Lecture 2: Image & Video Coding Techniques (II) -- Fundaments

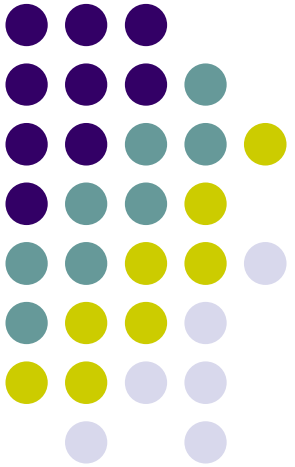
A/Prof. Jian Zhang

NICTA & CSE UNSW

COMP9519 Multimedia Systems

S2 2009

jzhang@cse.unsw.edu.au



NICTA

Pixel Representation *Tutorial 1*

- $Y_d C_b C_r$ Colour Space

For digital component signal (CCIR Rec 601), 8-bit digital variables are used, however:

1. Full digital range is not used to give working margins for coding and filtering.
2. RGB to $Y_d C_b C_r$ conversion is given by

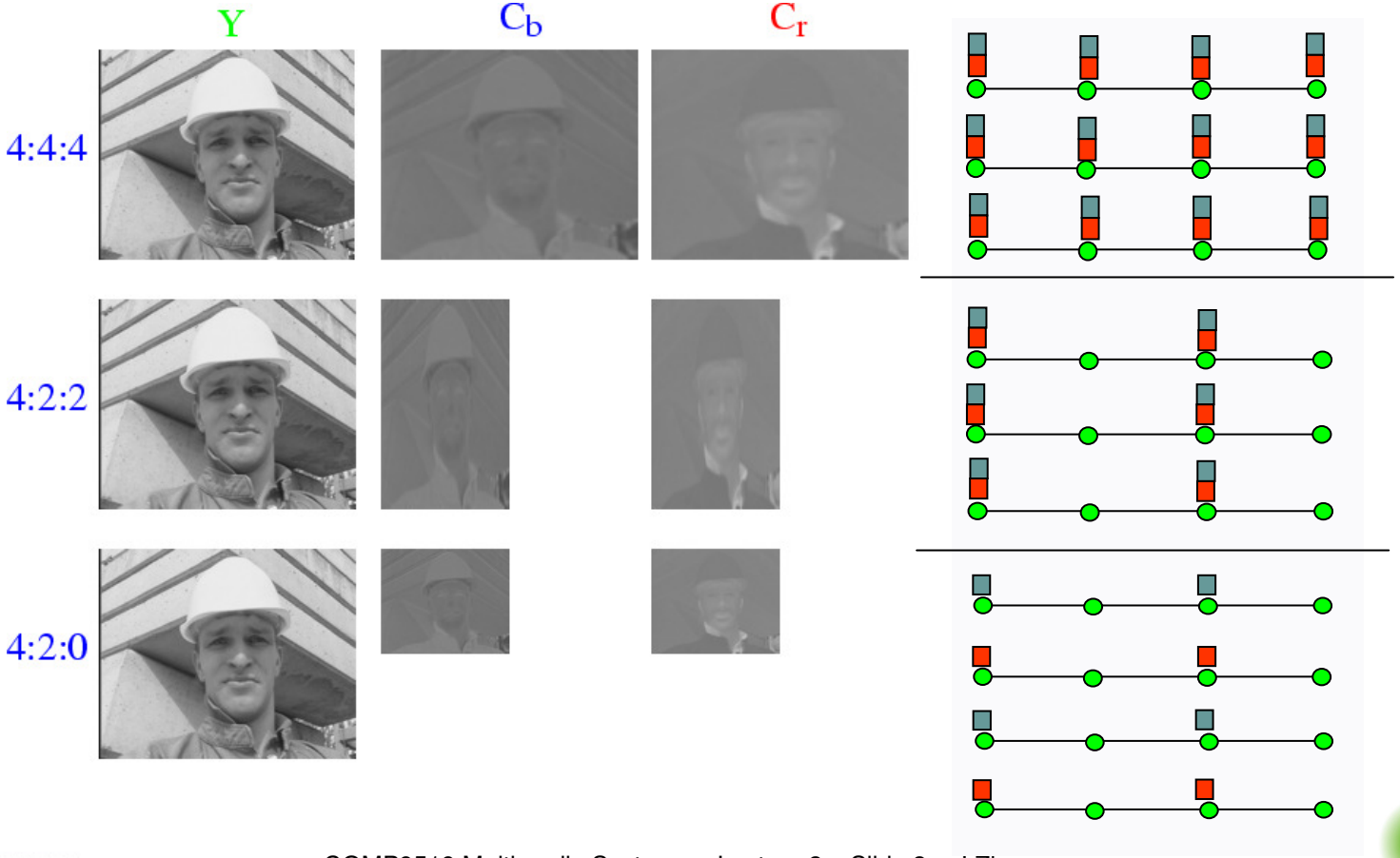
$$\begin{bmatrix} Y_d \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \qquad \begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y_d - 16 \\ C_b - 128 \\ C_r - 128 \end{bmatrix}$$

The positive/negative values of U and V are scaled and zero shifted in a transformation to the Cb and Cr coordinates.

where digital luminance, Yd, has a rang of (16-235) with 220 levels starting at 16, and digital chrominance difference signals, Cb and Cr, have a range of (16-240) with 225 levels centered at 128.

Chrominance sub-sampling *Tutorial 1*

- Human vision is relatively insensitive to chrominance. For this reason, chrominance is often sub-sampled.
- Chrominance sub-sampling is specified as a three-element ratio.



Chrominance sub-sampling

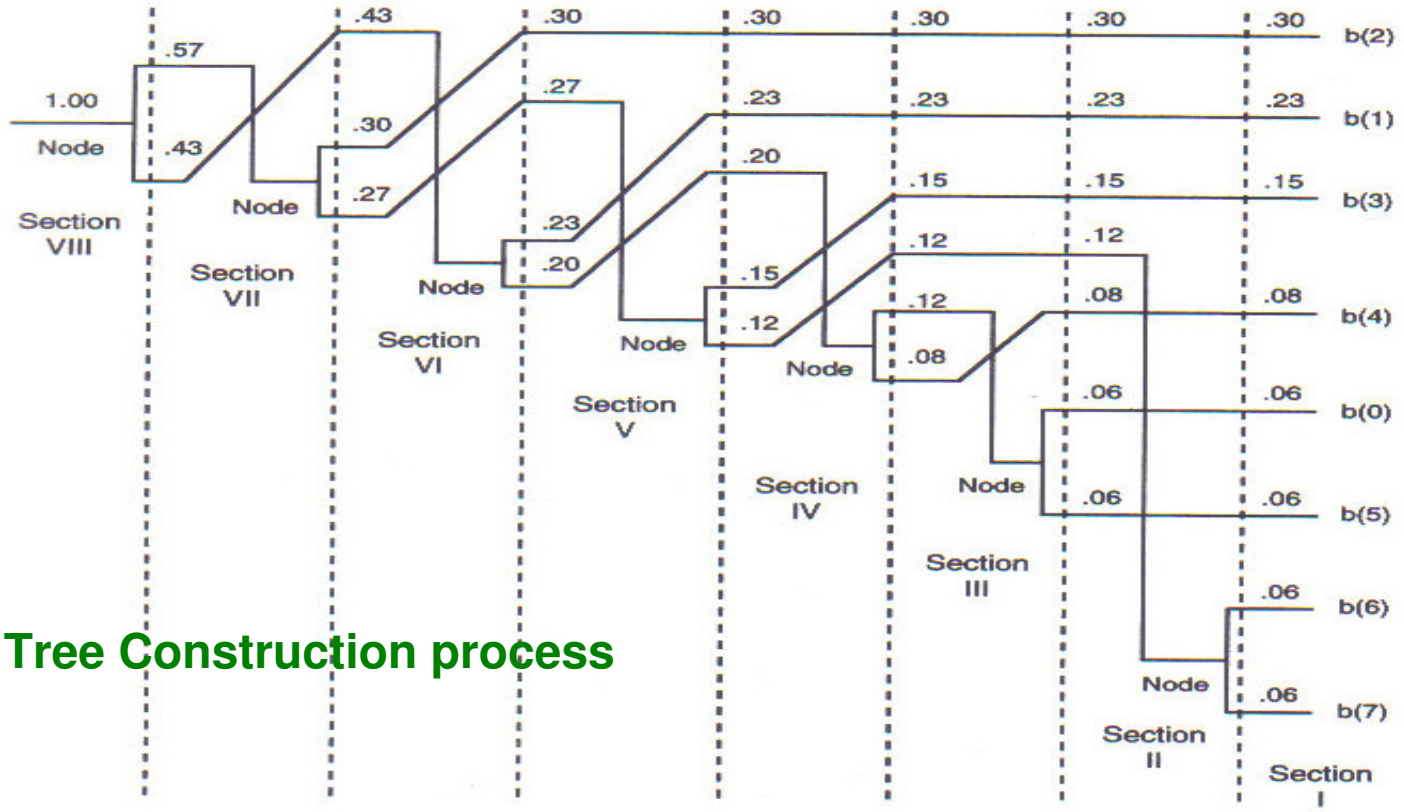
Tutorial 1

- In 4:4:4 format: Y, Cr & Cb – 720 x 576 pixels per frame
- In 4:2:2 format: Y – 720 x 576 and Cr & Cb – 360 x 576 pixels per frame
- In 4:2:0 format: Y – 720 x 576 and Cr & Cb – 360 x 288 pixels per frame
  - A commonly used format is **4:2:0** which is obtained by sub-sampling each colour component of 4:2:2 source vertically to reduce the number of lines to 288;

