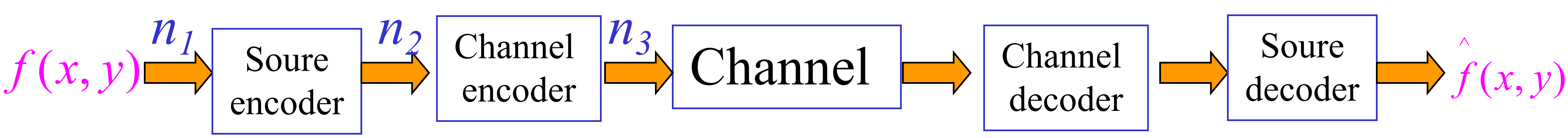
Chapter7 Image Compression

- Preview
- 7.1 Introduction
- 7.2 Fundamental concepts and theories
- 7.3 Entropy coding
- 7.4 Binary image coding
- 7.5 Predictive coding
- 7.6 Transform coding
- 7.7 Introduction to international standards

Preview

General communication model



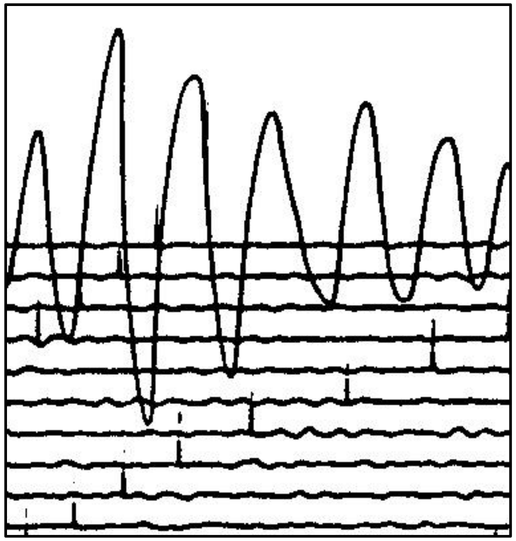
$$n_1 > n_2$$

$$n_3 > n_2$$

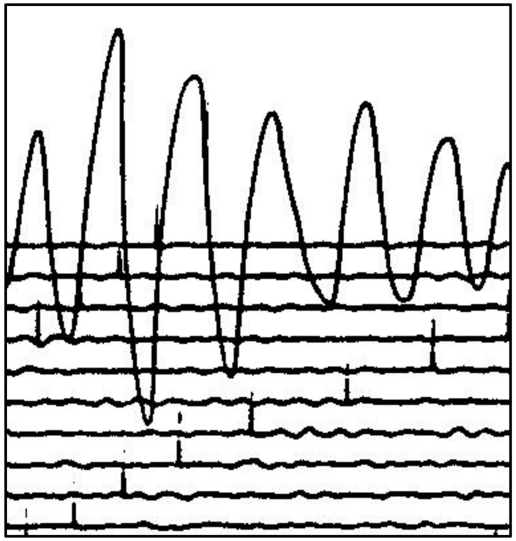
$$n_1 > n_3 > n_2$$

Preview

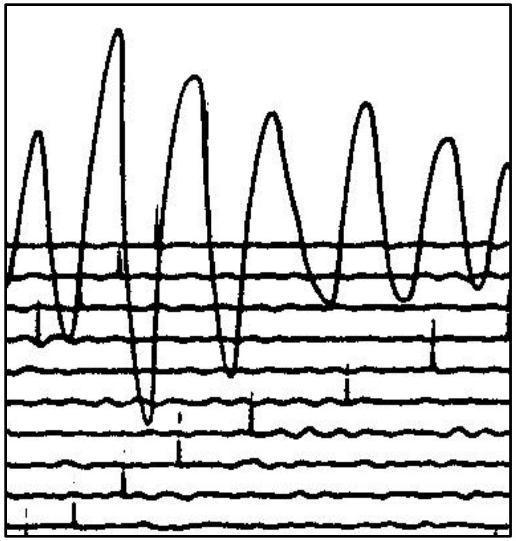
Lossless compression



18.7Kbyte



3.46Kbyte
winzip



2.62Kbyte
G4

Preview

Lossy compression

192kbyte

11.2Kbyte
PSNR=36.97dB

2.5kbyte
PSNR=25.11dB

7.1 Introduction

7.1.1 why code data?

- To reduce storage volume
- To reduce transmission time
  - One colour image
    - 12,000,000pixels
    - 3 channels, each 8 bits
    - 36 Mbyte
  - Video data
    - same resolution
    - 25 frames per second
    - 900 Mbyte/second

7.1 Introduction

7.1.2 Classification

- Lossless compression: reversible
- Lossy compression: irreversible

7.1.3 precondition： redundancy

- coding redundancy
- interpixel redundancy
- psychovisual redundancy

7.1 Introduction

7.1.3 precondition： redundancy

•coding redundancy

If r_k represents the gray levels of an image and each r_k occurs with probability $p_r(r_k)$, then

$$p_r(r_k) = \frac{n_k}{n}, k = 0,1,\cdots L-1$$

where L is the number of gray levels, n_k is number of times that the kth gray level appears in the image, and n is the total number of pixels in the image

7.1 Introduction

7.1.3 precondition： redundancy •coding redundancy

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

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

For example

r_k	$P_r(r_k)$	*Code1*	$l_1(r_k)$	*Code2*	$l_2(r_k)$
r_0	0.19	000	3	11	2
r_1	0.25	001	3	01	2
r_2	0.21	010	3	10	2
r_3	0.16	011	3	001	3
r_4	0.08	100	3	0001	4
r_5	0.06	101	3	00001	5
R_6	0.03	110	3	000001	6
r_7	0.02	111	3	000000	7

7.1 Introduction

7.1.3 precondition： redundancy

•coding redundancy

$$L_{avg} = \sum_{k=0}^{7} l_2(r_k) p_r(r_k)$$

$$= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16)$$

$$+ 4(0.08) + 5(0.06) + 6(0.03) + 6(0.02)$$

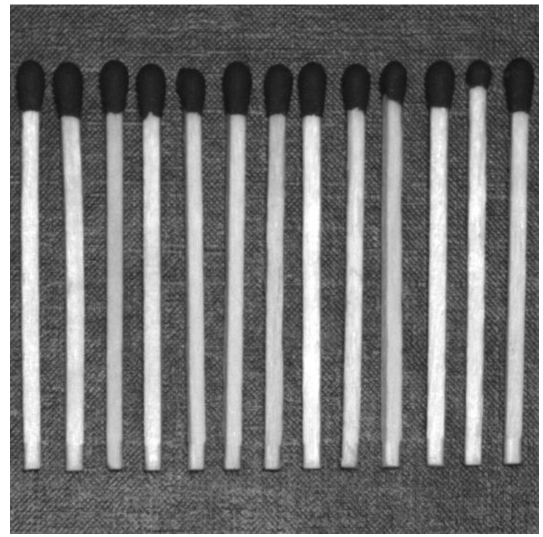
$$= 2.7 bits$$

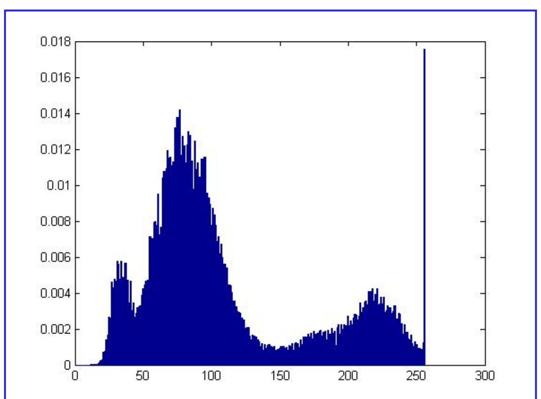
$$\Rightarrow \quad C_R = 3/2.7 = 1.11$$

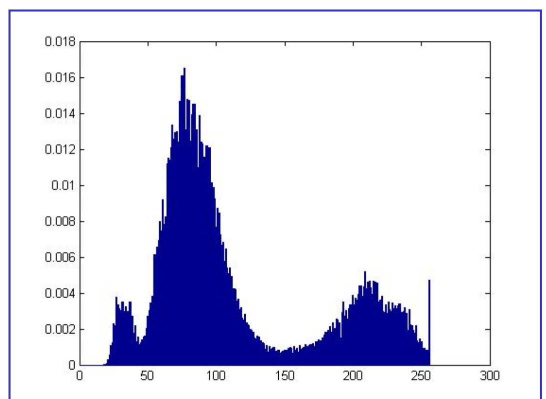
7.1 Introduction

7.1.3 precondition： redundancy

- interpixel redundancy







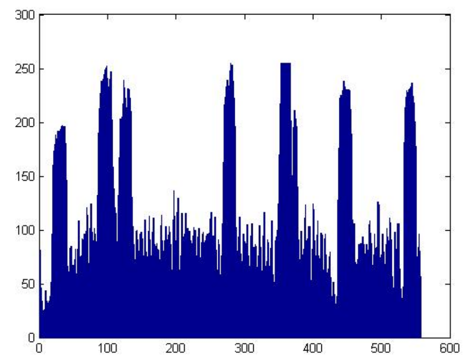
7.1 Introduction

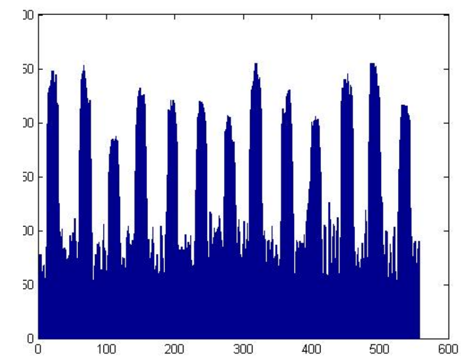
7.1.3 precondition： redundancy • interpixel redundancy

Autocorrelation coefficients

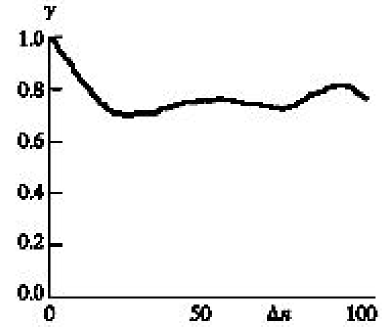
$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}$$

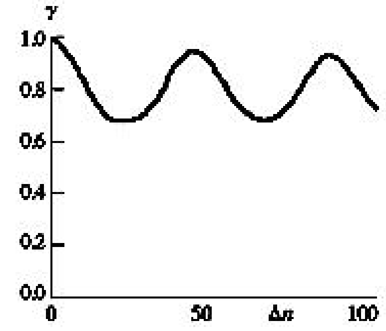
where





$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x,y) f(x, y + \Delta n)$$





7.1 Introduction

7.1.3 precondition： redundancy

- psychovisual redundancy

Psychovisual redundancy is associated with real or quantifiable visual information. The information itself is not essential for normal visual processing

8bits/pixel

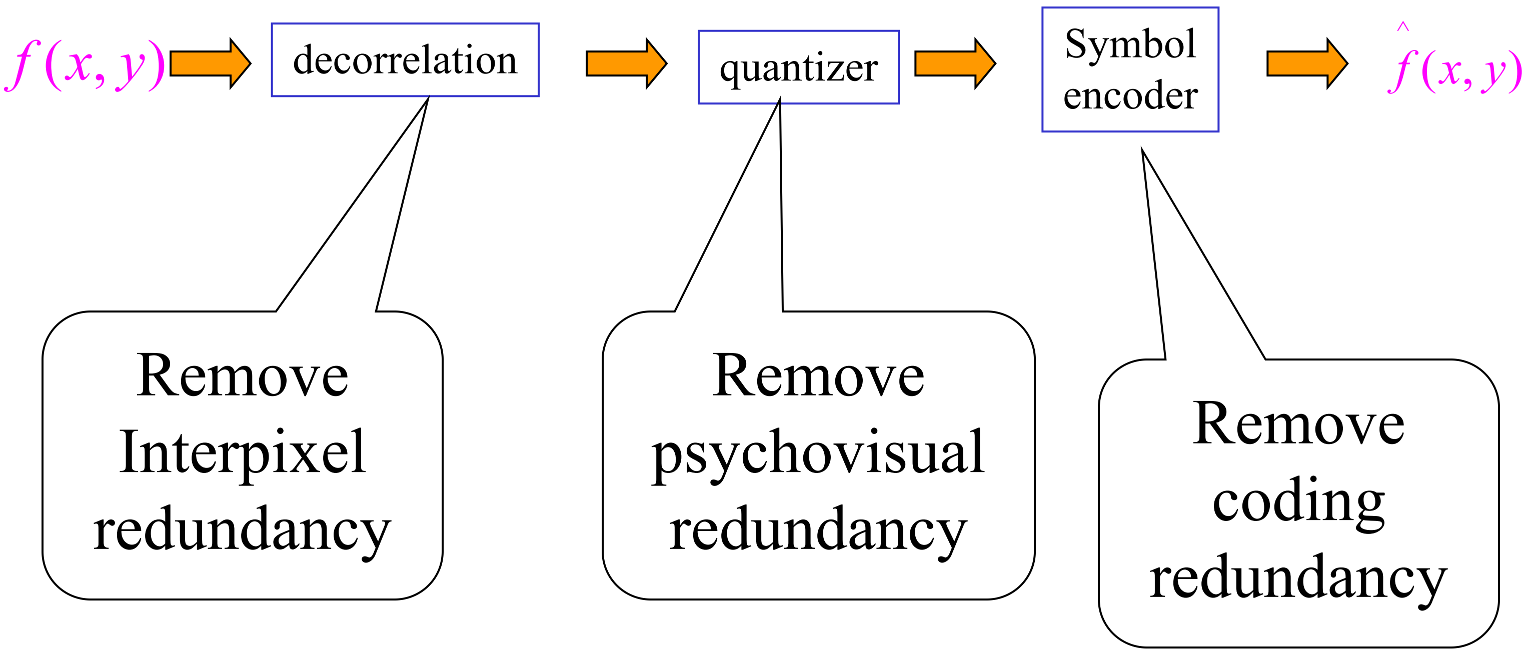
4bits/pixel

2bits/pixel

7.1 Introduction

7.1.3 precondition： redundancy

encoder model



7.1 Introduction

7.1.4 the path of removing redundancy

- Decorrelation + entropy coding
- Context_based model +coding: run length coding, dictionary_based coding (arj(DOS), zip(Windows), compress(UNIX))

