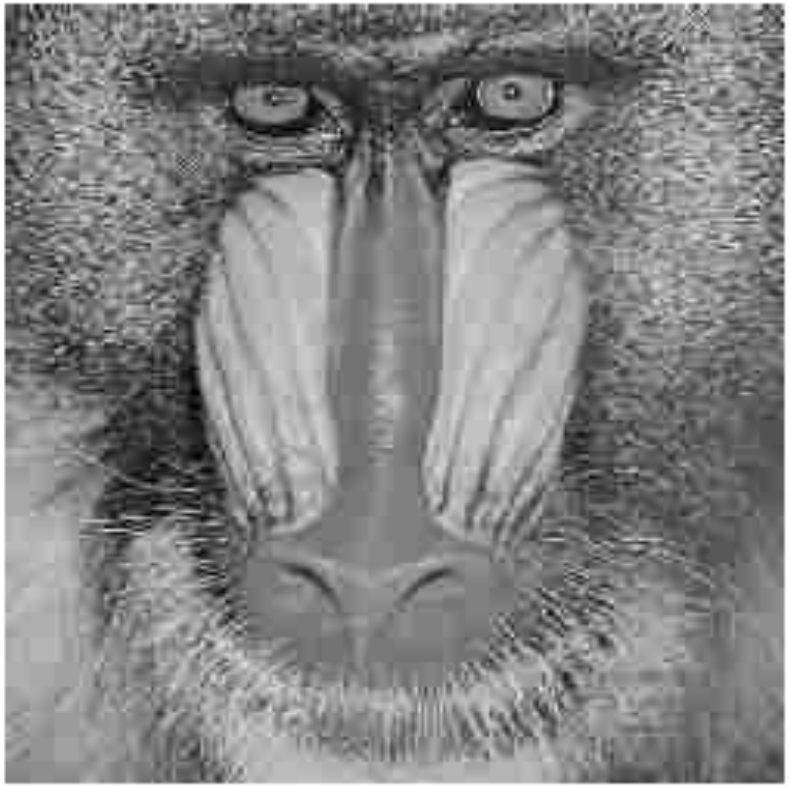
High contrast images, or images with a large proportion of high frequencies do not compress as well as smooth, low frequency images

University of Reading

# Further Example



Original Image



Quality 15:
88% Zeros

University of Reading

# Colour Image Compression

- The algorithms described can be easily extended to colour images
  - By processing each of the colour planes separately
  - Transforming image from RGB representation to other convenient representations, e.g. YUV, in which processing becomes easier

University of Reading

# Other Coding Methods

- Arithmetic Coding
- Hybrid Coding
- Sub-band / Wavelet Coding
- Contour and Texture Coding
- Fractals
- Model-Based Coding
- Document/Binary Image Coding
- Image Sequence Coding
- Colour and Multispectral Coding
- JPEG, JBIG, MPEG
- …