Introduction to Entropy Coding

Tutorial 1

1. From Right to Left, 2. Two bottom-most branches are formed a node
3. Reorder probabilities into descending order

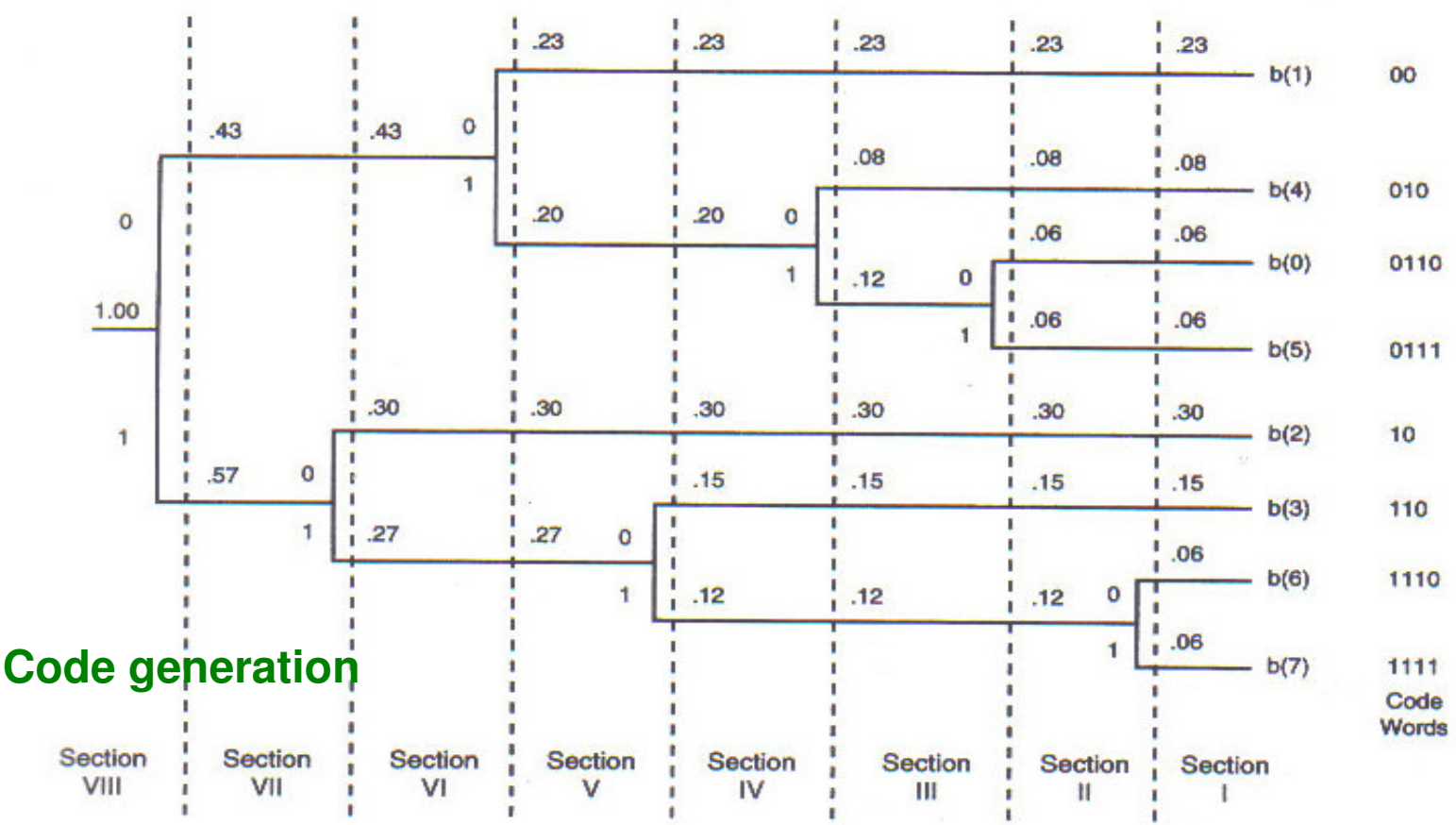


Tree Construction process

Introduction to Entropy Coding

Tutorial 1

1) Re-arrange the tree to eliminate crossovers, 2) The coding proceeds from left to right, 3) 0– step up and 1– step down.



Code generation

2.1 Introduction to Entropy Coding

- Arithmetic Coding
  - It overcomes limitation of Huffman coding: non-integer length coding, and probability distribution can be derived in real-time
  - It operates by replacing a stream of input symbols with a single floating point output number.
  - Consider the following symbols with probabilities

A sub-interval cab be defined by its lower end point and its width or lower and upper end points

TABLE 5.10
Source Alphabet and Cumulative Probabilities in Example 5.12

Source Symbol	Occurrence Probability	Associated Subintervals	CP
S_1	0.3	[0, 0.3)	0
S_2	0.1	[0.3, 0.4)	0.3
S_3	0.2	[0.4, 0.6)	0.4
S_4	0.05	[0.6, 0.65)	0.6
S_5	0.1	[0.65, 0.75)	0.65
S_6	0.25	[0.75, 1.0)	0.75

The sum of preceding Probabilities known as Cumulative Probability

$$CP(s_i) = \sum_{j=1}^{i-1} p(s_i)$$

Where $CP(S_1)=0$ is defined

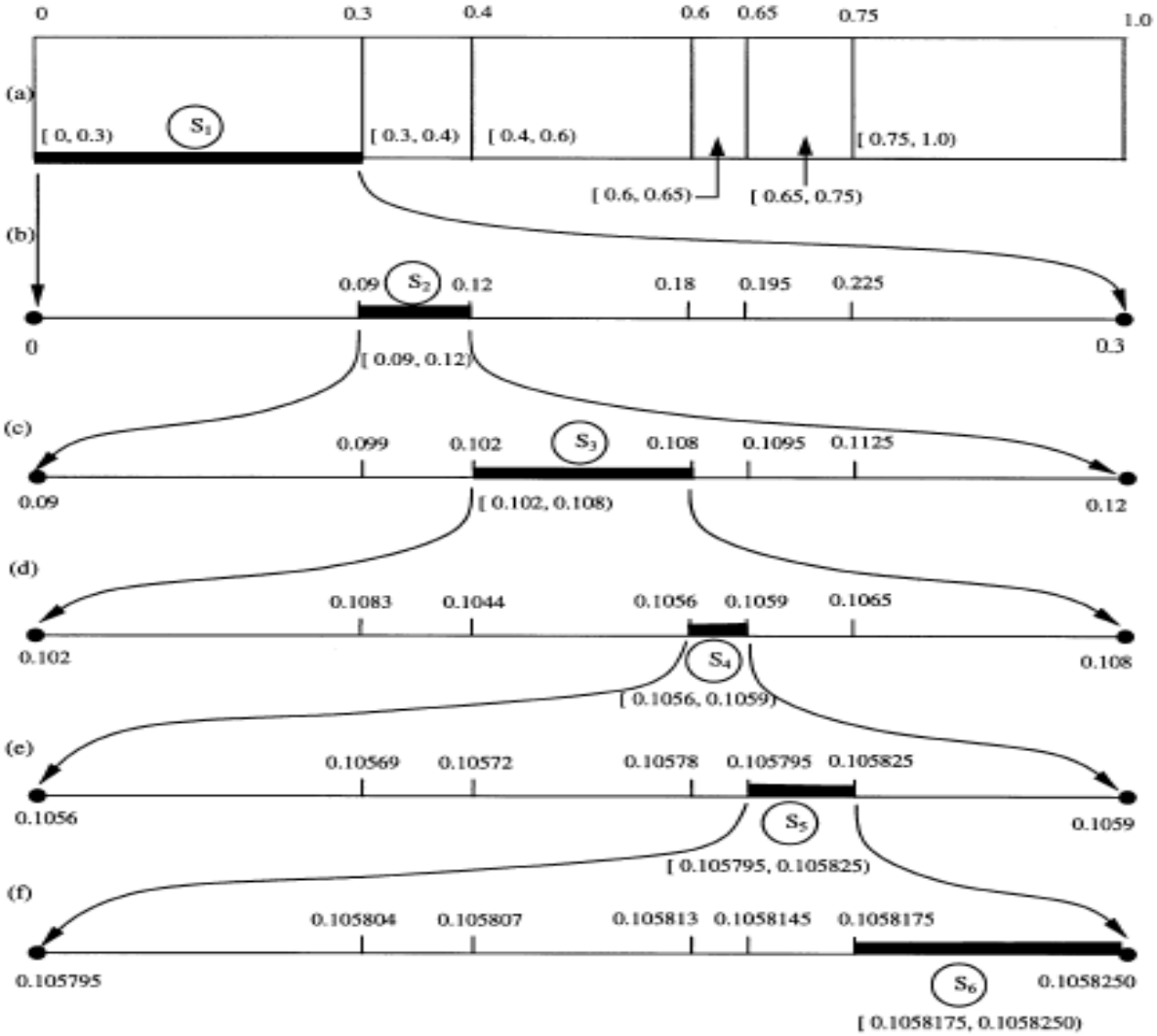
2.1 Introduction to Entropy Coding

- Arithmetic Coding
  - Suppose we wish to encode the string: **S_1; S_2; S_3; S_4; S_5; S_6;**
  - We start with the interval [L,H) and set to [0,1) for 6 symbols.
  - Since the first symbol is **S_1**, we pick up its subinterval [L,H) = [0.0,0.3), and any real symbols can be considered as disjoint to be divided in the same way.
  - To encode the **S_2**, we use the same procedure as used in above to divide the interval [0,0.3) into six sub-intervals. We pick up the **S_2** subinterval [0.09,0.12)
  - The subinterval recursion is equivalent to the two recursions
    - End point recursion and width recursion

$$L_{new} = L_{current} + W_{current} \cdot CP_{new}$$ Low end point recursion

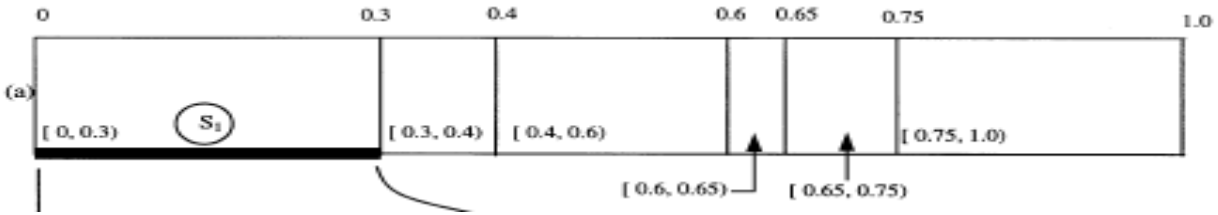
$$W_{new} = W_{current} \cdot P(S_i)$$ The width recursion

2.1 Introduction to Entropy Coding



2.1 Introduction to Entropy Coding

- Arithmetic Decoding
  - For encoding, the input is a source symbol string and the output is a subinterval (called the final subinterval). For our case, it is [0.1058175,0.1058250).
  - Decoding sort of reverses what encoding has done.
    - The decoder knows the encoding procedure and therefore has the information contained in the following figure



    - It compares the lower end point of the final subinterval 0.1058175 with all the end points in the above figure

0<0.1058175<0.3 ! ! !

2.1 Introduction to Entropy Coding

- The lower end falls into the subinterval associated with the symbol S_1 which is first decoded.
- After the first symbol is decoded, 0.09<0.1058175<0.12 ! ! !. The lower end is contained in the subinterval covers the S_2
- Repeat the process until the symbols S_4; S_5; S_6 are subsequently decoded.
- Using Huffman coding to encode the same symbols

TABLE 5.9
Source Alphabet and Huffman Codes in Example 5.9

Source Symbol	Occurrence Probability	Codeword Assigned	Length of Codeword
S_1	0.3	00	2
S_2	0.1	101	3
S_3	0.2	11	2
S_4	0.05	1001	4
S_5	0.1	1000	4
S_6	0.25	01	2

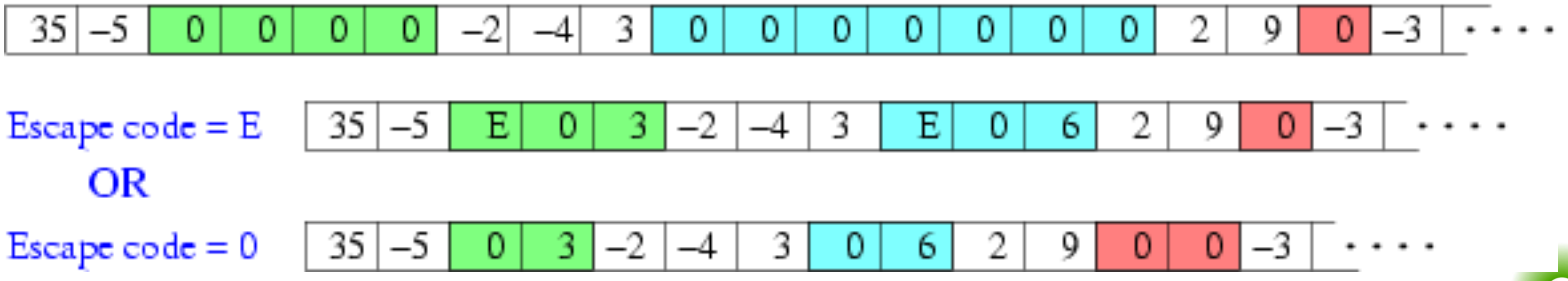
2.1 Introduction to Entropy Coding

- Human coding – converts each source symbol into a fixed codeword (but variable length). The output of $S_1; S_2; S_3; S_4; S_5; S_6$ are:
  - 00,101,11,1001,1000,01 which is a 17-bit code string
- Arithmetic coding converts a source symbol to a code symbol string
  - 0.1058175 = 0.0001101111111111 ~ [0.1058175, 0.1058250). Which is 15-bit code string
- This is a simple example about the arithmetic coding is more efficient than Human coding
- It is obvious that the width of the final subinterval becomes smaller when the length of the source symbol string become larger and larger. This was a big problem to widely apply the arithmetic coding and was solved in the late 1970s.
- Arithmetic coding now becomes an increasingly important coding