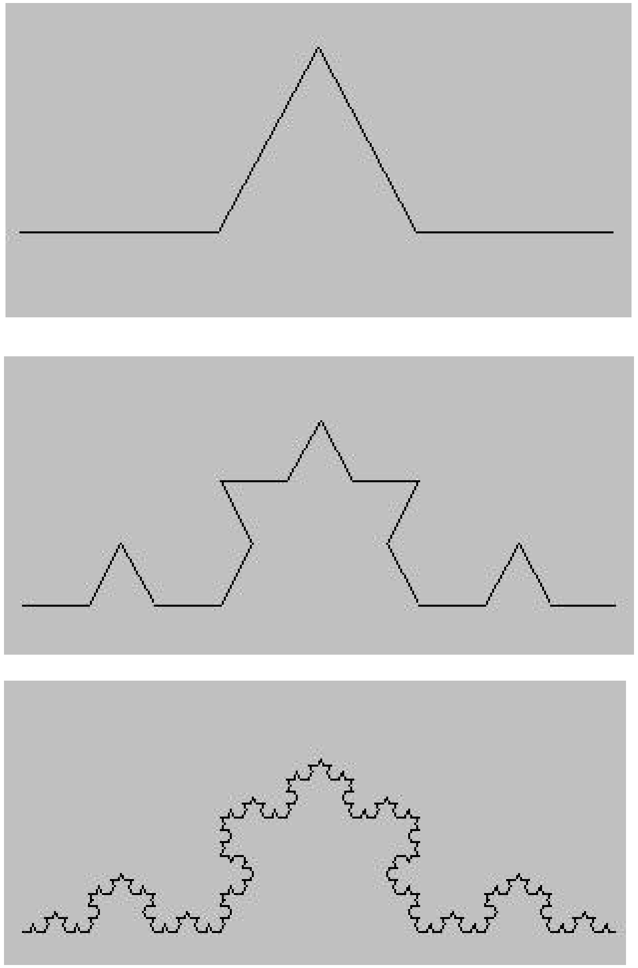
7.1.5 Decorrelation

- Prediction: Linear prediction, Non-linear prediction
- Transformation: DCT, KLT, Walsh-T, WT(wavelet)
- Others: Vector quatization、Fractal、 the second generation coding 、Inpating_based compression

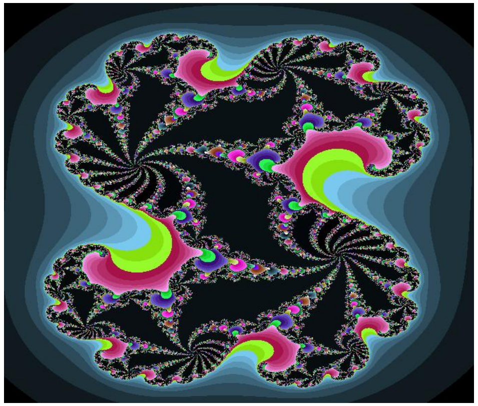
7.1 Introduction

7.1.5 Decorrelation

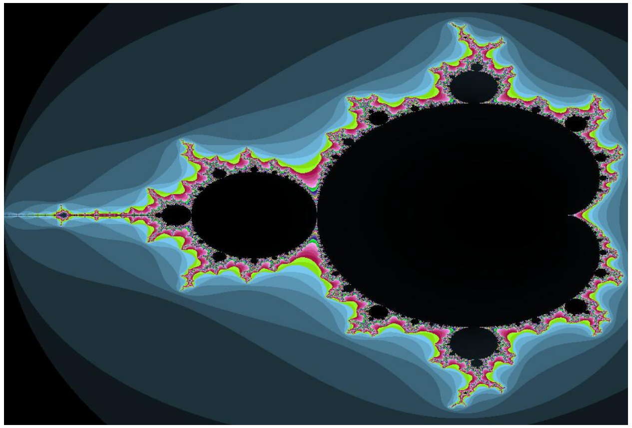
- Fractal



Koch curve



Julia Set



Mandelbrot Set

7.1 Introduction

7.1.5 Decorrelation

- the second generation coding

7.1 Introduction

7.1.6 Entropy coding

- Haffmam coding
- Arithmetic coding

7.1.7 Compression ratio

$$C_R = \frac{n_2}{n_1} \qquad \bar{L} = \frac{n_2}{M \times N} bits/pixel, \quad C_R = \frac{8bits/pixel}{\bar{L}}$$

7.1 Introduction

7.1.8 Fidelity criteria: objective

Let $f(x,y)$ represent an input image and $\hat{f}(x,y)$ denote the Decompressed image. For any value x and y, the error $e(x,y)$ bewteen $f(x,y)$ and $\hat{f}(x,y)$ can be defined as

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

So that the *total error* between the two images is

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

7.1 Introduction

7.1.8 Fidelity criteria: objective

The *root-mean-square error*, e_{ems}, between $f(x,y)$ and $\hat{f}(x,y)$ is

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

The *mean-square signal noise ratio* is defined as

$$SNR_{rms} = 10\lg\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2}$$

7.1 Introduction

7.1.8 Fidelity criteria: objective

The *signal noise ratio* is defined as

$$SNR = 10\lg \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\hat{f}(x,y) - \bar{f}(x,y)\right]^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2}$$

The *peak signal noise ratio* is defined as

$$PSNR = 10\lg \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f_{\max}{}^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2} = 10\lg \frac{f_{\max}{}^2}{e_{rms}^2}$$

7.1 Introduction

7.1.8 Fidelity criteria: subjective

Side-by-side comparisons

TABLE 8.3
Rating scale of the Television Allocations Study Organization. (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.

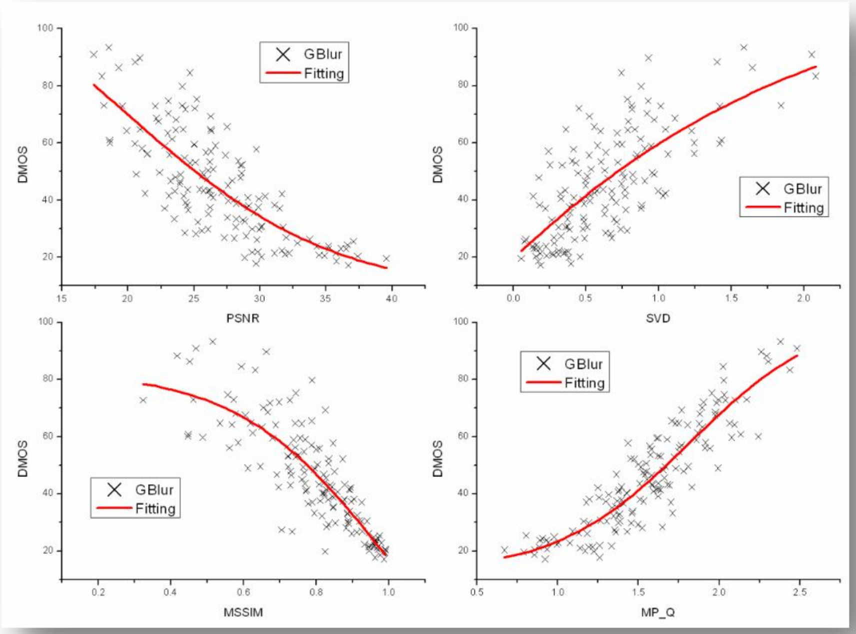
7.1 Introduction

7.1.8 Fidelity criteria: subjective

两幅图象有相同的PSNR==31.7db，同样数量的矩形噪声在左图中加到了平坦的天空区域，而在右图却加在了纹理比较丰富的地面岩石区域，噪声在左图中很易被察觉到，而在右图中确很难被发现（这就是所谓的掩蔽效应），两幅图象主观上质量有很大的差别，而PSNR却不能反映这一区别。

7.1 Introduction

7.1.8 Fidelity criteria: subjective



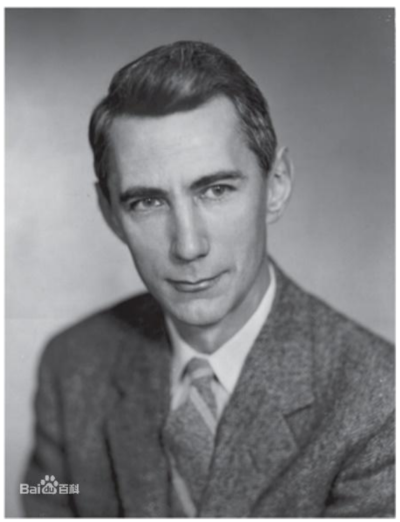
7.1.9 International stander standards

- G3, G4
- JEPG, JPEG2000

7.2 Fundamental concepts and theories

7.2.1 history

1948, Cloude Shannon, A mathematical theory of communication

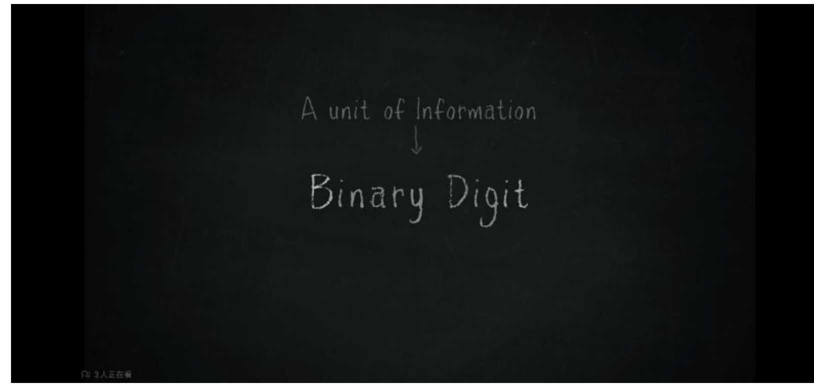


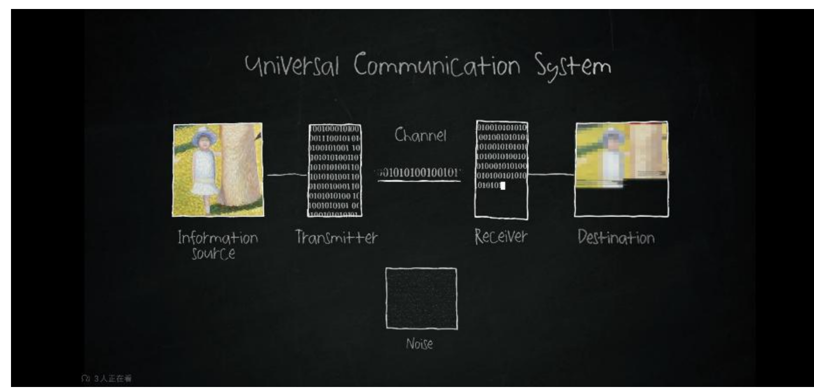
Claude Elwood Shannon
1916-2001

7.2 Fundamental concepts and theories

7.2.1 history

《The Bit Player》





7.2 Fundamental concepts and theories

7.2.2 terms

News: data

information: contents

Information source: symbols

Memoryless source： Entropy of the source

Markov source: Conditional entropy

7.2 Fundamental concepts and theories

7.2.2 Self-information

A random event E that occurs with probability $P(E)$ is said to contain $I(E)$ units of information.