2.1 Introduction to Entropy Coding

Ref: H.Wu

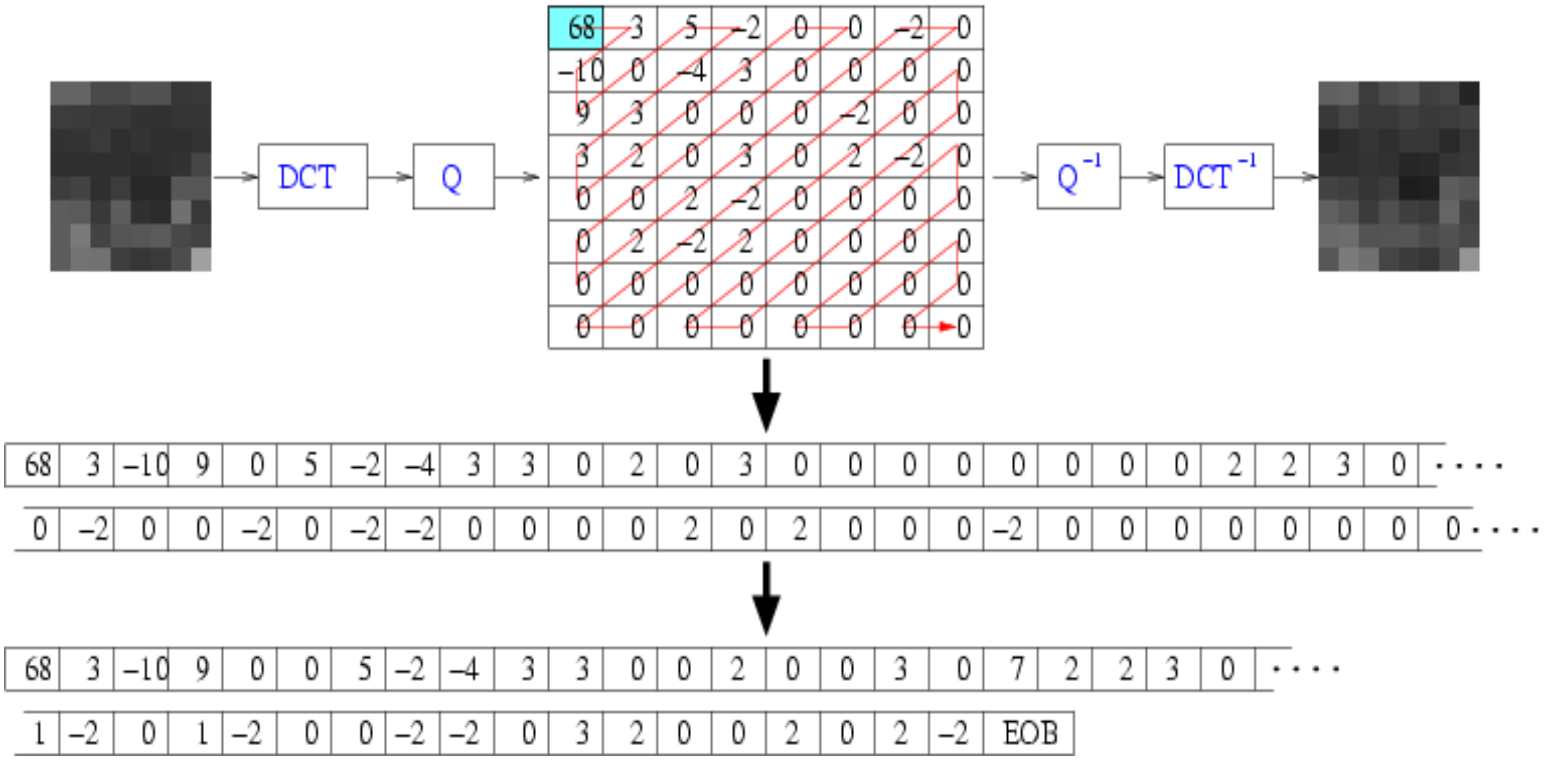
- Run-length Coding
  - In run-length coding, a run of consecutive identical symbols is combined together and represented by a single codeword.
  - The coding of facsimile information is one application of run length coding.
  - The various run-lengths are then represented by a variable length (e.g. Huffman) codeword. In the case of facsimile transmission, separate VLCs are defined for white and black
  - Use an escape code (out side of symbol set or '0'), followed by the run-length coded symbol and number of repetitions of symbol.



2.1 Introduction to Entropy Coding

- Run-length Coding (Example)

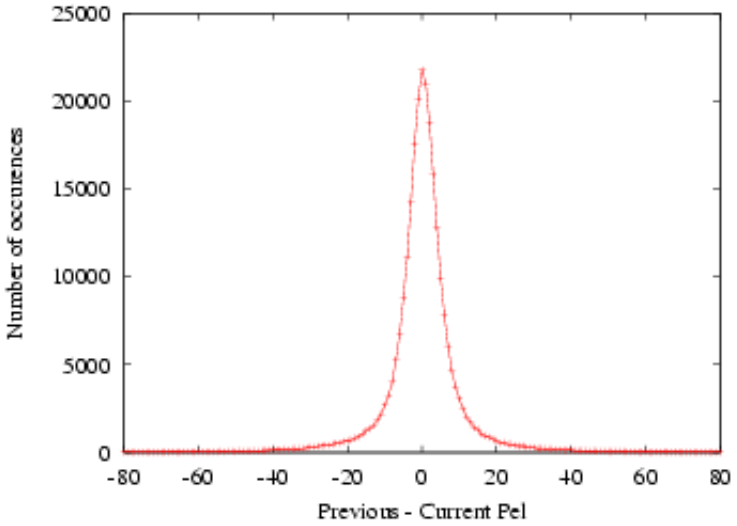
[Ref: H. Wu]

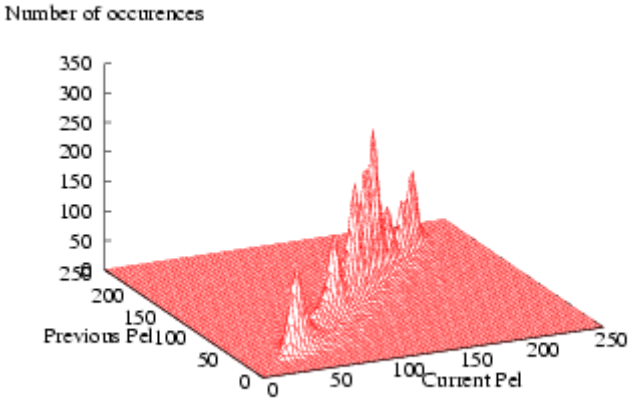


Run-length (using '0' as the escape code) coding DCT Coefficients

2.2 Spatial Statistical Redundancy

- Spatial redundancy is existed among pixels within a single frame of image. Especially, neighboring pixels are highly correlated.



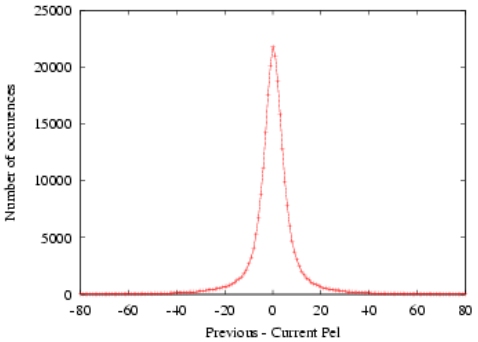


Ref: H. Wu

"Lena" image

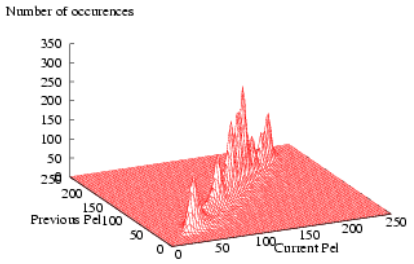
2.3 Introduction to Differential Coding

- Spatial redundancy is existed among pixels within a single frame of image. Especially, neighboring pixels are highly correlated.



•Strong correlation exists between adjacent pixel spatially

•A pixel is coded based on the difference between its value and a predicted value.

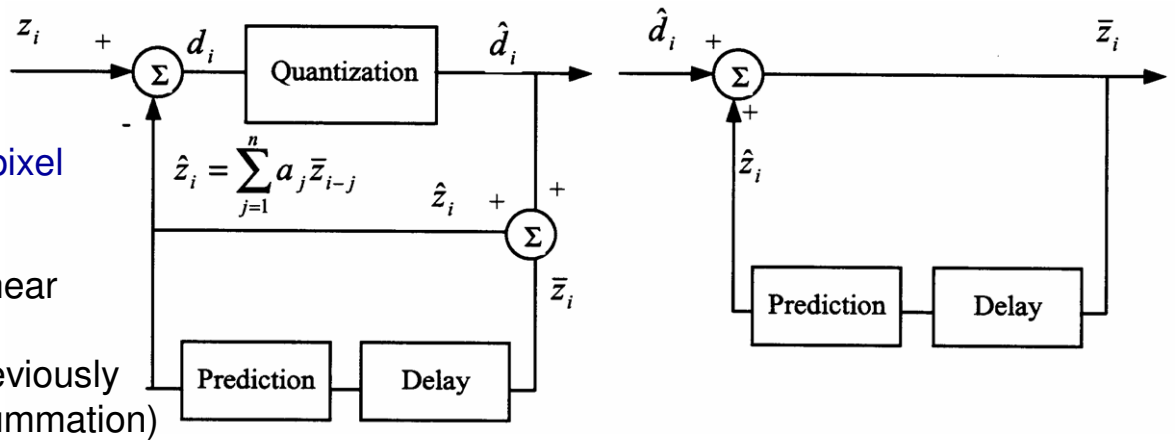


"Lena" image

Ref: H. Wu **2.3.1 Spatial Statistical Redundancy**

2.3 Introduction to Differential Coding

- By exploring spatial/temporal inter-pixel correlation, prediction and quantization coding scheme achieve efficiency and yet computationally simple coding technique.
- When the prediction error (difference error) is quantized, the differential coding is called Differential Pulse Code Modulation (DPCM)
-



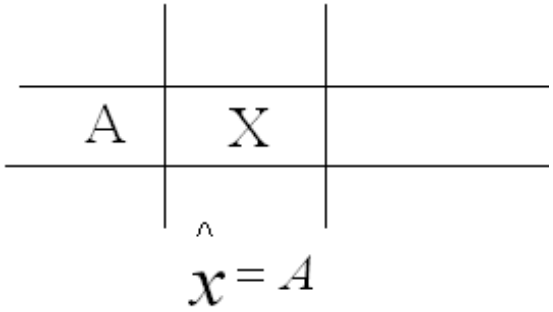
Where z_i is current input pixel

Where $\hat{z}_i = \sum_{j=1}^{n} a_i \bar{z}_{i-j}$ is a linear prediction function of N previously reconstructed samples (Summation)

$$\hat{z} = f(z_1, z_2 \ldots\ldots z_n)$$

2.3 Introduction to Differential Coding

- In this approach, the preceding pixel on the video/image line is used to predict the next pixel. The prediction error is then transmitted
- Example



91	99 (+7)	96 (-3)
93	101 (+8)	97 (-4)
101	103 (+2)	104 (+1)

- This approach achieves compression because the entropy of the prediction error is less than the entropy of the original image.
- The better the prediction , the lower the entropy of the prediction error.

2.3 Introduction to Differential Coding

"Lena" image

1-D prediction error of "Lena" image

Entropy = 3.06 bits/pixel

2.3 Introduction to Differential Coding

- This approach can be applied to a two dimensional predictor by taking advantage of vertical correlation as well.
- Example

B	C	D
A	X	

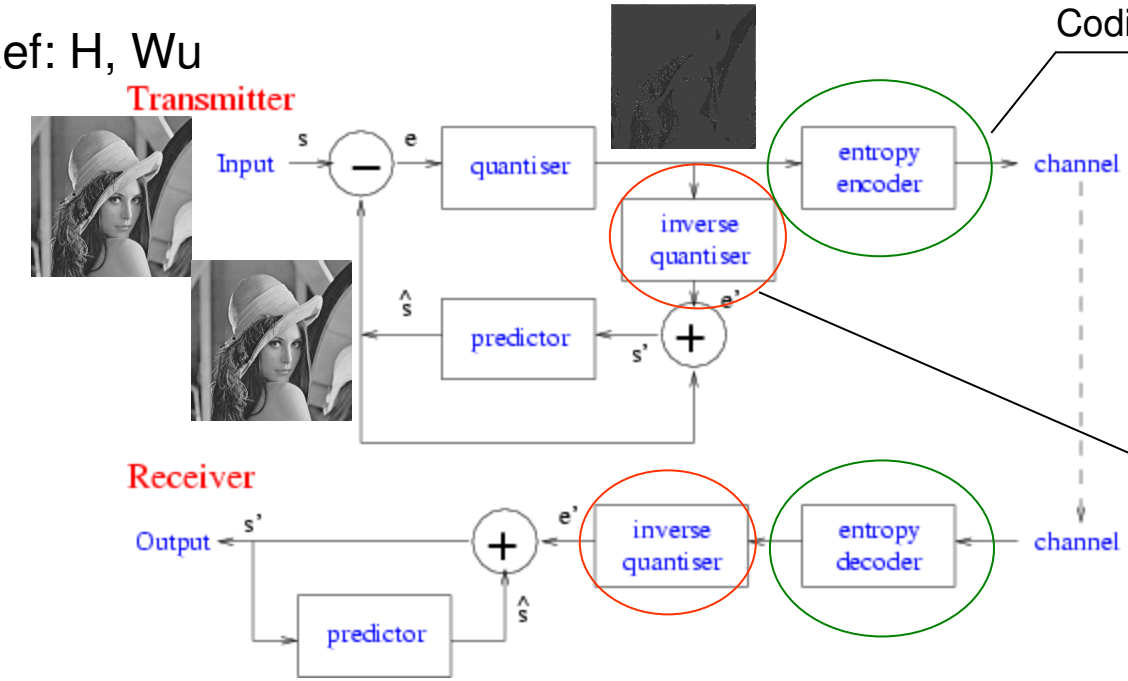
$$\hat{x} = k_1 A + k_2 B + k_3 C + K_4 D$$

Entropy = 1.44. bits/pixel

2-D prediction error of "Lena" image

2.3 Introduction to Differential Coding

Ref: H, Wu



Coding efficiently

Code assigner is a strategy to code the symbols of coefficients after the transform.

Improve accuracy

•Prediction error: $e(n) = s(n) - \hat{s}(n)$

Where $\hat{s}(n) = f[s'(n-1), s'(n-2), s'(n-3), \ldots]$

•Reconstruction: $s'(n) = \hat{s}(n) + e'(n)$

2.4 Introduction to Image Quantization

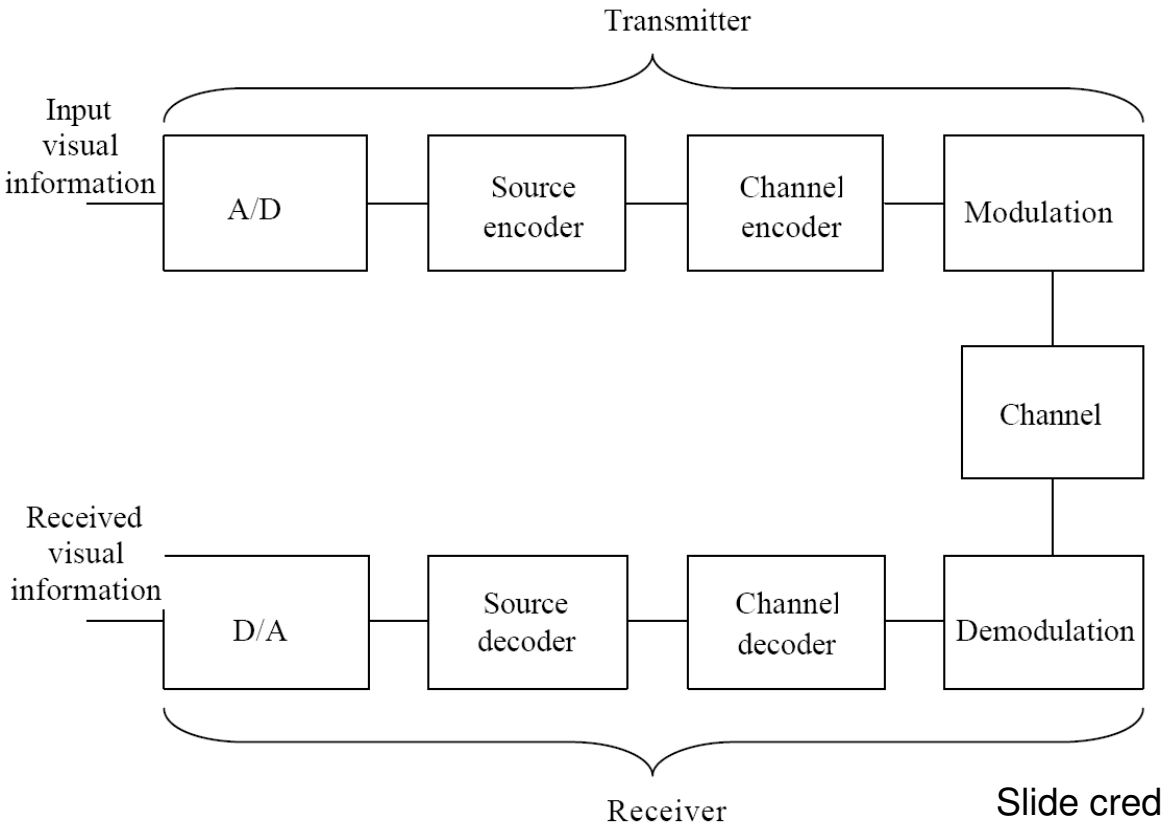
- Quantization
  - The major mechanism for loss in image/video codecs (encoder/decoder) is as a result of quantization
  - However, when quantization is performed carefully in frequency domain, it is difficulty for a viewer to notice any degradation.

Original

JPEG (compressed 70%)

2.5 Quantization

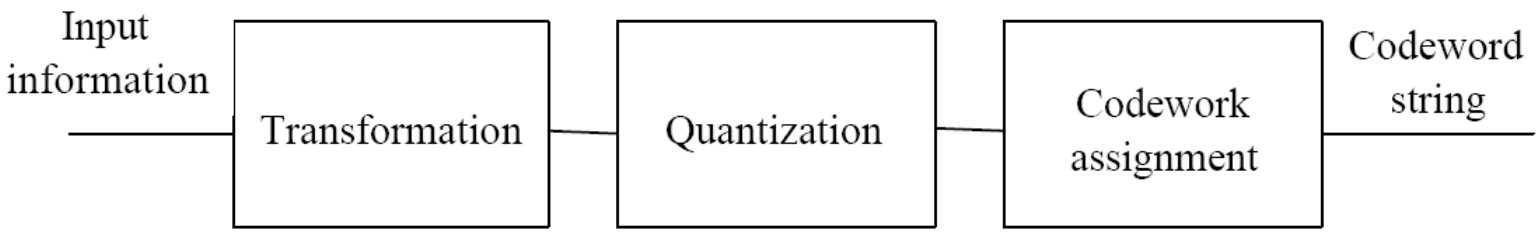
- Block diagram of a visual communication system



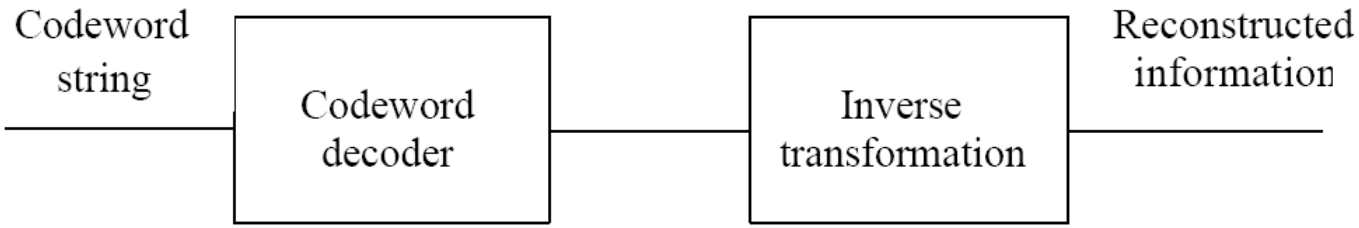
Slide credit:. Yun Qing Shi

2.5 Quantization

- Block diagram of a source encoder and decoder



(a) source encoder



Slide credit: Yun Qing Shi

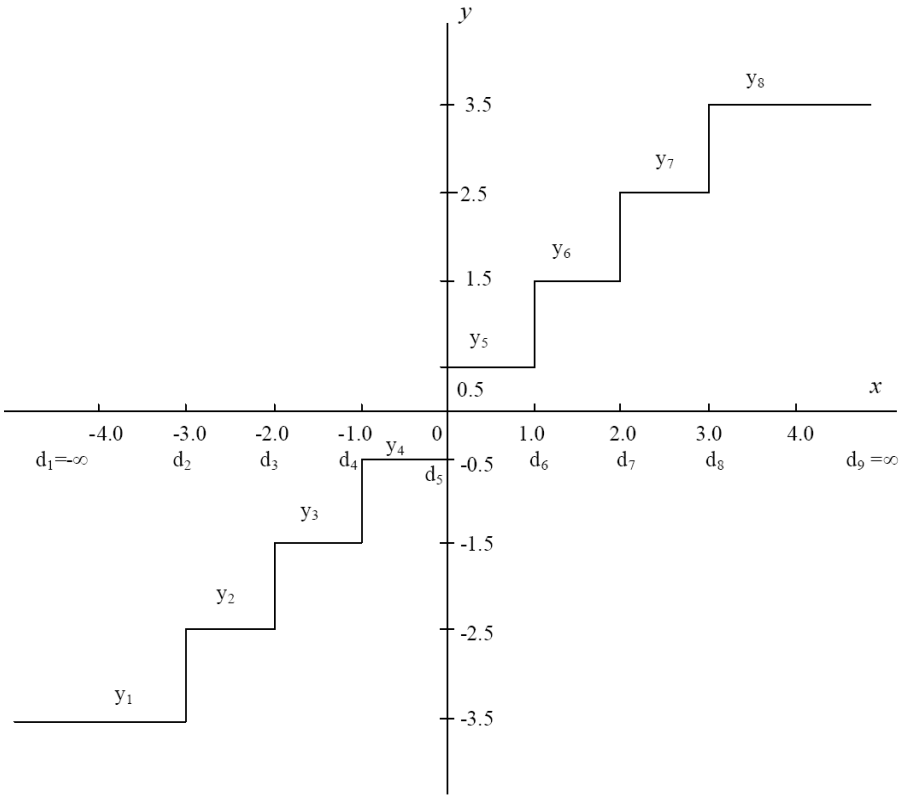
2.5 Quantization

- Quantization is an irreversible process
- Quantization is a source of information loss
- Quantization is a critical stage in image and video compression including:
  - the distortion of reconstructed image and video
  - the bit rate of the encoder.