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

For example $P(E) = 0, \quad I(E) = \infty$

$P(E) = 1, \quad I(E) = 0$

Unit: Base=2, bits (binary digits)

Base=e, nats (nature digits)

Base=10, Hartly

7.2 Fundamental concepts and theories

7.2.3 Entropy of the source

definition

Suppose $X=\{x_0, x_1, \ldots x_{N-1}\}$ is a discrete random variable, and the probability of x_j is $P(x_j)$,namely

$$\begin{bmatrix} X \\ P(X) \end{bmatrix} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{N-1} \\ p(x_0) & p(x_1) & \cdots & p(x_1) \end{bmatrix}$$

then

$$H(X) = \sum_{j=0}^{N-1} p(x_j) \log \frac{1}{p(x_j)} = -\sum_{j=0}^{N-1} p(x_j) \log p(x_j)$$

7.2 Fundamental concepts and theories

7.2.3 Entropy of the source

properties

(1) $H(X) \geq 0$

(2) $H(X) = H(1,0,\cdots 0) = H(0,1,\cdots 0) = \cdots = H(0,0,\cdots 1) = 0$

(3) $H(X) = H(x_0, x_1 \cdots x_{N-1}) \leq \log N$

When $P(x_j)=1/N$ for all j,, $H(X)=logN$

example $X=\{1,2,3,4,5,6,7,8\}$, $p_j=1/8$ for each j

$$H(X) = \sum_{j=1}^{8} \frac{1}{8} \log_2 8 = 3bits/symbol$$

7.2 Fundamental concepts and theories

7.2.3 Entropy of the source

definitions

$$H_0(X)=\sum_i p(x_i)\log\frac{1}{p(x_i)}$$

$$H_1(X)=\sum_i p(x_i)\sum_i p(x_i \mid x_{i-1})\log\frac{1}{p(x_i \mid x_{i-1})}$$

$$H_2(X)=\sum_i p(x_i)\sum_i p(x_i \mid x_{i-1}x_{i-2})\log\frac{1}{p(x_i \mid x_{i-1}\, x_{i-2})}$$

$$\vdots$$

$$H_m(X)=\sum_i p(x_i)\sum_i p(x_i \mid x_{i-1}\cdots x_{i-m})\log\frac{1}{p(x_i \mid x_{i-1}\cdots x_{i-m})}$$

property

$$H_\infty=\cdots=H_m<H_{m-1}<\cdots<H_2<H_1<H_0$$

7.2 Fundamental concepts and theories

7.2.3 Noiseless coding theorem

Also called *Shannon first theorem*. Defines the minimum average code word length per source symbol that can be achieved

$$\boxed{H(X) \le L_{avg}}$$

For memoryless (zero-memory) source:

$$H(X) = \sum_i p(x_i) \log \frac{1}{p(x_i)}$$

But, the image data usually are Markov source, how to code?

7.2 Fundamental concepts and theories

7.2.3 Noiseless coding theorem

Coding steps

$$\begin{tikzcd} \text{marcov}\ \boxed{\mathbf{X}} \arrow[r, "\text{decorrelation}"] \arrow[d, "\text{entropy coding}"'] & \boxed{\mathbf{Y}}\ \text{memoryless} \arrow[d, "\text{entropy coding}"'] \\ \boxed{L_X} & \boxed{L_Y} \end{tikzcd}$$

$$H(X)=H_m(X)<H_0(X), \qquad L_X= H_0(X)$$

$$H(Y)=H_0(Y),\ L_Y= H_0(Y)$$

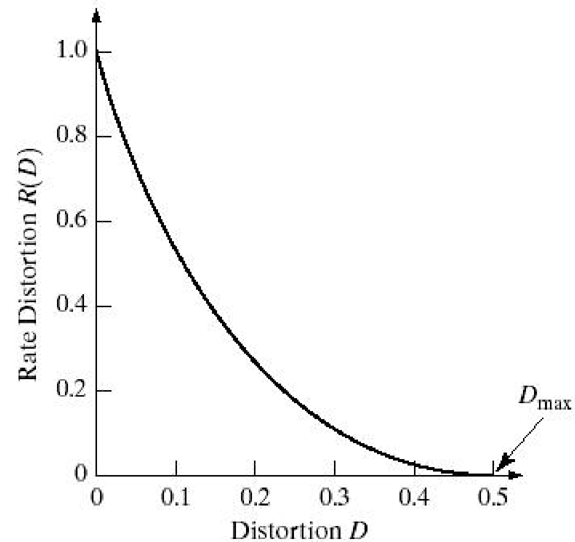
$$H(X)=H(Y) \Rightarrow L_Y= H_m(X)<L_X$$

7.2 Fundamental concepts and theories

7.2.4 Noise coding theorem

For a given rate distortion *D*, the rate distortion function *R(D)* is less than *H(X)*, and *R(D)* is the minimum of coding length L_{avg}

FIGURE 8.10 The rate distortion function for a binary symmetric source.

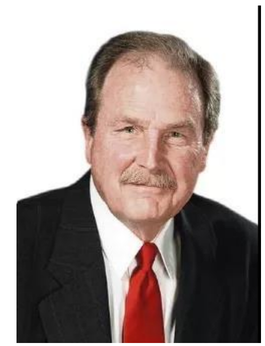


7.3 Entropy Coding

7.3.1 Huffman coding

Huffman, 1952

Robert M. Fano



David A. Huffmam

- code the low probabilities symbols with long word
  code the high probabilities symbols with short word
- smallest possible number of code symbols per source symbol
- the source symbols must be coded *one at a time*

7.3 Entropy Coding

7.3.1 Huffman coding

step1:create a series of source reduction by ordering the probabilities

step2:combine the lowest probability symbols into a single symbol that replaces them in the next source reduction

step3:repeat step2 until the lowest probability is 1

For example

Symbol	Probability	1	2	3	4
A	0.4	0.4	0.4	0.4	0.6
B	0.3	0.3	0.3	0.3	0.4
C	0.1	0.1	0.2	0.3	
D	0.1	0.1	0.1		
E	0.06	0.1			
F	0.04				

7.3 Entropy Coding

7.3.1 Huffman coding

step4:code each reduced source with 0 and 1, staring with the smallest source and working back to the original source

Symbol	**Probability**	**1**	**2**	**3**	**4**
A	**0.4 1**	**0.4 1**	**0.4 1**	**0.4 1**	**0.6 0**
B	**0.3 00**	**0.3 00**	**0.3 00**	**0.3 00**	**0.4 1**
C	**0.1 011**	**0.1 011**	**0.2 010**	**0.3 01**	
D	**0.1 0100**	**0.1 0100**	**0.1 011**		
E	**0.06 01010**	**0.1 0101**			
F	**0.04 01011**				

$$L_{avg} = 0.4\times1+0.3\times2+0.1\times3+0.1\times4+0.06\times5+0.04\times5 = 2.2 bits/symbol$$

$$H_0(A) = -\sum_{i=1}^{6} p(a_i)\log p(a_i) = 2.14 bits/symbol$$

$$L_{avg} > H_0(X)$$

7.3 Entropy Coding

7.3.2 Arithmetic coding

- An entire sequence of source symbols(or message) is assigned as a single arithmetic code word
- the code word itself defines an interval of real numbers between 0 and 1

The encoding process of an arithmetic coding can be explained through the following example

7.3 Entropy Coding

7.3.2 Arithmetic coding: example

encoding

- Let us assume that the source symbols are $\{a_1\, a_2\, a_3 a_4\}$ and the probabilities of these symbols are {0.2, 0.2, 0.4, 0.2}
- To encode a message of a sequence: $a_1\, a_2\, a_3 a_3 a_4$
- The interval [0,1) can be divided as four sub-intervals: [0.0, 0.2), [0.2, 0.4), [0.4, 0.8), [0.8, 1.0),

symbols	probabilites	Initial intervals
a_1	0.2	[0.0,0.2)
a_2	0.2	[0.2,0.4)
a_3	0.4	[0.4,0.8)
a_4	0.2	[0.8,1.0)

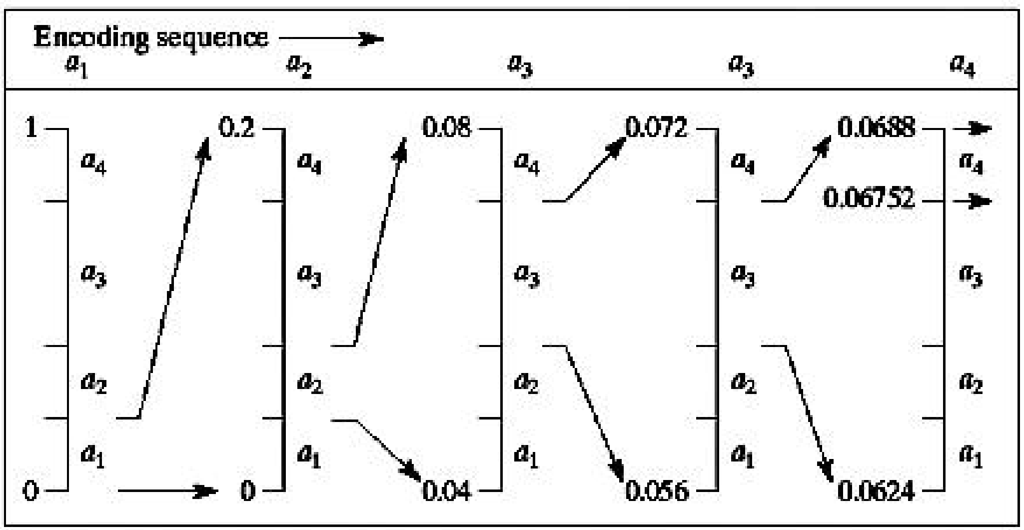
7.3 Entropy Coding

7.3.2 Arithmetic coding: example

•We take the first symbol a_1 from the message and find its encoding range is [0.0, 0.2).

•The second symbol $a2$is encodedby taking the 20th-40th of interval [0.0, 0.2) as the new interval [0.02,0.04).

•And so on. Visually, we can use the figure:



•Finally, choose a number from the interval of [0.0688,0.06752) as the output: 0.06800

7.3 Entropy Coding

7.3.2 Arithmetic coding : example

decoding

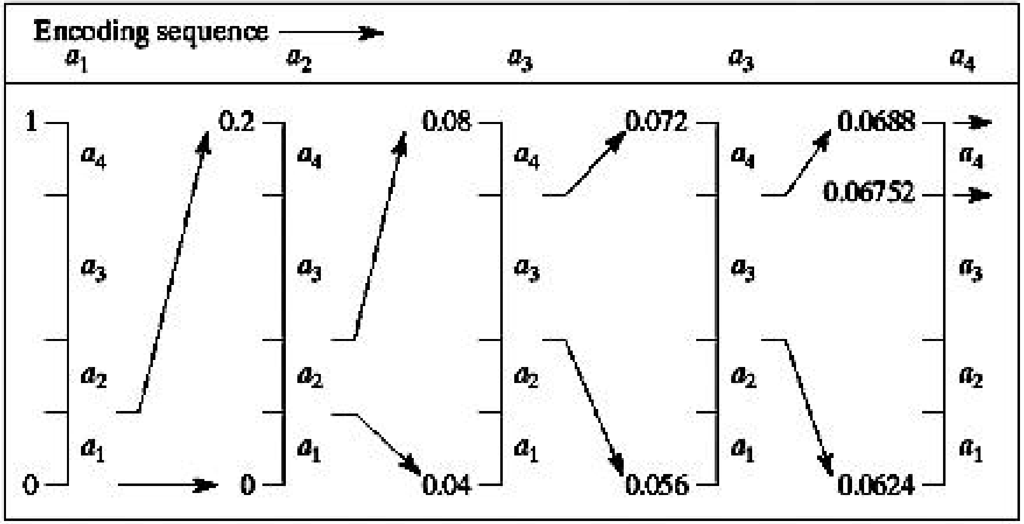
$0 < 0.068800 < 0.2 \Rightarrow a_1$

$0.04 < 0.068800 < 0.08 \Rightarrow a_2$

$0.056 < 0.068800 < 0.072 \Rightarrow a_3$

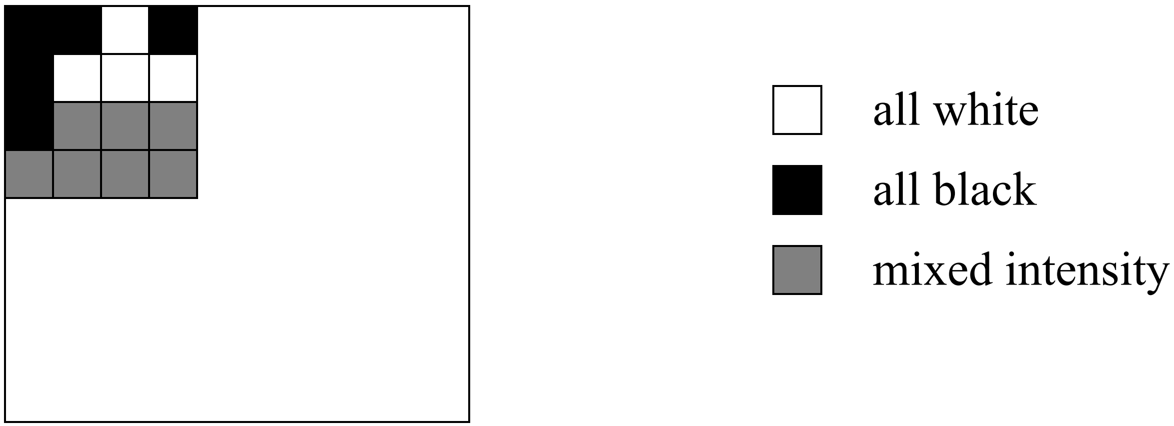
$0.0624 < 0.068800 < 0.0688 \Rightarrow a_3$

$0.06752 < 0.068800 < 0.0688 \Rightarrow a_4$



7.4 Binary Image coding methods

7.4.1 constant area coding



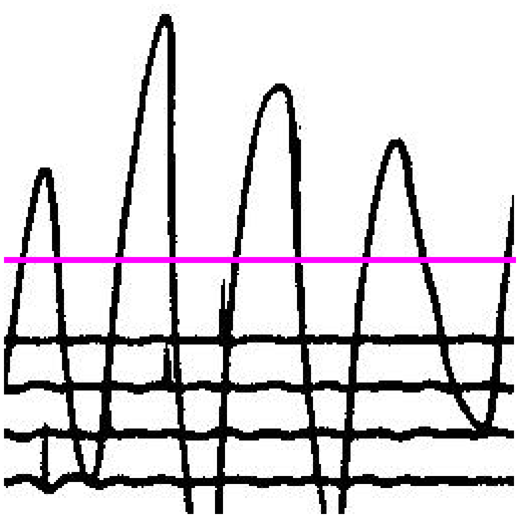
- The most probable category is coded with 1-bit code word 0
- The other two category is coded with 2-bit code word 10 and 11
- The code assigned to the mixed intensity category is used as a prefix

7.4 Binary Image coding methods

7.4.2 run length coding

- Represent each row of a image by a sequence of lengths that describe runs of black and white pixels

- the standard compression approach in facsimile (FAX) coding. CCITT, G3,G4.



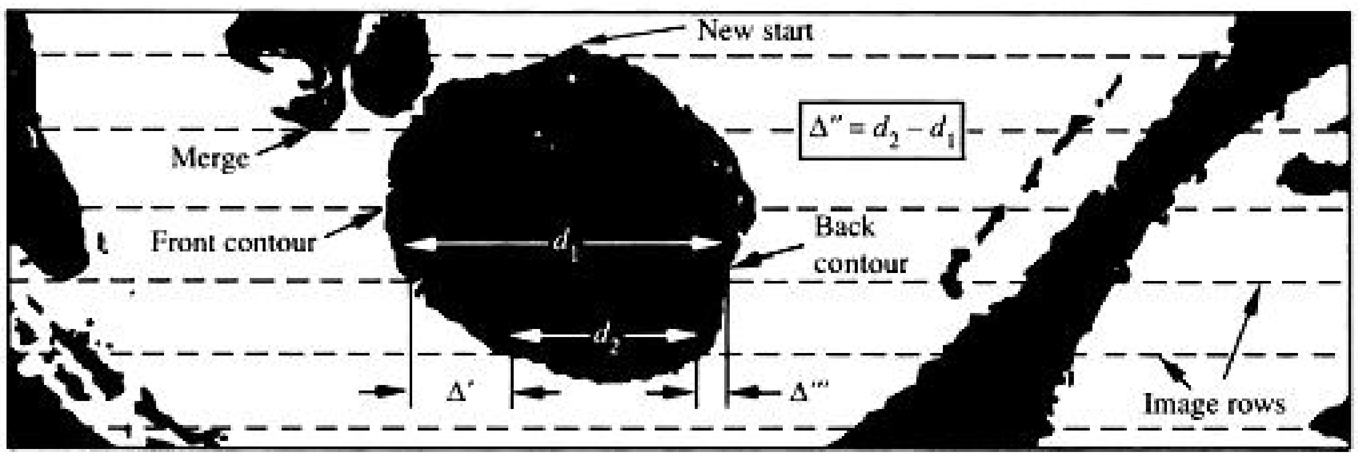
(1,8)	(0,3)	(1,14)	(0,4)	(1, 28)	(0,3)	(1,23)	(0,3)	(1,29)
(0, 3)	(1,28)	(0,3)	(1,31)	(0,4)	(1,25)	(0,4)	(1,38)	(0,4)

8	3	14	4	28	3	23	3	29	3	28
9	3	31	4	25	4	38	4			

7.4 Binary Image coding methods

7.4.3 contour tracing and coding

- Δ' is the difference between the starting coordinates of the front contours adjacent lines
- Δ'' is the difference between the front-to-back contour lengths contours adjacent lines
- code: new start, Δ' and Δ''



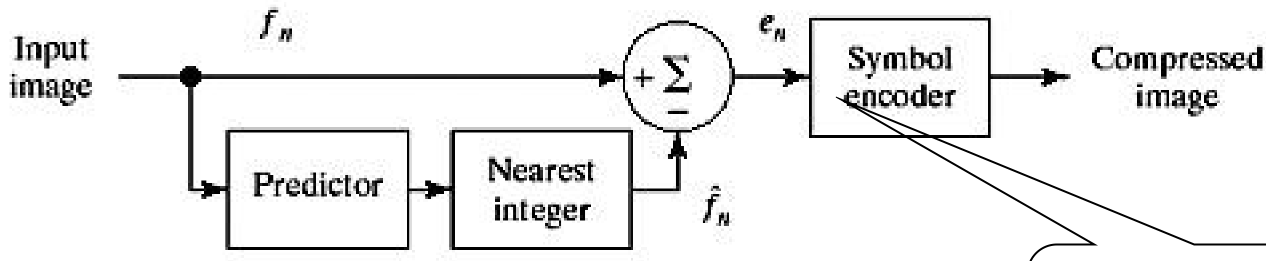
But for gray image…

162	161	159	161	162	160	158	156	156	161
162	161	159	161	162	160	158	156	156	161
163	158	159	159	160	158	155	155	156	158
159	157	159	156	159	159	154	152	155	153
155	157	156	156	158	157	156	155	154	156
156	157	155	151	157	156	155	156	156	154
157	156	156	156	156	156	154	156	155	155
158	157	155	155	156	155	155	155	155	155
156	155	156	153	156	155	156	155	154	156
155	155	157	154	157	155	157	158	158	158

7.5 Predictive coding

7.5.1 lossless coding

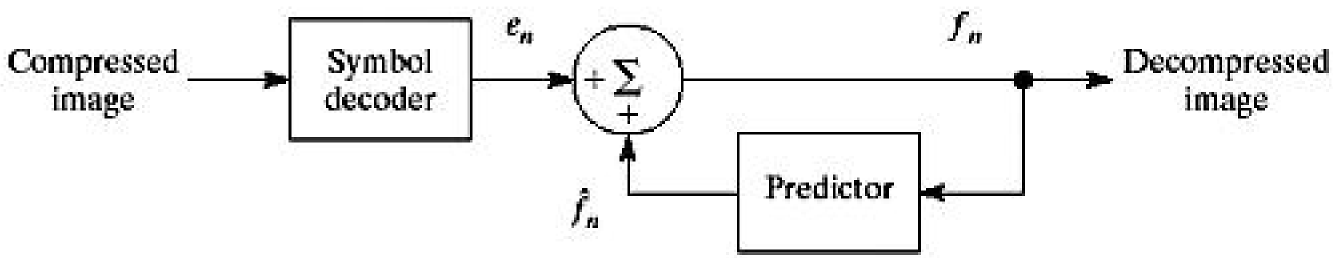
encoder



Huffman coding
or
Arithmetic coding

$$e_n = f_n - \hat{f}_n$$

decoder



$$f_n = e_n + \hat{f}_n$$

7.4 Predictive coding

7.5.1 lossless coding

In most case, the prediction is formed by a linear combination of m previous pixels. That is,

$$\hat{f}_n = round\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right]$$

where m is the order of the linear predictor

α_i are prediction coefficients

For example, 1-D linear predictive can be written

$$\hat{f}(x,y) = round\left[\sum_{i=1}^{m} \alpha_i f(x, y-i)\right]$$

7.4 Predictive coding

7.5.1 lossless coding: Experimental result

m=1,α=1

$$\hat{f}(x,y) = round\left[f(x,y-1)\right]$$

Original image

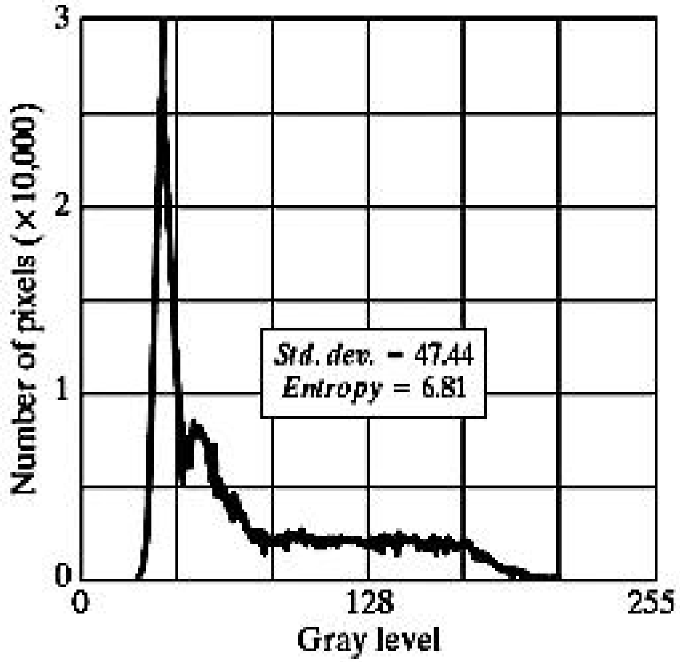
Residual image
(128 represents 0)

7.4 Predictive coding

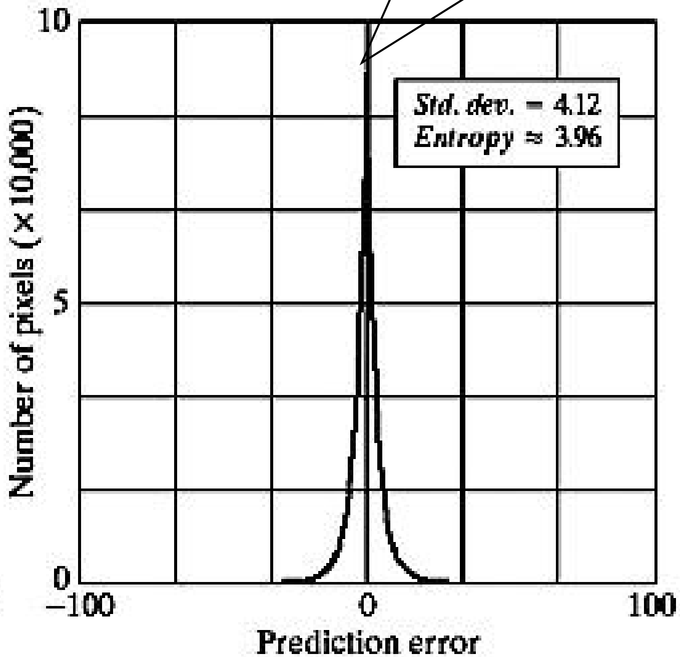
7.5.1 lossless coding: Experimental result

$m=1$, $\alpha=1$

$$p_e(e)=\frac{1}{\sqrt{2}\sigma_e}e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$



Histogram of original image



Histogram of residual image

7.4 Predictive coding

7.5.1 lossless coding: JPEG lossless compression standard

c	*b*
a	*x*

Location of pixels

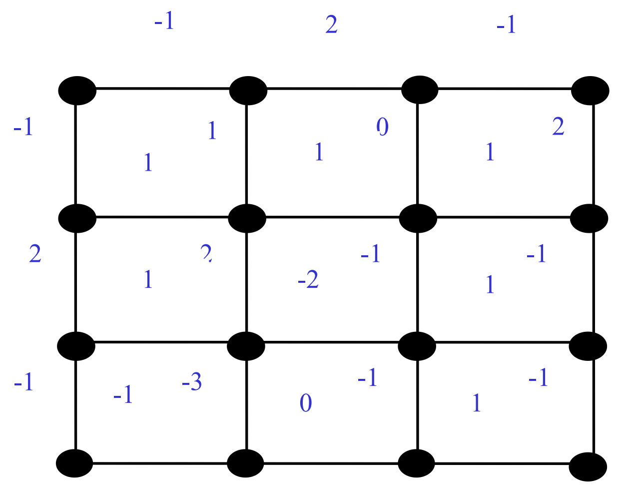
Prediction schemes

Prediction Scheme	Prediction value
0	Null
1	a
2	b
3	c
4	$a+b-c$
5	$a+(b-c)/2$
6	$b+(a-c)/2$
7	$(a+b)/2$