Slide credit:. Yun Qing Shi

2.5 Quantization

- Uniform quantization



Slide credit:. Yun Qing Shi

Input-out characterise of a uniform mid-tread quantizer

2.5 Quantization

- Quantization Distortion
  - In terms of objective evaluation, the quantisation error e_q (or quantization noise) can be defined as follows:

$$e_q = x - Q(x)$$

    - *x* and *Q(x)* are input and quantized output, respectively.
- Mean square quantisation error, MSEq

$$MSE_q = \sum_{i=1}^{N} \int_{d_i}^{d_{i+1}} (x - Q(x))^2 f_x(x) dx$$

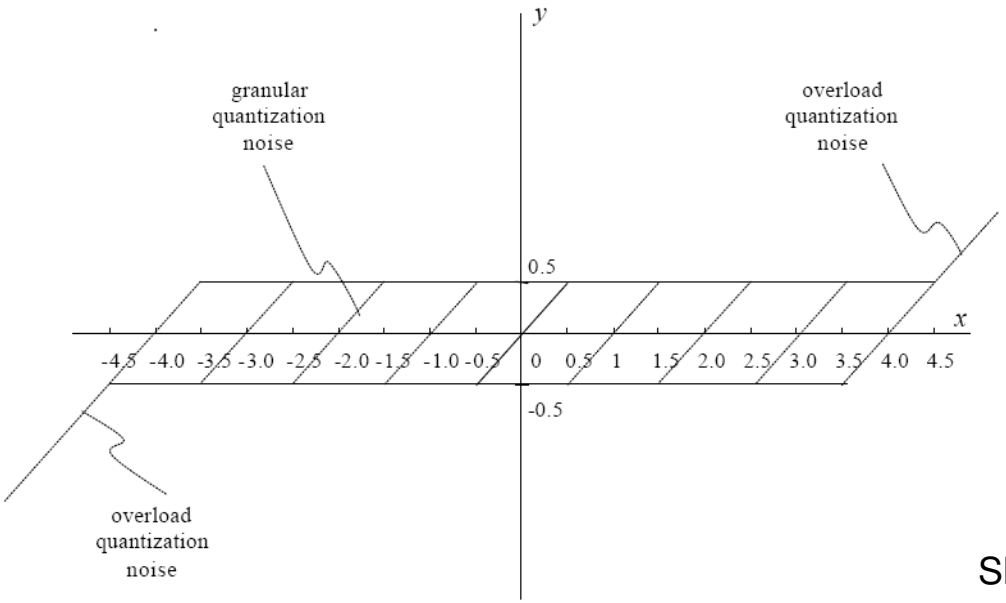
  - $f_x(x)$: probability density function *(pdf)* the outer decision levels may be $-\infty$ or ∞ when the *pdf,* $f_x(x)$ remains unchanged fewer reconstruction levels (small N, coarse qantization) results in more distortion.

Slide credit:. Yun Qing Shi

2.5 Quantization

- Odd symmetry of the input-output characteristic respect to the x=0 axis

  Implies that : $E(x) = 0$ and $MSE_q = \sigma_q^2$ is the variance of the quantization nose equation



Slide credit:. Yun Qing Shi

Quantization noise of the uniform mid-tread quantizer

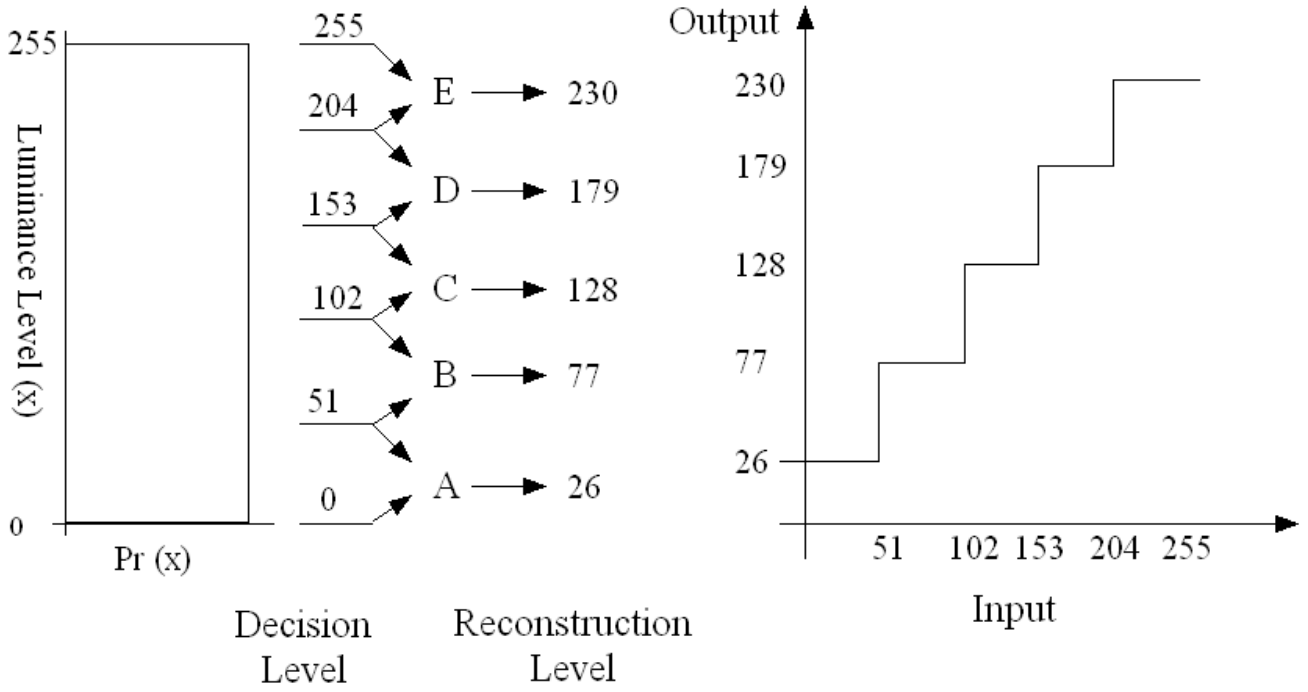
2.5 Quantization

- The quantization noise is signal dependent.
  - Granular quantization noise is bounded by $\pm 0.5\Delta$
  - Right/left most intervals are unbounded as the input approaches either $-\infty$ or ∞ overloaded noise

Slide credit:. Yun Qing Shi

2.6 Introduction to Image Quantization

- Linear Quantization
  - The simplest form of quantize is a linear quantize where the quantize step is constant



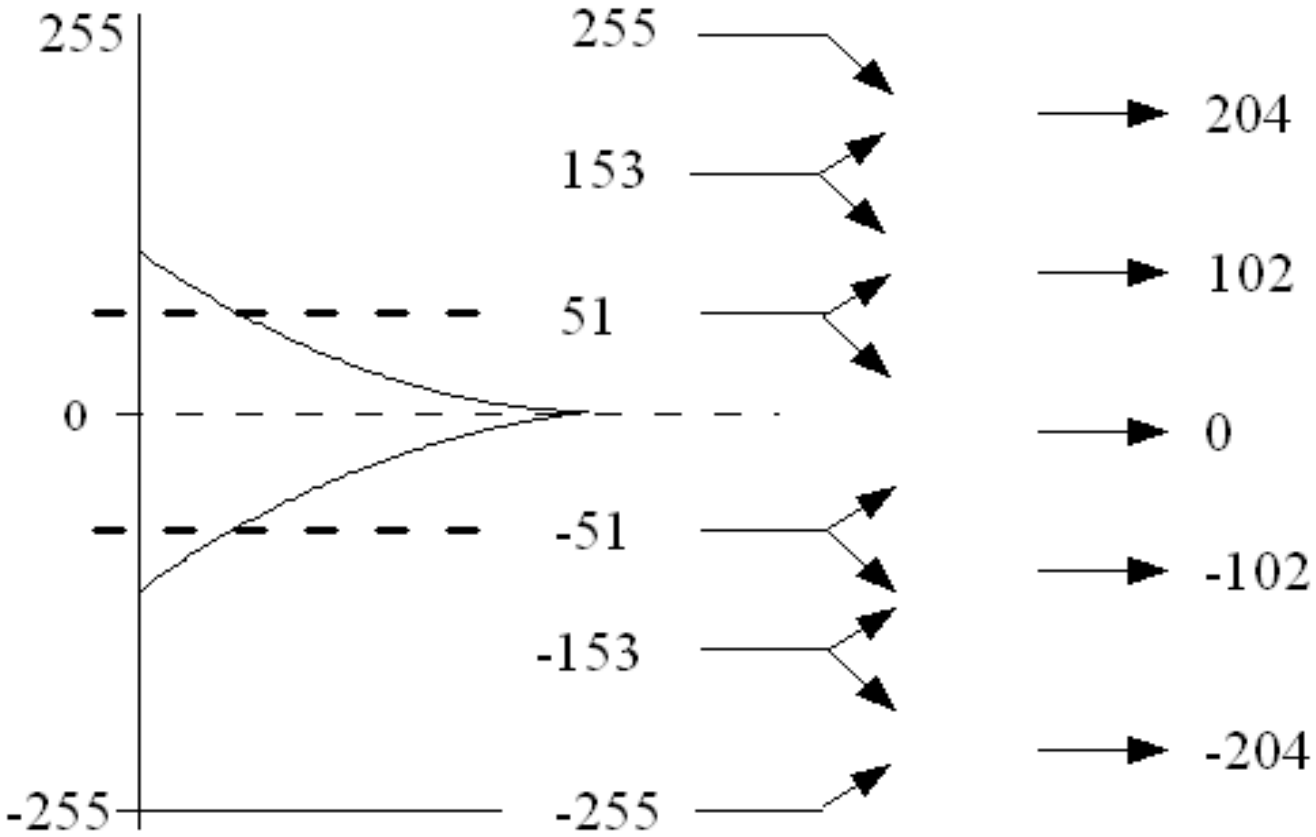
  - Maximum error = ± Half quantize step size

2.6 Introduction to Image Quantization

- Linear/Non-linear Quantization
  - For the case of the linear quantizer, a simple relationship exits between the number of symbols after quantization and the error introduced by quantization
  - This approach is sensible for a signal with a uniform distribution of values such as an original image
  - However, it makes very little sense for difference image where the differences are non-uniformly distributed about zero,
  - For these types of signals, a non-linear quantization is needed.

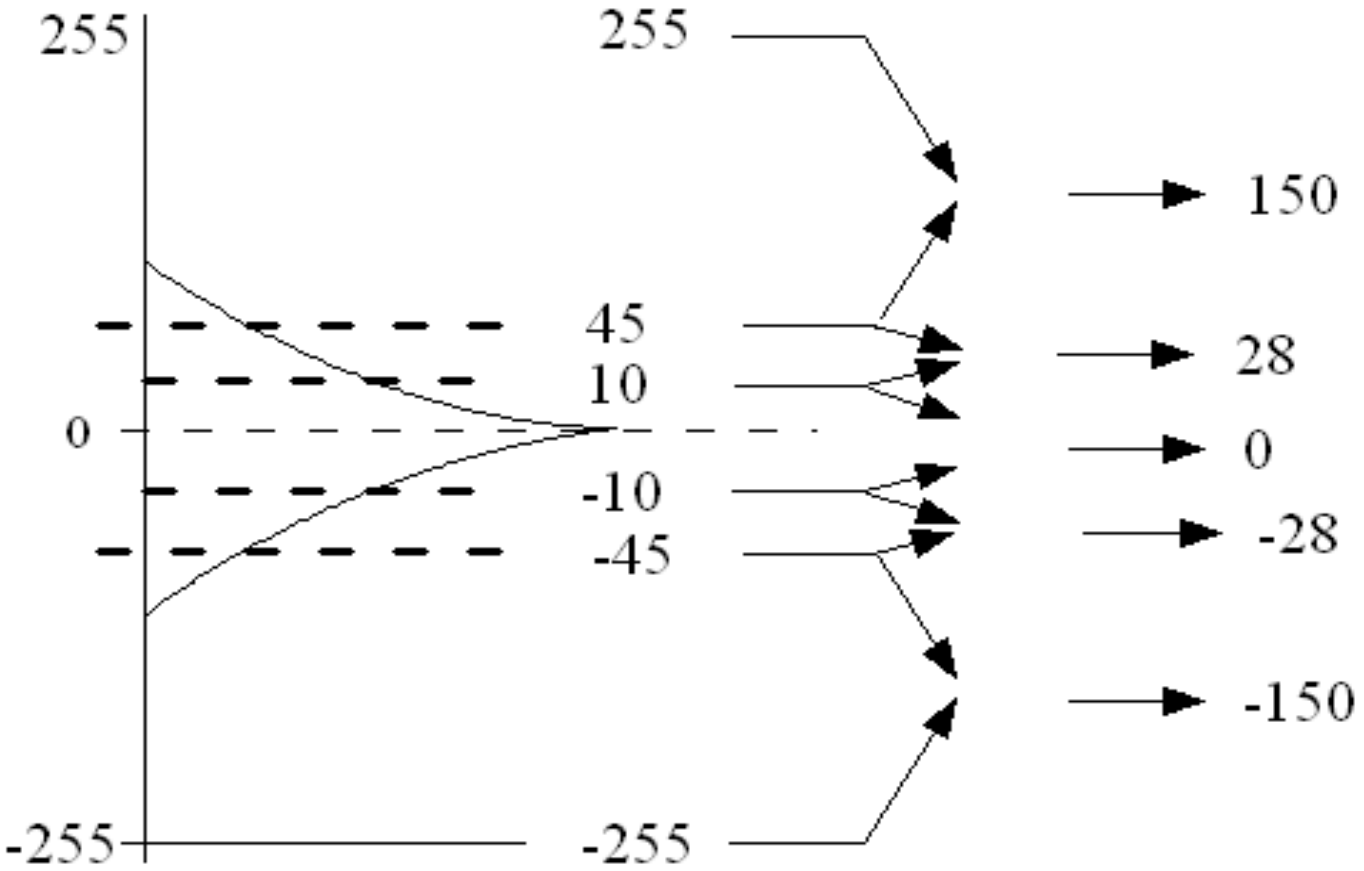
2.6 Introduction to Image Quantization

- Linear Quantizer Applied to a Difference Image



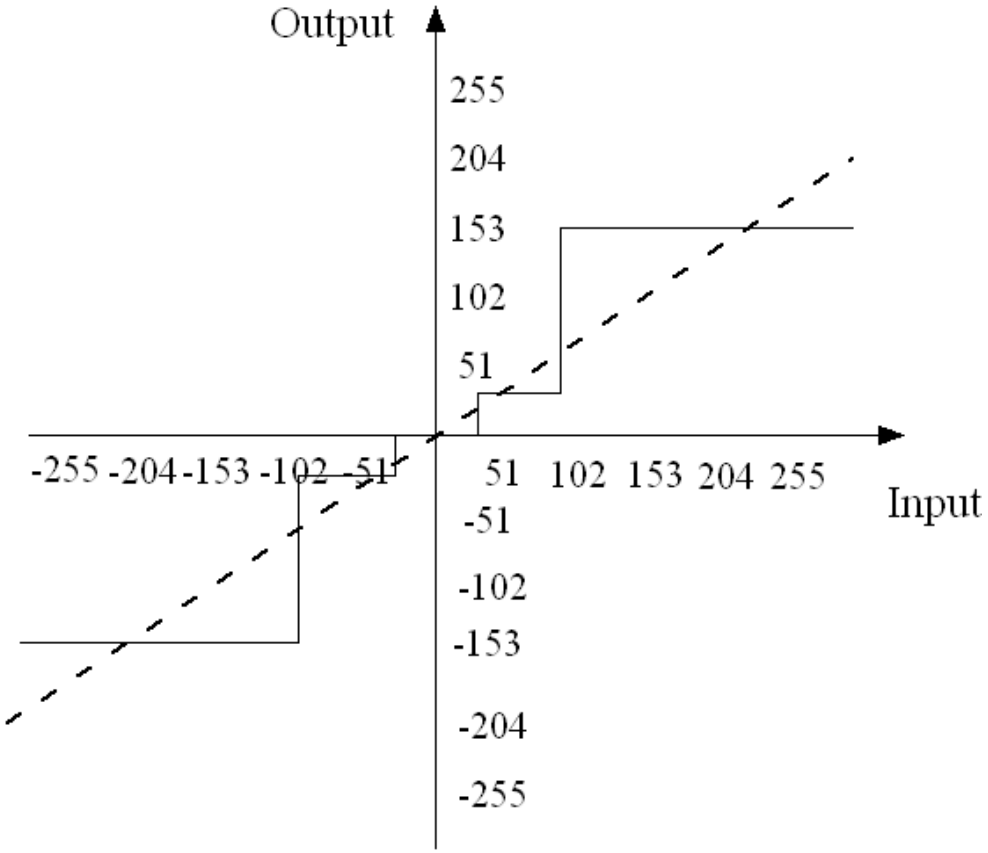
2.6 Introduction to Image Quantization

- Non-Linear Quantizer Applied to a Difference Image

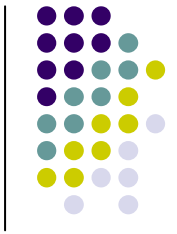


2.6 Introduction to Image Quantization

- Non-Linear Quantizer Transfer Function



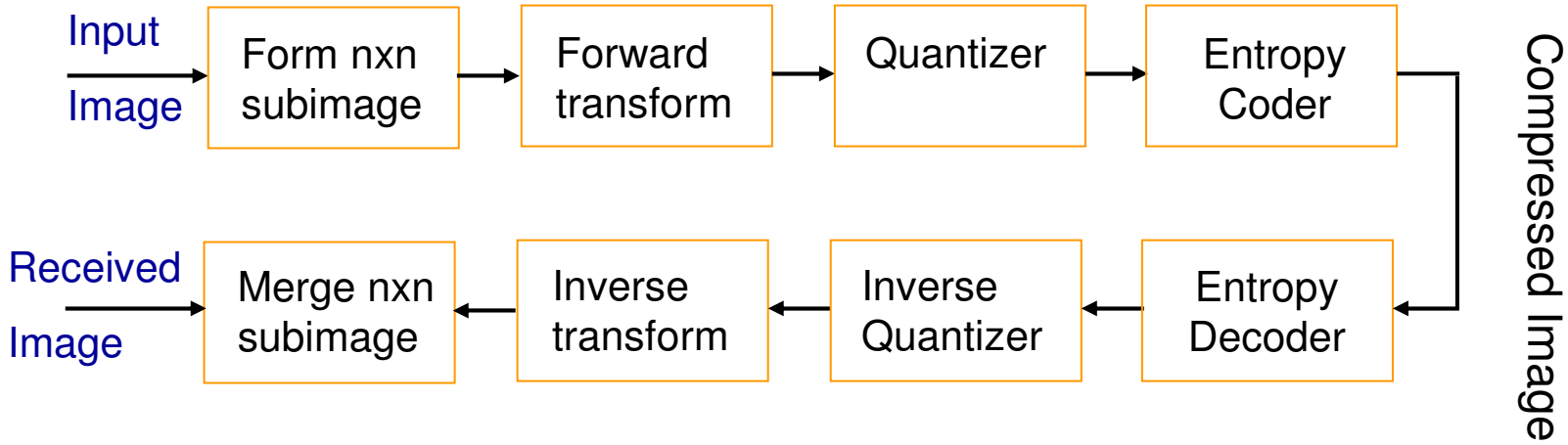
2.6 Introduction to Image Quantization



- Quantization and HVS
  - In most of the codecs, the quantization stepsize is controlled in such a way that distortion is as imperceptible as possible
  - Much study has been conducted for the characteristics of the HVS (human visual system). It is well known that various features in video/image tend to mask the visibility of errors introduced by coding. These include: Edges, Motion and high luminance values
  - Codec try to focus quantization errors in these regions.

2.7 Transform Coding
--- Basic Transform coding

- Block Based Transform Coding
  - Transform coding is performed by taking an image and breaking it down into sub-image (block) of size nxn. The transform is then applied to each sub-image (block) and the resulting transform coefficients are quantized and entropy coded.



2.7 Transform Coding
--- Basic Transform coding

- Consider the following block of data

$$[F] = \begin{bmatrix} 5 & 6 & 8 & 10 \\ 6 & 6 & 5 & 7 \\ 4 & 5 & 3 & 6 \\ 8 & 7 & 5 & 5 \end{bmatrix}$$

- A 2-Dimensional transform can be carried out in a separable way (i.e. first down the columns and then along the rows).

2.7 Transform Coding
--- Basic Transform coding

- The 1-Dimensional transform is calculated according to [C']=[T][F];
- Where [T] is the transform matrix

$$[C'] = 1/2 \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 5 & 6 & 8 & 10 \\ 6 & 6 & 5 & 7 \\ 4 & 5 & 3 & 6 \\ 8 & 7 & 5 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 11.5 & 12.0 & 10.5 & 14.0 \\ -0.5 & 0.0 & 2.5 & 3.0 \\ 1.5 & 1.0 & 2.5 & 1.0 \\ -2.5 & -1.0 & 0.5 & 2.0 \end{bmatrix}$$

4 Coefficients contain 94% of total energy

2.7 Transform Coding
--- Basic Transform coding

- A 2-dimentional transform can be extended according to :[C] = [C'][T]^T =[T] [F][T]^T

$$[C]=1/2\begin{bmatrix} 11.5 & 12.0 & 10.5 & 14.0 \\ -0.5 & 0.0 & 2.5 & 3.0 \\ 1.5 & 1.0 & 2.5 & 1.0 \\ -2.5 & -1.0 & 0.5 & 2.0 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

$$=\begin{bmatrix} 24.0 & -0.5 & 1.5 & -2.0 \\ 2.5 & -3.0 & 0.0 & -0.5 \\ 3.0 & -0.5 & -0.5 & 1.0 \\ -0.5 & -3.0 & 0.0 & -1.5 \end{bmatrix}$$

93% of energy now in one term that is in position (0,0)

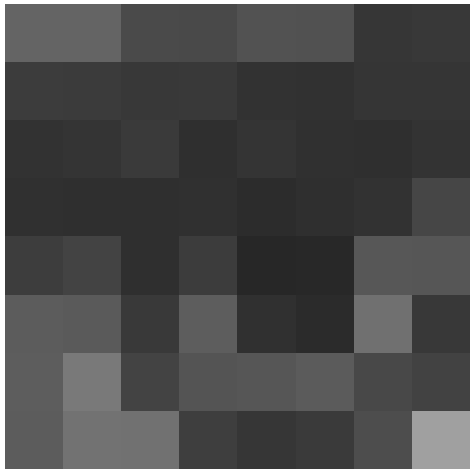
2.7 Transform Coding
--- Basic Transform coding

- Discrete Cosine Transform
  - For a 2-D input block U, the transform coefficients can be found as $Y = CUC^T$
  - The inverse transform can be found as $U = C^T YC$
  - The NxN discrete cosine transform matrix C=c(k,n) is defined as:

$$c(k,n) = \begin{cases} \dfrac{1}{\sqrt{N}} & for\ k = 0\ and\ 0 \le n \le N-1, \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N} & for\ 1 \le k \le N-1\ and\ 0 \le n \le N-1. \end{cases}$$