7.4 Predictive coding

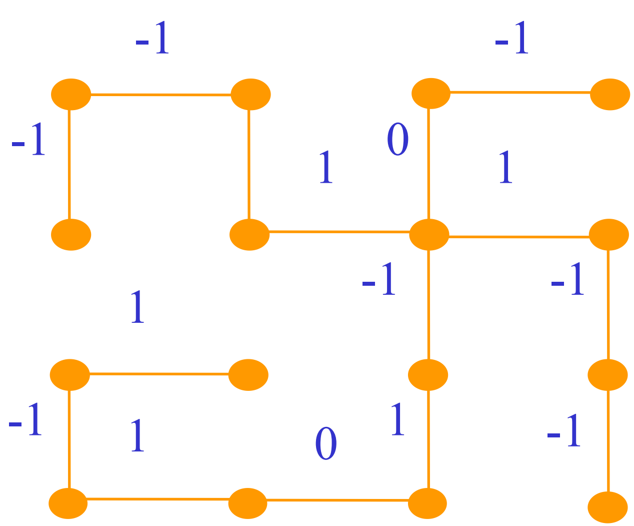
7.5.1 lossless coding: predictive tree

21	20	22	21
20	21	22	23
22	23	21	22
21	20	20	21

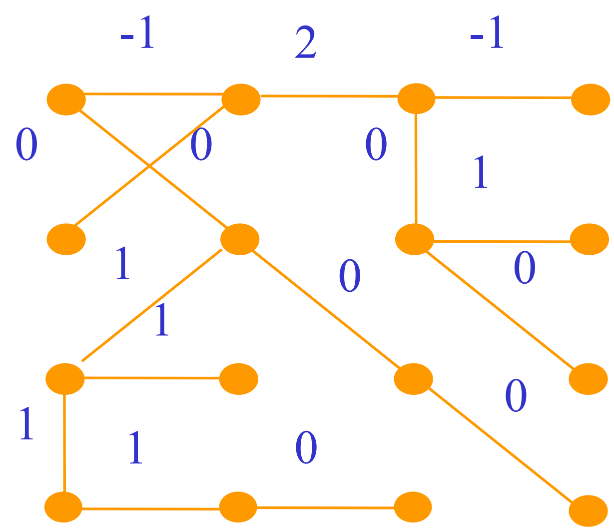


7.4 Predictive coding

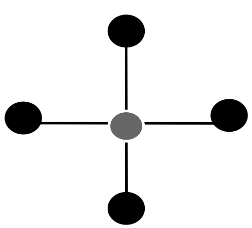
7.5.1 lossless coding: predictive tree

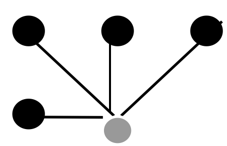


MAW预测树



SNMAW预测树

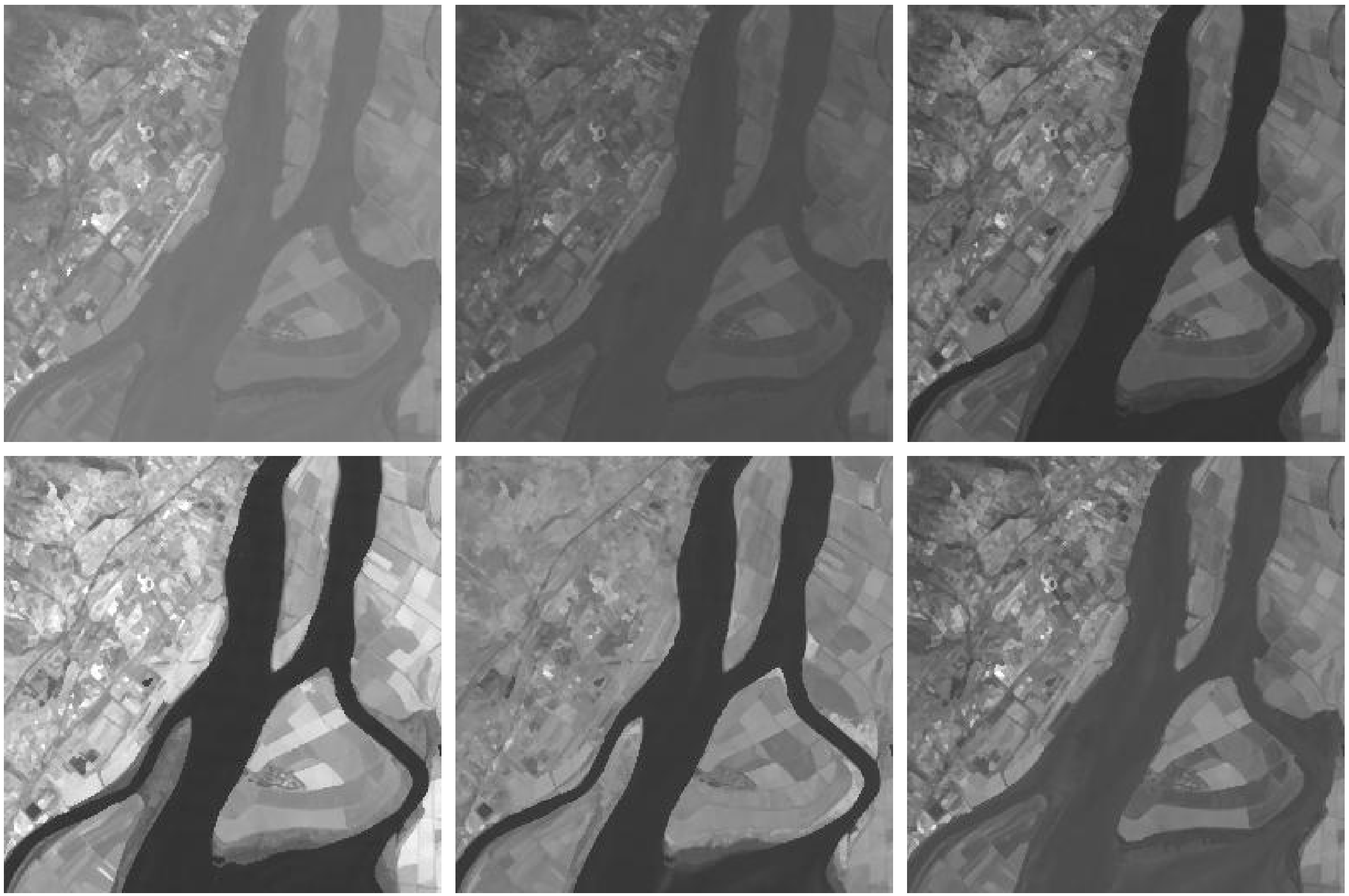




Cost?

7.4 Predictive coding

7.5.1 lossless coding: predictive tree



7.4 Predictive coding

7.5.1 lossless coding: predictive tree

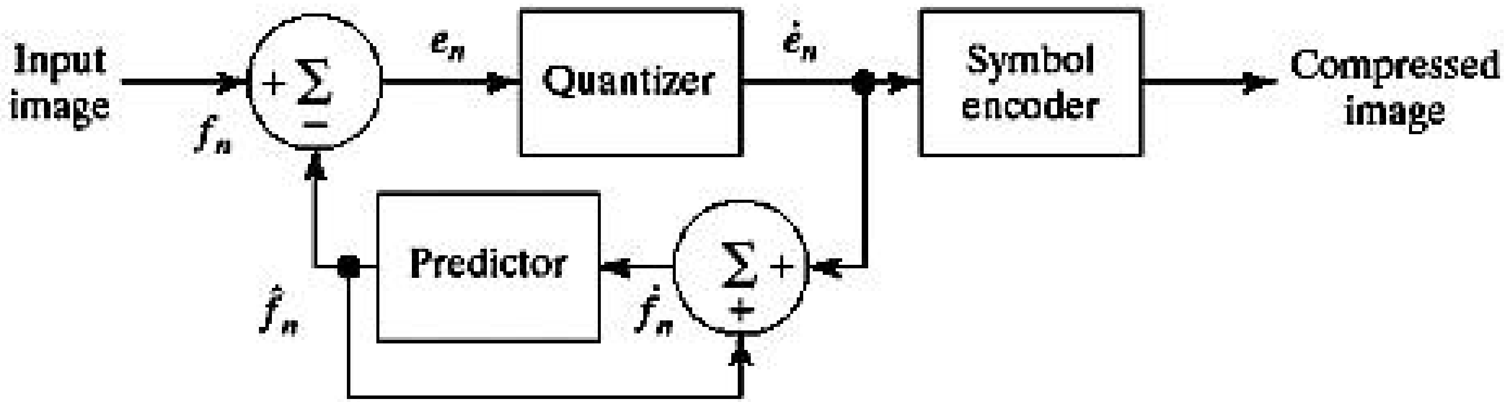
波段	Orignal (bits/pixel)	Best JPEG (bits/pixel)	MAW预测树 ER(bits/pixel)	MAW预测树 T(s)	SNMAW预测树 ER(bits/pixel)	SNMAW预测树 T(s)
1	5.74	3.39	2.24	0.99	1.66	0.09
2	5.19	2.96	1.89	0.78	1.36	0.09
3	6.24	3.71	2.49	0.97	1.91	0.09
4	6.04	3.71	2.53	1.04	1.96	0.09
5	6.48	4.20	2.91	1.22	2.32	0.09
6	6.02	3.58	2.40	0.91	1.83	0.09
平均	**6.04**	**3.59**	**2.41**	**0.99**	**1.84**	**0.09**

7.4 Predictive coding

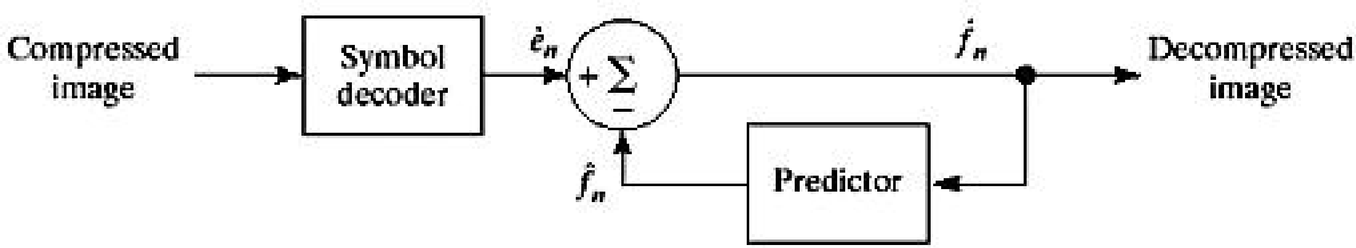
7.5.2 lossy coding: model

encoder

$$\dot{f}_n = \dot{e}_n + \hat{f}_n$$



decoder



7.4 Predictive coding

7.5.2 lossy coding: Optimal predictors

differential pulse code modulation(DPCM)

- Minimizes the encoder's mean-square prediction error

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\}$$

Subject to the constraint that

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i}$$

$$\Rightarrow \quad E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^{m} \alpha_i f_{n-1}\right]^2\right\}$$

7.4 Predictive coding

7.5.2 lossy coding: Optimal predictors

Experiment: four predictors:

(1) $\hat{f}(x,y) = 0.97f(x,y-1)$

(2) $\hat{f}(x,y) = 0.5f(x,y-1) + 0.5f(x-1,y)$

(3) $\hat{f}(x,y) = 0.75f(x,y-1) + 0.75f(x-1,y) - 0.5f(x-1,y-1)$

(4) $$\hat{f}(x,y) = \begin{cases} 0.97f(x,y-1) & \text{if } \Delta h \le \Delta v \\ 0.97f(x-1,y) & \text{otherwise} \end{cases}$$

where

$$\Delta h = \left| f(x-1,y) - f(x-1,y-1) \right|$$

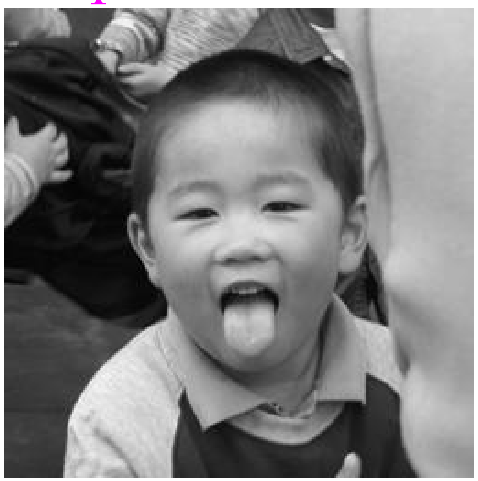
$$\Delta v = \left| f(x,y-1) - f(x-1,y-1) \right|$$