2.7 Transform Coding
--- Basic Transform coding

- 8x8 2-D DCT

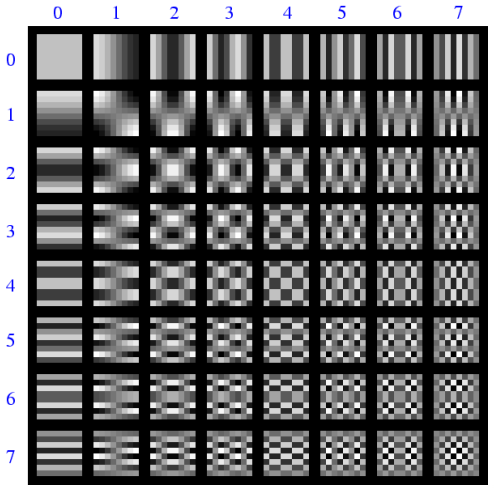


Original 8x8
pixel image

8x8 2-D DCT

8x8 2-D DCT
coefficients

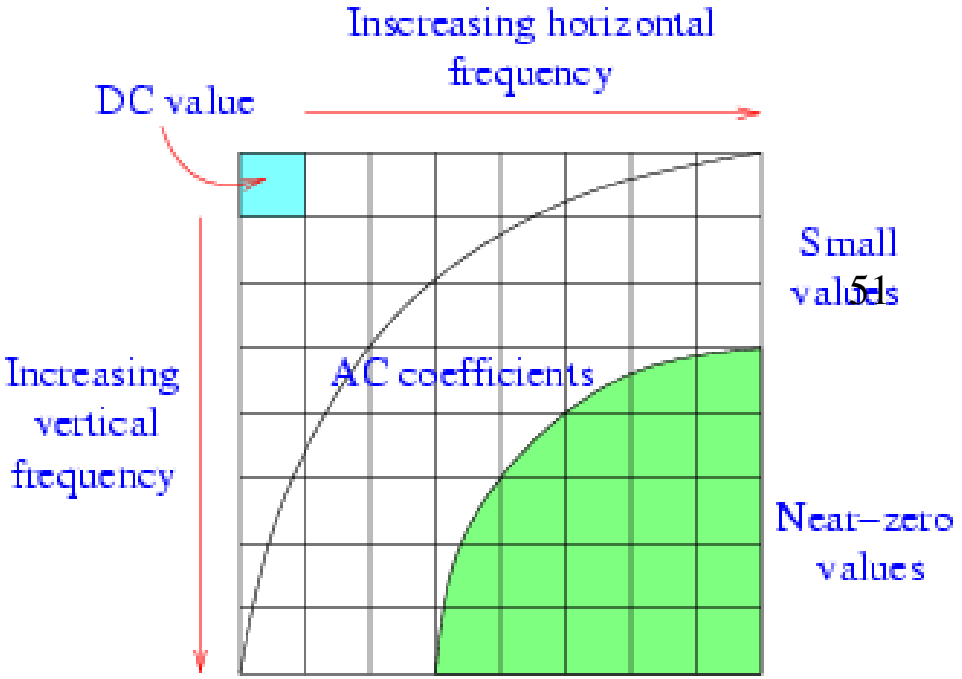
543.87	22.53	50.40	−21.20	−0.62	−4.91	−22.37	−12.54
−76.02	8.84	−40.14	30.71	4.25	2.03	15.01	6.62
83.64	32.46	5.76	−2.03	−2.68	−34.57	20.62	−2.62
32.91	23.27	−10.25	34.64	−4.21	29.63	−39.55	−11.59
15.62	−17.05	30.56	−27.60	8.62	−19.00	21.84	3.71
12.27	20.77	−31.24	36.16	−2.39	−17.86	2.94	−10.00
11.49	5.52	11.37	−12.96	−0.92	14.98	−3.26	9.73
7.64	1.15	−5.81	9.25	−3.84	−15.97	2.03	−1.61



2.7 Transform Coding
--- Basic Transform coding

- The distribution of 2-D DCT Coefficients

Ref: H. Wu



68	3	5	–2	0	0	–2	0
–10	0	–4	3	0	0	0	0
9	3	0	0	0	–2	0	0
3	2	0	3	0	2	–2	0
0	0	2	–2	0	0	0	0
0	2	–2	2	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Zig-Zag scan

2.7 Transform Coding --- Optimal Transform

- An Optimal transform should achieve:
  - Completely de-correlate the data
  - Maximize the amount of energy packed into the lowest order coefficients
  - Given a 1-D general data column vector X with the mean vector, the covariance matrix (2D) is defined as:

$$COV(\mathbf{X}) = E\left[\left(\mathbf{X}-\bar{\mathbf{X}}\right)\left(\mathbf{X}-\bar{\mathbf{X}}\right)^T\right]$$

  - Where E(•) is the expectation operator. Given an orthogonal transform represented by the matrix [T] and the corresponding transform coeff. Vector by [Y]=[T][X], we have:

$$COV(Y) = E[(TX - \overline{XT})(XT - \overline{XT})^T] = TCOV(X)T^T$$

2.7 Transform Coding
--- Transform Efficiency

- De-correlation and energy packing effects (Eg. DCT)
  - The transform domain covariance matrix COV($\mathbf{Y}_{DCT}$) is generated by using the 1-D 4-point discrete cosine transform (DCT)

$$COV(\mathbf{Y}_{DCT}) = \mathbf{T}_{DCT} COV(\mathbf{X}) \mathbf{T}_{DCT}^{T}$$

$$= \begin{bmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6533 & 0.2706 & -0.2706 & -0.6533 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{bmatrix} \times \begin{bmatrix} \mathbf{1.0000} & 0.9100 & 0.8281 & 0.7536 \\ 0.9100 & \mathbf{1.0000} & 0.9100 & 0.8281 \\ 0.8281 & 0.9100 & \mathbf{1.0000} & 0.9100 \\ 0.7536 & 0.8281 & 0.9100 & \mathbf{1.0000} \end{bmatrix}$$

$$\times \begin{bmatrix} 0.5000 & 0.6533 & 0.5000 & 0.2706 \\ 0.5000 & 0.2706 & -0.5000 & -0.6533 \\ 0.5000 & -0.2706 & -0.5000 & 0.6533 \\ 0.5000 & -0.6533 & 0.5000 & -0.2706 \end{bmatrix}$$

$$= \begin{bmatrix} 1.7458 & 1.8241 & 1.8241 & 1.7458 \\ 0.1831 & 0.0779 & -0.0779 & -0.1831 \\ 0.0077 & -0.0860 & -0.0860 & 0.0077 \\ 0.0132 & -0.0366 & 0.0366 & -0.0132 \end{bmatrix} \times \begin{bmatrix} 0.5000 & 0.6533 & 0.5000 & 0.2706 \\ 0.5000 & 0.2706 & -0.5000 & -0.6533 \\ 0.5000 & -0.2706 & -0.5000 & 0.6533 \\ 0.5000 & -0.6533 & 0.5000 & -0.2706 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{3.5699} & 0.0000 & -0.0782 & 0.0000 \\ 0.0000 & \mathbf{0.2814} & 0.0000 & -0.0026 \\ -0.0782 & 0.0000 & \mathbf{0.0937} & 0.0000 \\ 0.0000 & -0.0026 & 0.0000 & \mathbf{0.0550} \end{bmatrix}$$

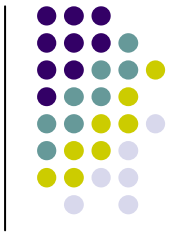
Ref: H. Wu

2.7 Transform Coding
--- Transform Efficiency

- From the previous slides, there are two distinctive effects by the DCT
  - All off-diagonal elements of the transform domain covariance matrix are significantly reduced, indicating that the transform coefficients are effectively de-correlated as a result of the transform;
  - The majority of the data energy (as represented by the variance) has been transferred to a few low-order coefficients after the transform.
- The most desirable transform domain covariance matrix will have a diagonal form (with values of corresponding to the eignvalues of the COV matrix) which achieves 100% decorrelation efficiency for transform coefficients

$$[T][COV(X)][T]^T = \Lambda$$

2.7 Transform Coding
--- Optimal Transform & Efficiency



- The optimal Transform exists – eg. Karhunen-Loeve Transform (KLT), but has some limitations
  - The KLT has the maximum energy packing capability with completely de-correlation for signal in transform domain
  - KLT involves the estimation of the COV, its diagonalisation and the contraction of the basis vectors
  - The basis vectors (transform matrix) are not fixed, and must be generated for images with different correlation characteristics.
  - The transform remains of more theoretical than practical interest as far as image coding is concerned.
  - Due to the lack of a fast implementation, it is not used practically.

2.7 Transform Coding
--- Discrete Cosine Transform

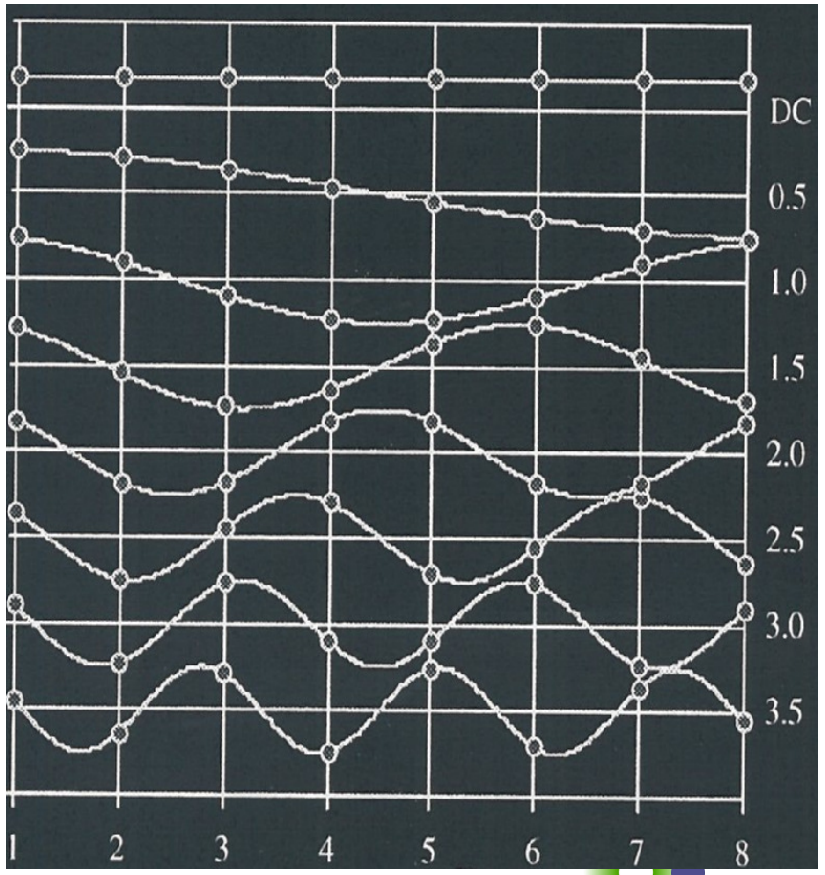
- A transform is calculated by finding the correlation between the input sub-image and a series of basis vectors
- The transform has separable property (2D transform = 1D transform along rows and then down the columns)
- The DCT has the advantage for video coding of good redundancy reduction with performance close to the KLT [K.R.Rao]

$$c(k,n) = \begin{cases} \dfrac{1}{\sqrt{N}} & for\ k = 0\ and\ 0 \le n \le N-1, \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N} & for\ 1 \le k \le N-1\ and\ 0 \le n \le N-1. \end{cases}$$

2.7 Transform Coding
--- Discrete Cosine Transform

- In the case of the DCT, the basis vectors (functions) are a series of harmonically related cosine functions
- Basis Vectors for N=8 DCT

$c(k,n)$	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$
$k=0$	0.35	0.35	0.35	0.35	0.35	0.35	0.35	0.35
$k=1$	0.49	0.42	0.28	0.10	–0.10	–0.28	–0.42	–0.49
$k=2$	0.46	0.19	–0.19	–0.46	–0.46	–0.19	0.19	0.46
$k=3$	0.42	–0.10	–0.49	–0.28	0.28	0.49	0.10	–0.42
$k=4$	0.35	–0.35	–0.35	0.35	0.35	–0.35	–0.35	0.35
$k=5$	0.28	–0.49	0.10	0.42	–0.42	–0.10	0.49	–0.28
$k=6$	0.19	–0.46	0.46	–0.19	–0.19	0.46	–0.46	0.19
$k=7$	0.10	–0.28	0.42	–0.49	0.49	–0.42	0.28	–0.10



2.8 Still Image Coding Standard (JPEG)

- Joint Photographic Experts Group.
- ISO/IEC JTC1/SC 29/WG 1: Subcommittee (SC) 29, Working Group (WG).
- Formed in 1986 by ISO and ITU-T (CCITT) and Became an international standard (IS) in 1991.
- General purpose, applicable to almost all continuous-tone still image application
- Lossy (DCT based) vs Lossless (2D DPCM)
- 8/12 bits sample precision up to 65535 lines and 65535 pixels per line
- Huffman vs arithmetic coding
- Sequential, progressive & hierarchical modes
- Compression ratio of 10:1 to 50:1.
- Colour-space independent: up to 255 colour components, components can be sub-sampled and interleaved, YUV is better than RGB

2.8 Still Image Coding Standard (JPEG)

- 3.2.1 JPEG DCT-Based Encoding
- 3.2.2 Coding of DCT Coefficients
- 3.2.3 JPEG DCT- based Decoding

2.8 JPEG DCT-Based Encoding

785	−28	−2	4	−6	3	−3	6
43	−1	−5	9	1	−3	0	7
−5	9	−8	3	−11	10	6	3
−10	−9	−2	21	−7	4	0	−1
7	−9	2	2	−7	−2	3	0
16	−13	−14	7	−5	2	0	0
0	19	−21	−14	8	3	4	1
18	−5	−9	−11	13	−3	3	−2

98	103	89	93	81	74	79	102
91	96	103	111	99	96	86	95
99	95	82	75	80	102	100	84
104	97	104	106	93	87	80	87
104	103	110	104	97	96	92	81
108	110	110	106	106	105	103	115
111	110	110	104	101	91	87	97
110	106	95	104	115	111	106	105

DCT

39	2	0	0	0	0	0	0
3	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
−1	0	0	1	0	0	0	0
0	0	0	0	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

Quantiser

Zigzag

(0, -2)
(0, 3)
(5, 1)
(0, -1)
(10,1)
(3, 1)
EOB

Entropy Coding

0101
0111
1111010101
000
1111110101
1110101
1010

Quantization Tables

Huffman Tables

User

Ref: H. Wu

2.8 JPEG DCT-Based Encoding

- Recommended JPEG quantisation matrix $N_{j,k}$ =

For luminance:

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

For chrominance:

$$\begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

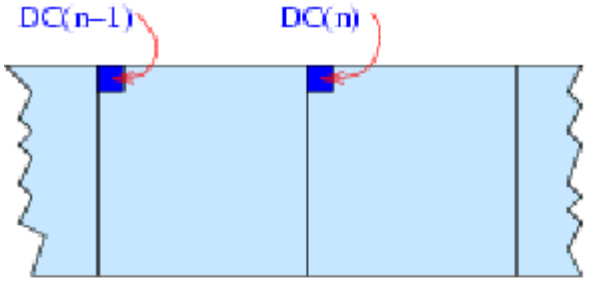
- Quantisation of coefficients $Y_{j,k}$ with a quality factor Q_s

$$\hat{Y}_{j,k} = Q[Y_{j,k}] = round\left(\frac{Y_{j,k}}{N_{j,k} \times Q_s}\right)$$

- High-frequency coefficients and colour components can be quantised more.

2.8 Coding of DCT Coefficients (DC)

- DC coefficient is coded differentially as (size, amplitude). There are 12 categories of size



$$\Delta DC_i = DC_i - DC_{i-1}$$

-3 36 39

Size: 2

-3	00
-2	01
2	10
3	11

Final code:01100

[Coeff]	Size	[Code]	Length
0	0	00	2+0
1	1	010	3+1
2..3	2	011	3+2
4..7	3	100	3+3
8...15	4	101	3+4
16..31	5	110	3+5
32..63	6	1110	4+6
64..127	7	11110	5+7
128..255	8	111110	6+8
256..511	9	1111110	7+9
512..1023	10	11111110	8+10
1024..2047	11	111111110	9+11

2.8 Coding of DCT Coefficients (DC)

Huffman Coding of DC Coefficients

SSSS	DIFF Values	Additional Bits
0	0	–
1	–1,1	0,1
2	–3,–2,2,3	00,01,10,11
3	–7,…,–4,4,…,7	000,…,011,100,..,111
4	–15,…,–8,8,…,15	0000,..,0111,1000,…,1111
5	–31,…,–16,16,…,31	00000,…,01111,10000,…,11111
6	–63,…–32,32,…63	…,…
7	–127,…,–64,64,…,127	…,…
8	–255,…,–128,128,…,255	…,…
9	–511,…,–256,256,…,511	…,…
10	–1023,…,–512,512,…,1023	…,…
11	–2047,…,–1024,1024,…,2047	…,…

Huffman Table for Luminance DC Coefficient Differences

Category	Code Length	Codeword
0	2	00
1	3	010
2	3	011
3	3	100
4	3	101
5	3	110
6	4	1110
7	5	11110
8	6	111110
9	7	1111110
10	8	11111110
11	9	111111110

2.8 Coding of DCT Coefficients (AC)

- AC coefficients are re-arranged to a sequence of **(run, level)** pairs through a zigzag scanning process
- Level is further divided into **(Size Categories, Amplitude).**
- Run and size are then combined and coded as a single event (2D VLC)
  - An 8-bit code 'RRRRSSSS' is used to represent the nonzero coefficients
    - The SSSS is defined as size categories from 1 to 11
    - The RRRR is defined as run-length of zeros in the zig-zag scan or number of zeros before a nonzero coefficient
    - The composite value of RRRRSSSS is then Huffman coded

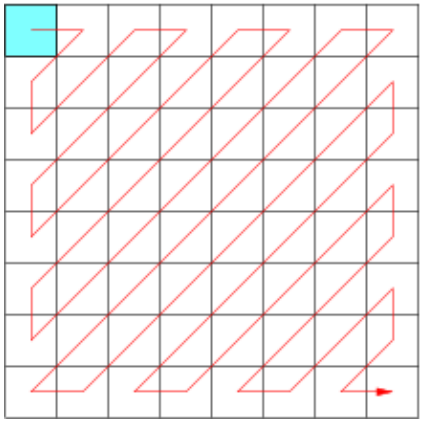
Ex: 1) RRRRSSSS=11110000 represents 15 run '0' coef. and followed by a '0' coef.

2) Multiple symbols used for run-length of '0' coef. exceeds 15

3) RRRRSSSS=00000000 represents end-of-block (EOB)

2.8 Coding of DCT Coefficients (AC)

Zig-Zag scan



SSSS

RRRR	0	1	2	...	9	10	11
0	EOB						
.	N/A						
.	N/A			Composite values			
15	N/A						
	ZRL						

Zero Run	Category	Code length	Codeword
0	1	2	00
0	2	2	01
0	3	3	100
0	4	4	1011
0	5	5	11010
0	6	6	111000
0	7	7	1111000
.	.	.	.
.	.	.	
1	1	4	1100
1	2	6	111001
1	3	7	1111001
1	4	9	111110110
.	.	.	.
.	.	.	.
2	1	5	11011
2	2	8	11111000
.	.	.	.
.	.	.	.
3	1	6	111010
3	2	9	111110111
.	.	.	.
.	.	.	.
4	1	6	111011
5	1	7	1111010
6	1	7	1111011
7	1	8	11111001
8	1	8	11111010
9	1	9	111111000
10	1	9	111111001
11	1	9	111111010
.	.	.	.
.	.	.	.
EOB		4	1010

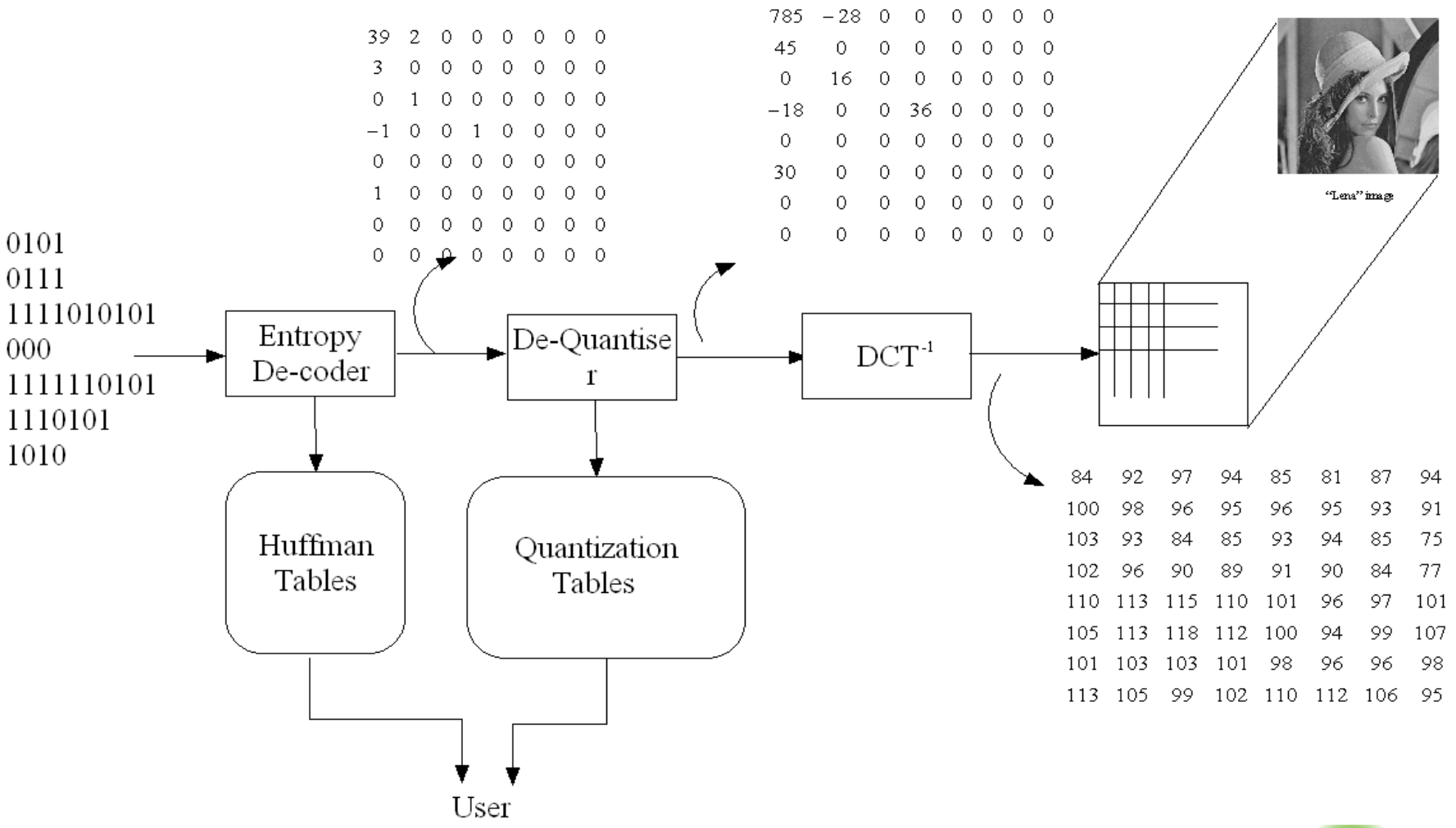
2.8 Coding of DCT Coefficients (AC)

1. Code the amplitude (sign and values) of the nonzero AC coefficient with VLC that is similar to the DC code table.
2. The Huffman tables for AC coef. are defined in Annex K of the JPEG standard
3. Support for arithmetic coding is included (but is patented).
4. Arithmetic coding can give a 8.8%-17.4% improvement in compression over Huffman coding (mean of entropy coding) with fixed table

Huffman Coding for AC Coefficients

Category (SSSS)	AC Coefficient Range
1	–1,1
2	–3,–2,2,3
3	–7,…,–4,4,…,7
4	–15,…,–8,8,…,15
5	–31,…,–16,16,…,31
6	–63,…,–32,32,…,63
7	–127,…,–64,.64,…,127
8	–255,…,–128,128,…,255
9	–511,…,–256,256,…,511
10	–1023,.,–512,512,…,1023
11	–2047,…,–1024,1024,…,2047

2.9 JPEG DCT-Based Decoding



2.9 JPEG DCT-Based Decoding

- Lossless JPEG
  - Pixel based prediction
  - Prediction error entropy coded
  - Compression of around 2:1
- Progressive JPEG
  - Gradual build-up of an Image
    - Spectral selection
      - A subset of coefficients is transmitted during each pass for all blocks. One usually starts with the low frequency coef. and moves towards to the high freq.
    - Successive approximation
      - The most selected coef. are coded and transmitted during the first pass. In each subsequent pass, one extra bit is coded and transmitted to increase the precision of the coef.