$f(x-1,y-1)$	$f(x-1,y)$	$f(x-1,y+1)$
$f(x,y-1)$	$f(x,y)$	

7.4 Predictive coding

7.5.2 lossy coding: Optimal predictors

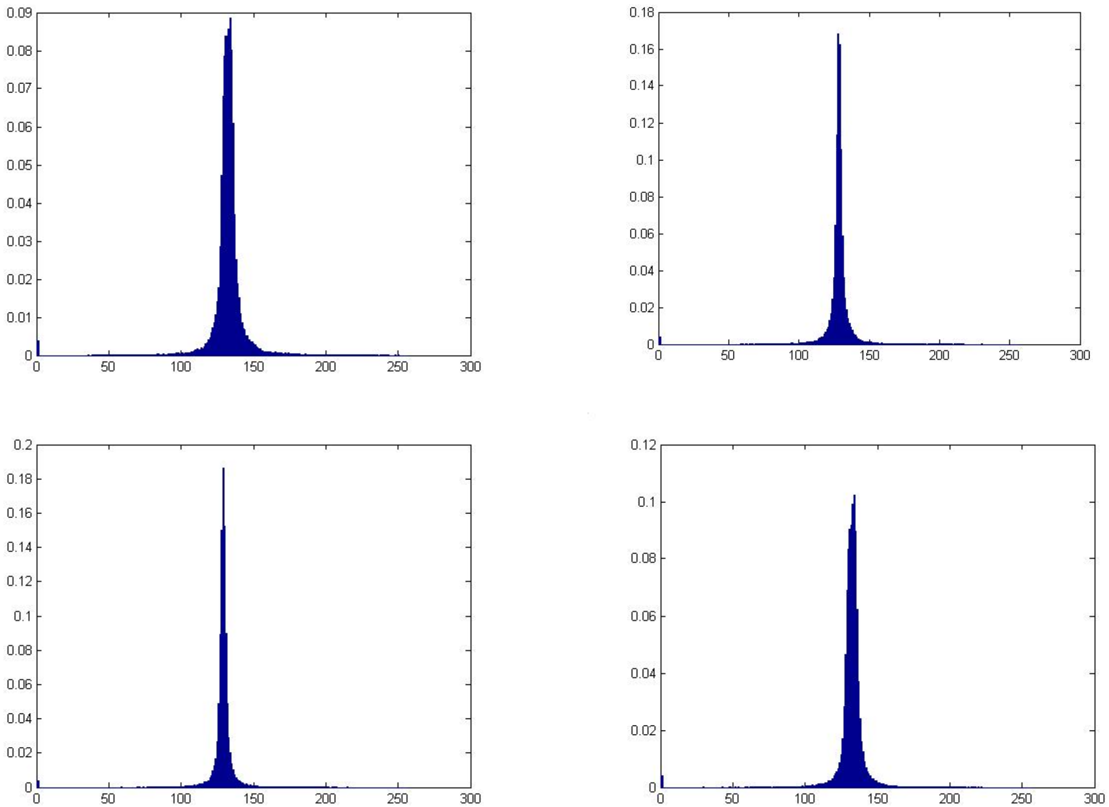
Experiment



original

7.4 Predictive coding

7.5.2 lossy coding: Optimal predictors

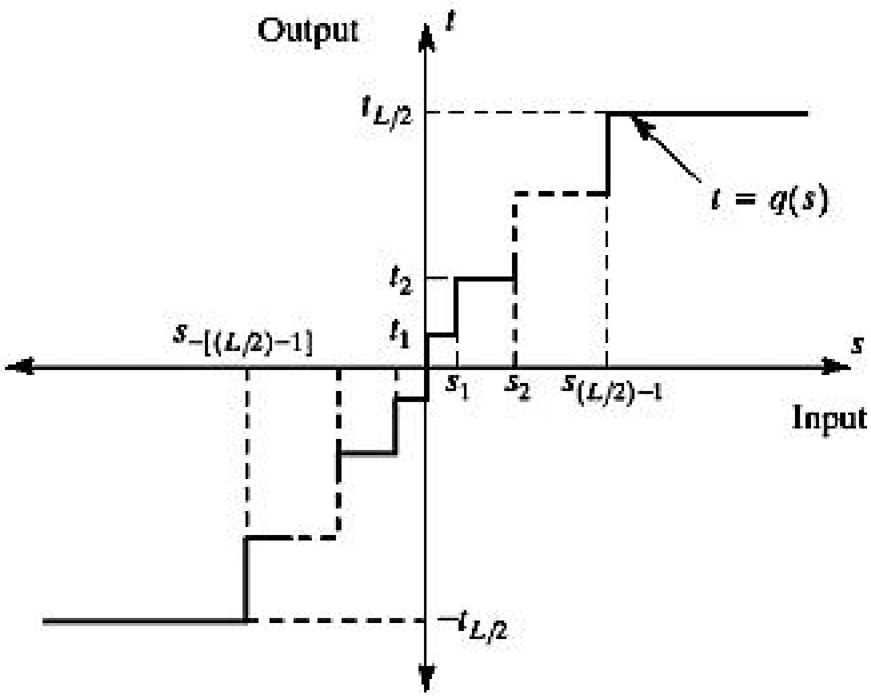


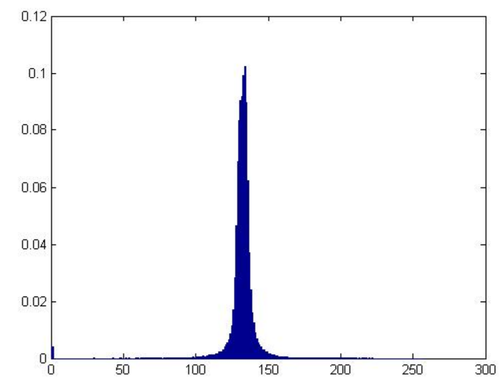
Conclusion:Error decreases as the order of the predictor increases

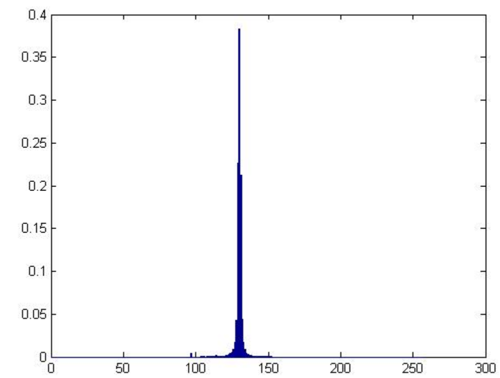
7.4 Predictive coding

7.5.2 lossy coding: Optimal quantization

The staircase quantization function *t=q(s)* is shown as







7.6 Transform Coding

7.6.1 Review of image transform: definitions

$$\mathbf{X} \rightarrow \boxed{\boldsymbol{T}} \rightarrow \mathbf{Y}$$

Direct transform $Y = TX$ $\Rightarrow$ $y_{m,n} = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} x_{i,j}\Phi(i,j,m,n)$

Inverse transform $X = T^{-1}Y$

polynomial expression

Vector expression

Question: why transformation can compress data?

7.6 Transform Coding

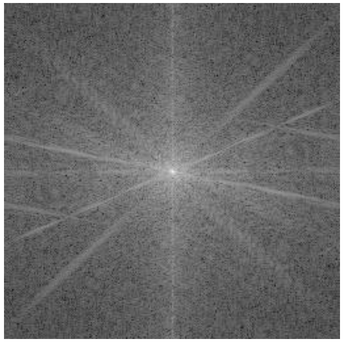
7.6.1 review of image transform: properties

- **Entropy keeping:** $H(X)=H(Y)$
- **Energy keeping:** $$\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}|X(m,n)|^2=\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}|Y(k,l)|^2$$
- **decorrelation:** $$H_\infty=\cdots=H_m<H_{m-1}<\cdots<H_2<H_1<H_0$$

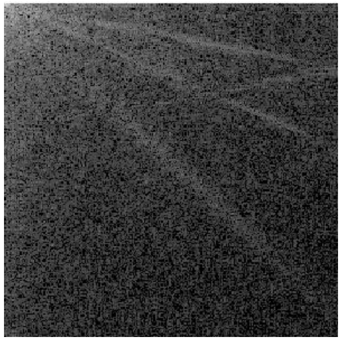
$$H(Y)=H_0(Y) \qquad H(X)=H_m(X) \qquad H_0(X)>H_0(Y)$$

- **Energy re-assigned:**

original



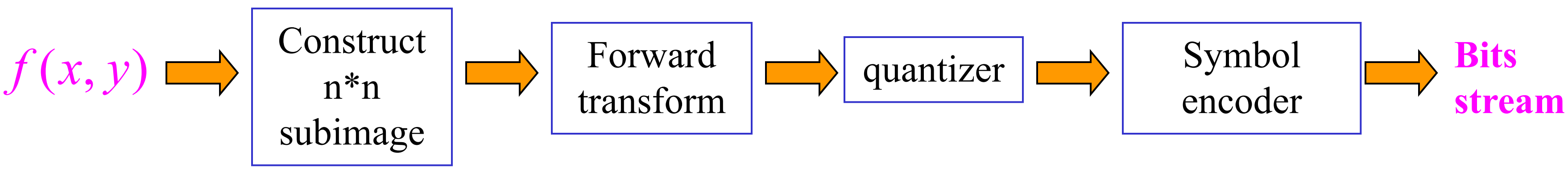
DFT



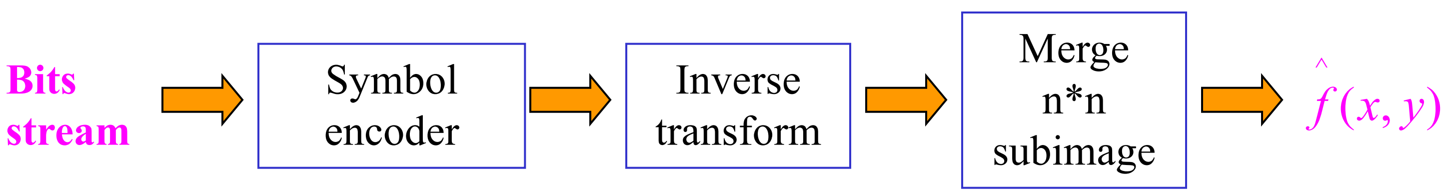
DCT

7.6 Transform Coding

7.6.2 Transform coding system



encoder



decoder

7.6 Transform Coding

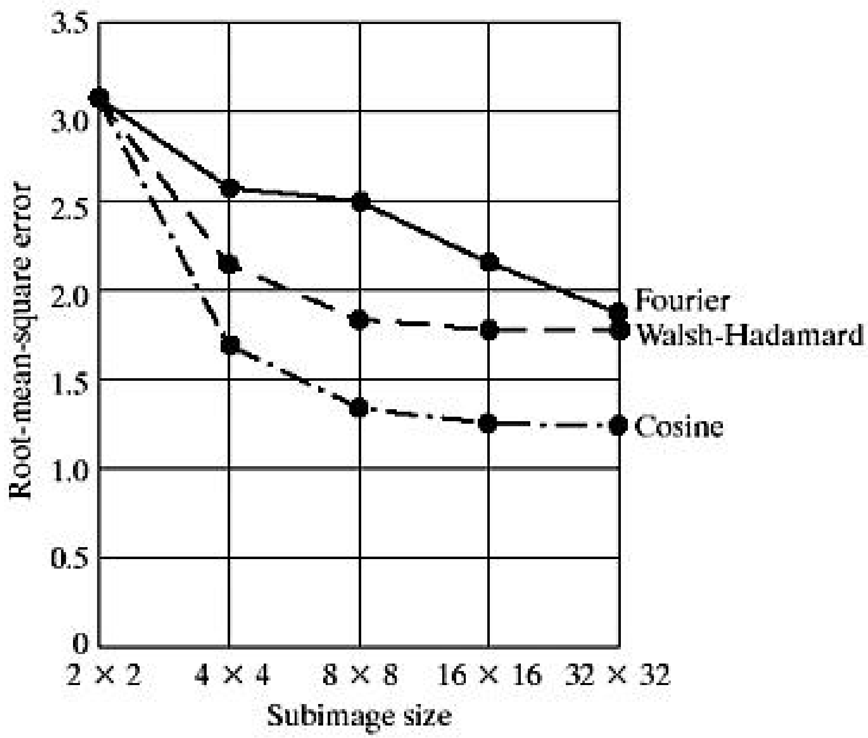
7.6.3 Transform selection

Information packing ability: *KLT>DCT>DFT>WHT*

Computational complexity: *WHT<DCT<DFT<KLT*

7.5.4 sub-image size selection

- Computational complexity increase as the subimage size increase
- Correlation decrease as the subimage size increase
- The most popular subimage size are 8*8, and 16*16



7.6 Transform Coding

7.6.5 bit allocation: zonal coding

- transform coefficients of maximum variance carry the most information and should be retained

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	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
0	0	0	0	0	0	0	0

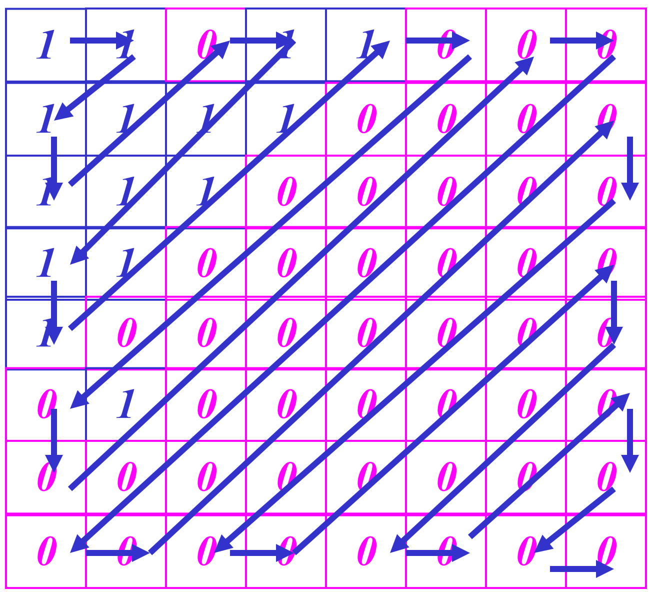
Zonal mask

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Bit allocation

7.6 Transform Coding

7.6.5 bit allocation: threshold coding



threshold mask

Zig-zag

7.6 Transform Coding

7.6.5 bit allocation: threshold coding

There are three basic ways to threshold a transformed coefficients

- A single *global threshold* can be applied to all subimages
- A *different threshold* can be used for each subimage
- The threshold can be varied as a *function* of the location of each coefficient within the subimage

$$\hat{T}(u,v) = round\left[\frac{T(u,v)}{Z(u,v)}\right]$$

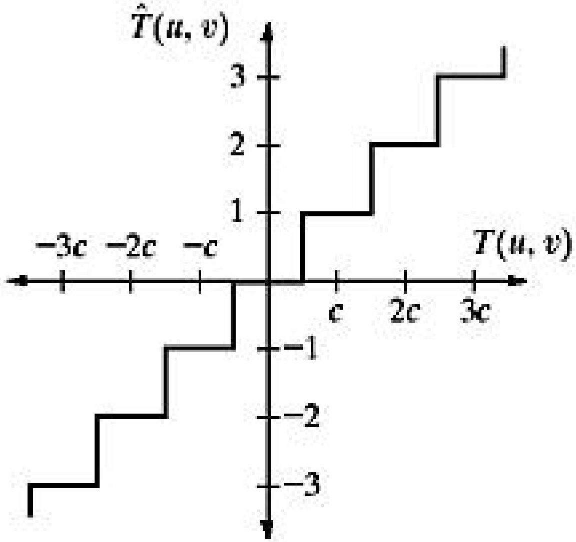
where *T(u,v):* transform coefficients

Z(u,v): quantization matrix

7.6 Transform Coding

7.6.5 bit allocation: threshold coding

Z(u,v) is assigned a particular value *c*



Z(u,v) used in JPEG stardard

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

7.6 Transform Coding

7.6.5 bit allocation: threshold coding

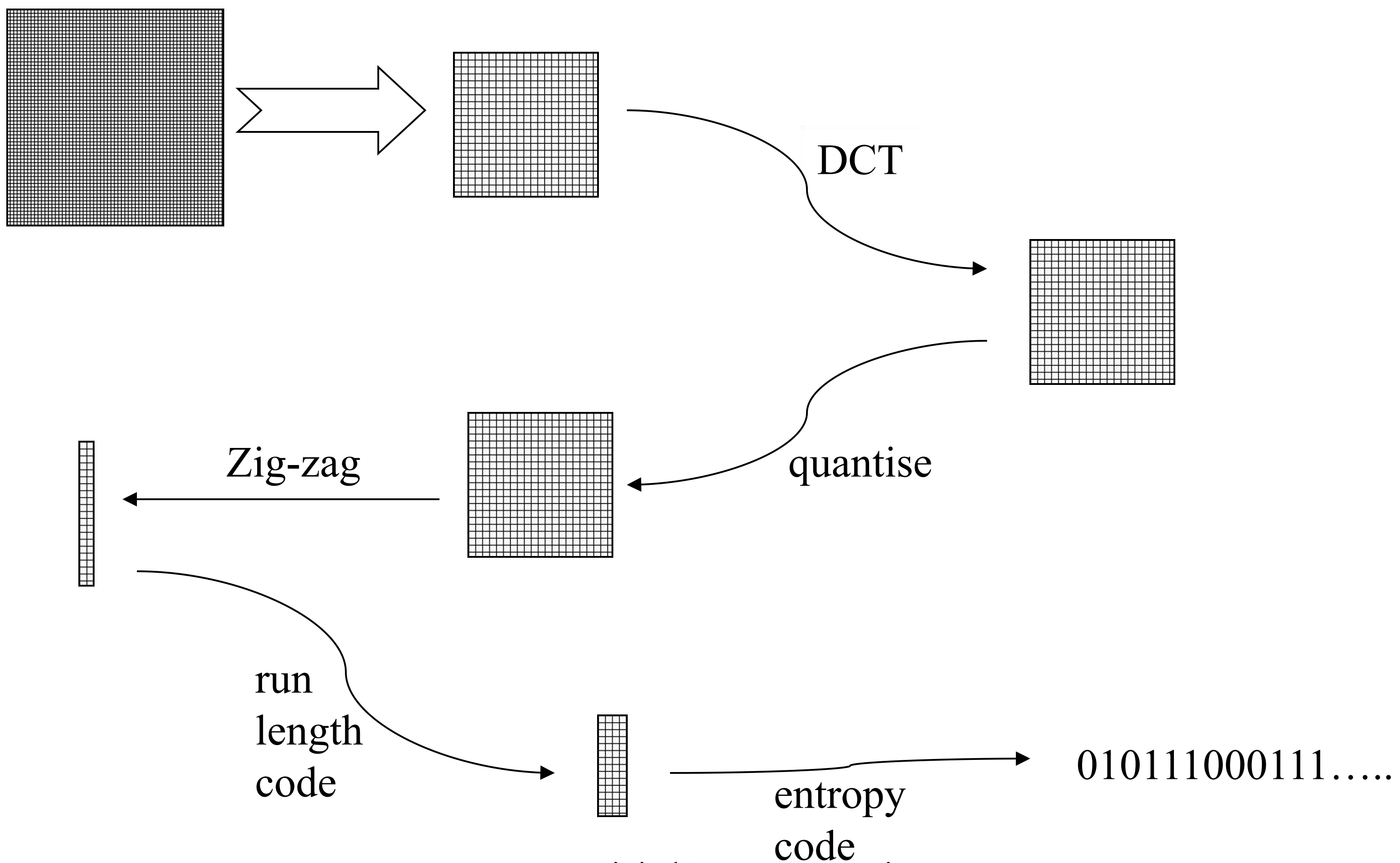
Experiment results

Left column: quantize with *Z(u,v)*

Right column: quantize with 4**Z(u,v)*

7.5 Transform Coding

7.6.3 JEPG lossy compression standard



Original image

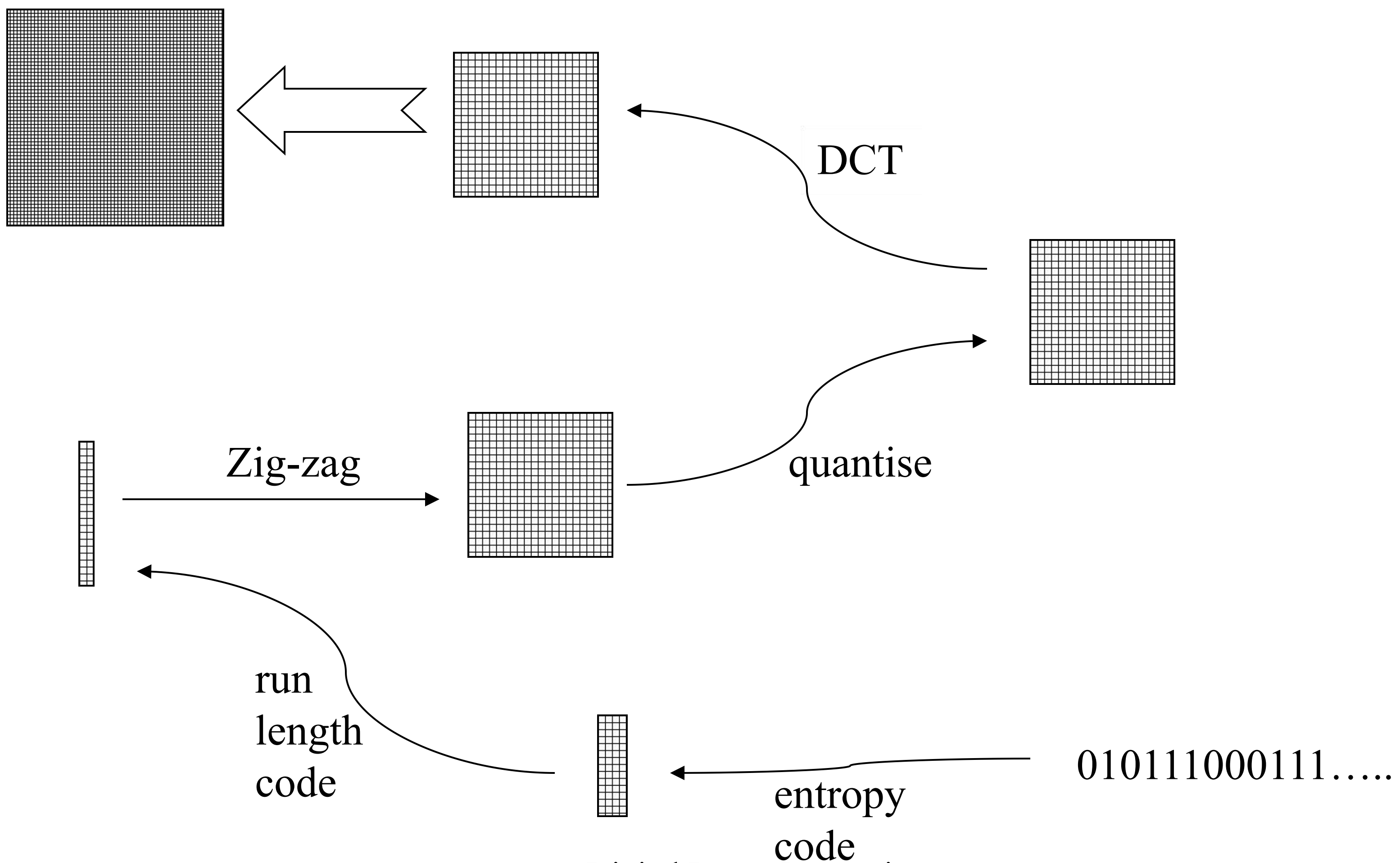
$$\begin{matrix} 139 & 144 & 149 & 153 \\ 144 & 151 & 153 & 156 \\ 150 & 155 & 160 & 163 \\ 159 & 161 & 162 & 160 \end{matrix} \xrightarrow{\text{DCT}} \begin{matrix} 1260 & -1 & -12 & -5 \\ -23 & -17 & -6 & -3 \\ -11 & -9 & -2 & 2 \\ -7 & -2 & 0 & 1 \end{matrix}$$

$$\xrightarrow{\text{quantise}} \begin{matrix} 79 & 0 & -1 & 0 \\ -2 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

$$\xrightarrow{\text{Zig-zag}} 79\ 0\ -2\ -1\ -1\ -1\ 0\ 0\ -1\ 0\ 0\ 0\ 0\ 0\ 0$$

$$\xrightarrow{\text{run length code}} \begin{matrix} 0 & 79 \\ 1 & -2 \\ 0 & -1 \\ 0 & -1 \\ 0 & -1 \\ 2 & -1 \\ 0 & 0 \end{matrix} \xrightarrow{\text{Huffman code}} 10011011100011\ldots.$$

JPEG Decoding



Result of Coding and Decoding

$$\begin{pmatrix} 139 & 144 & 149 & 153 \\ 144 & 151 & 153 & 156 \\ 150 & 155 & 160 & 163 \\ 159 & 161 & 162 & 160 \end{pmatrix}$$

Original block

$$\begin{pmatrix} 144 & 146 & 149 & 152 \\ 148 & 150 & 152 & 154 \\ 155 & 156 & 157 & 158 \\ 160 & 161 & 161 & 162 \end{pmatrix}$$

Reconstructed block

$$\begin{pmatrix} -5 & -2 & 0 & 1 \\ -4 & 1 & 1 & 2 \\ -5 & -1 & 3 & 5 \\ -1 & 0 & 1 & -2 \end{pmatrix}$$

errors

7.5 Transform Coding

7.6.3 JEPG lossy compression standard

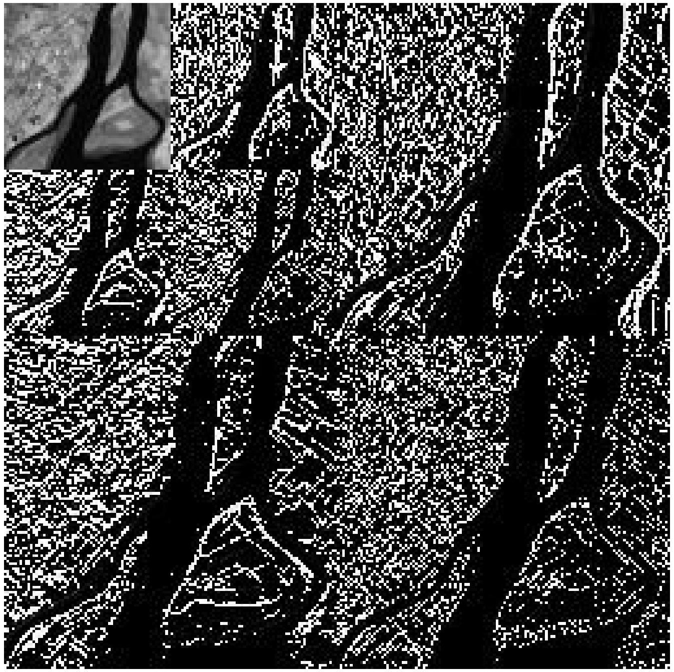
$f(x,y)$ → Construct 8*8 subimage → DCT → quantizer → Zig-zag → Huffman coding → **Bits stream**

encoder

Question: why there are mosaics in JEPG images?

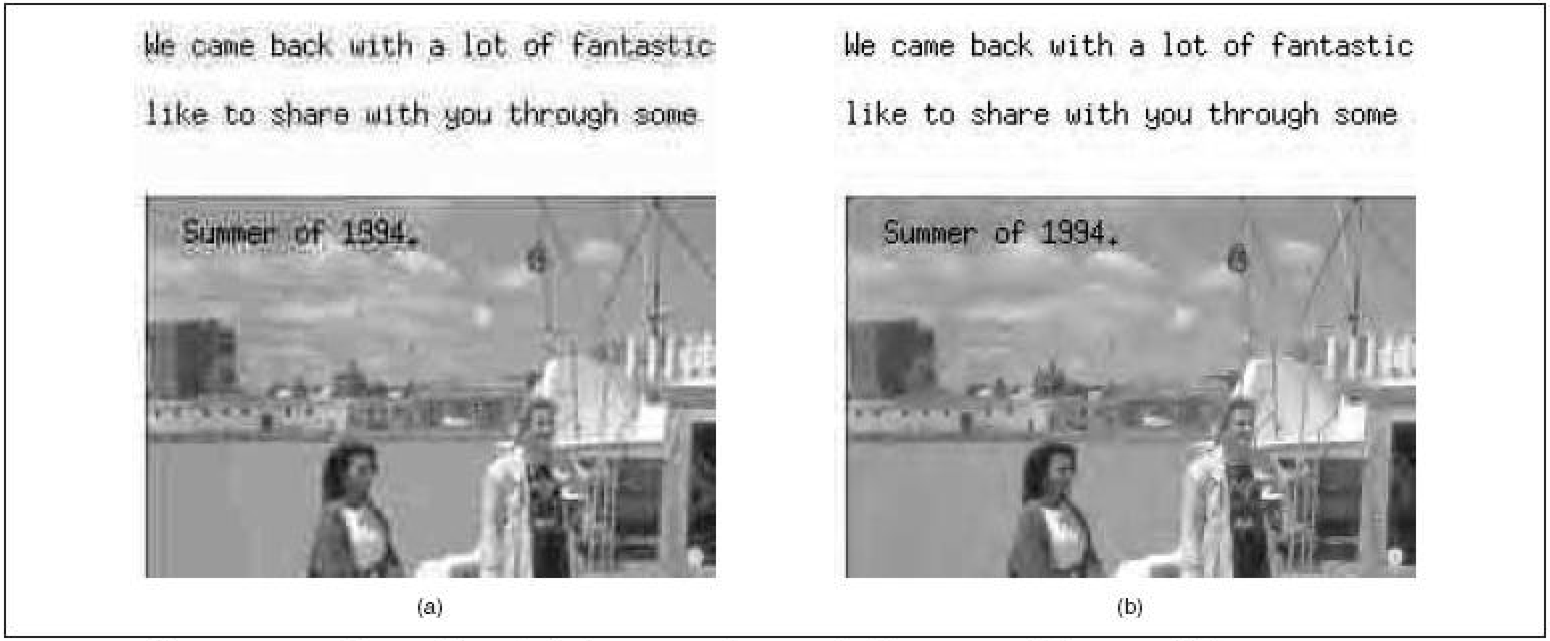
7.5 Transform Coding

7.6.3 JEPG 2000 lossy compression standard



(a) (b)

▲ 21. Reconstructed image "ski" after compression at 0.25 b/p by means of (a) JPEG and (b) JPEG 2000.



▲ 22. *Part of the reconstructed image "cmpnd1" after compression at 0.5 b/p by means of (a) JPEG and (b) JPEG 2000.*

7.7 Introduction to international standards

- Image compression standards:
  - -JPEG,
  - -JPEG2000
- Video compression standards:
  - -MPEG-1,MPEG-2,MPEG-4,
  - -H.261,H.263,H.263+,H.264
  - -H.264/AVC

Video compression standards

ITU-T Video Coding Experts Group (VCEG)

– H.261 (1990) → H.263 (1995) → H.263+ (1998) → H.26L

ISO Motion Picture Experts Group (MPEG)

– MPEG1(1991) → MPEG2 (1994) → MPEG4 (1999)

Joint Video Team (JVT) (VCEG/MPEG) 2001

– H.264/MPEG4 part 10. Official title: Advanced Video Coding (AVC)

– International standard: December 2002

H.264

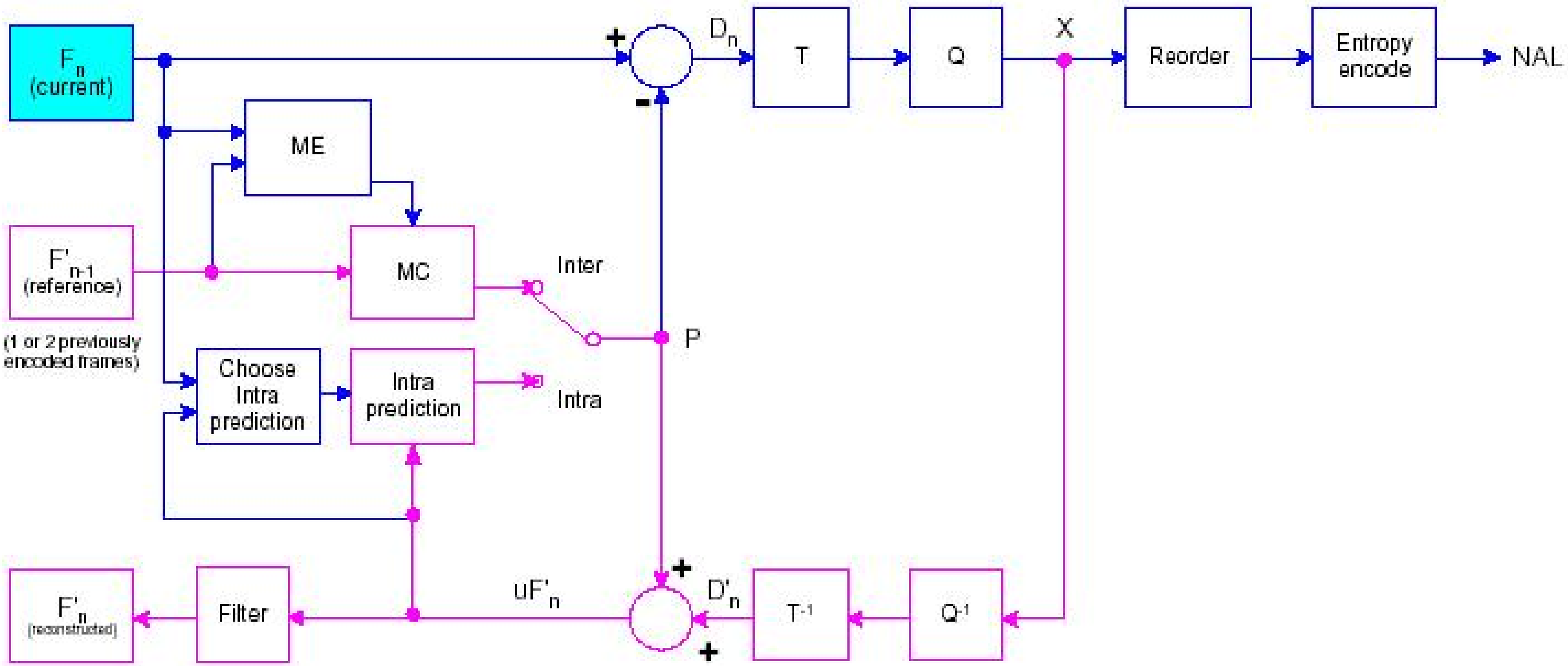
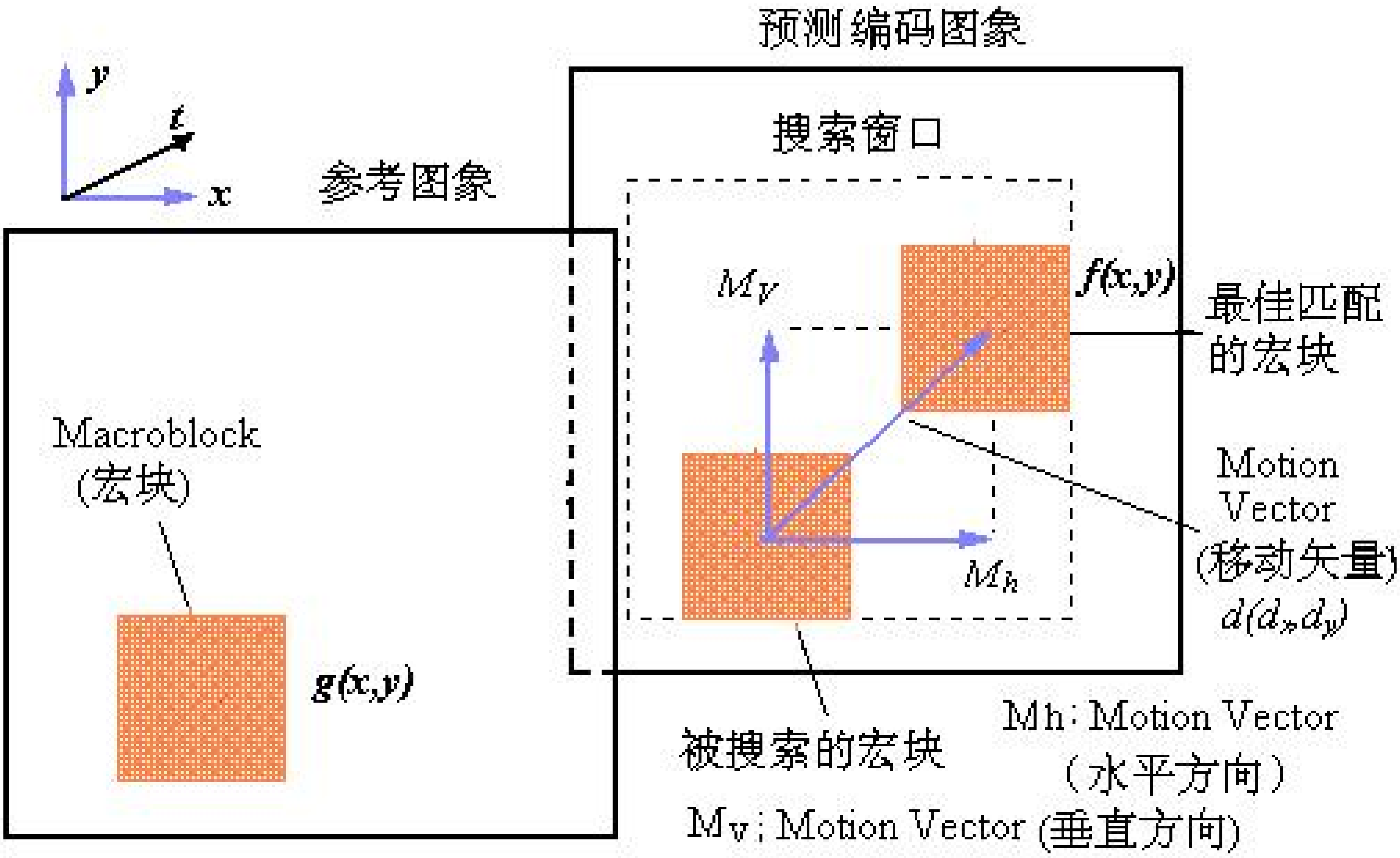


Figure 2-1 AVC Encoder

ME

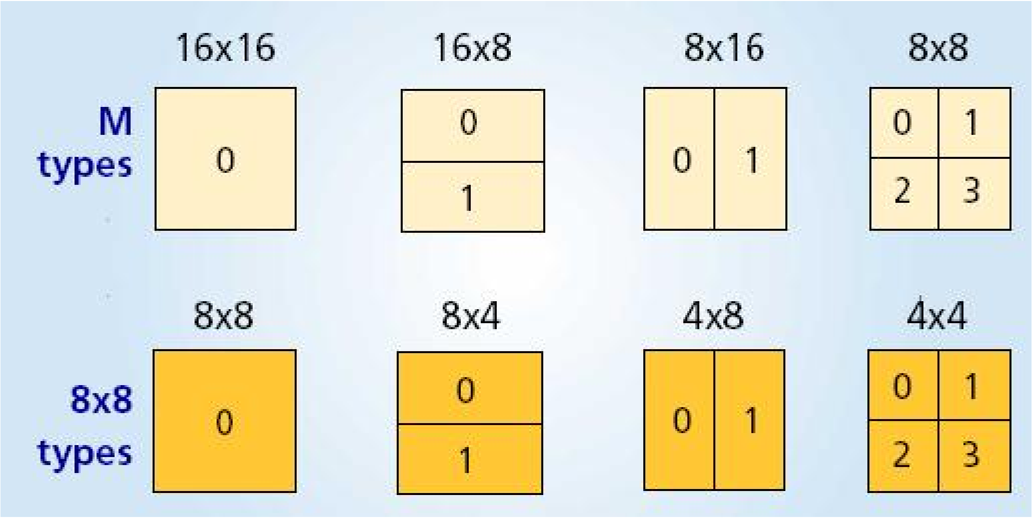


y
t
x
参考图象
预测编码图象
搜索窗口
Macroblock
(宏块)
g(x,y)
f(x,y)
最佳匹配
的宏块
M_V
M_h
Motion
Vector
(移动矢量)
$d(d_x,d_y)$
被搜索的宏块
Mh : Motion Vector
（水平方向）
Mv : Motion Vector (垂直方向)

Prediction of inter Macroblocks –tree structured MC

16*16 block → 16*16, 16*8, 8*16, 8*8

8*8 block → 8*8



These partition and sub-partition give rise to a large mumber of possible combinations within a macroblock

Example:

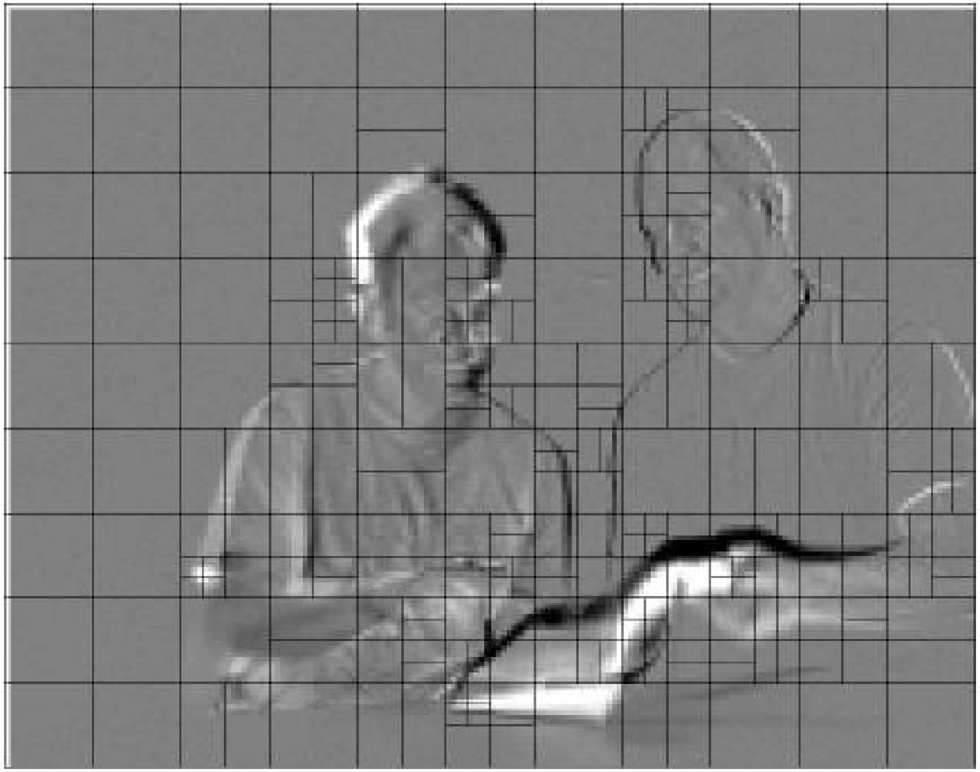
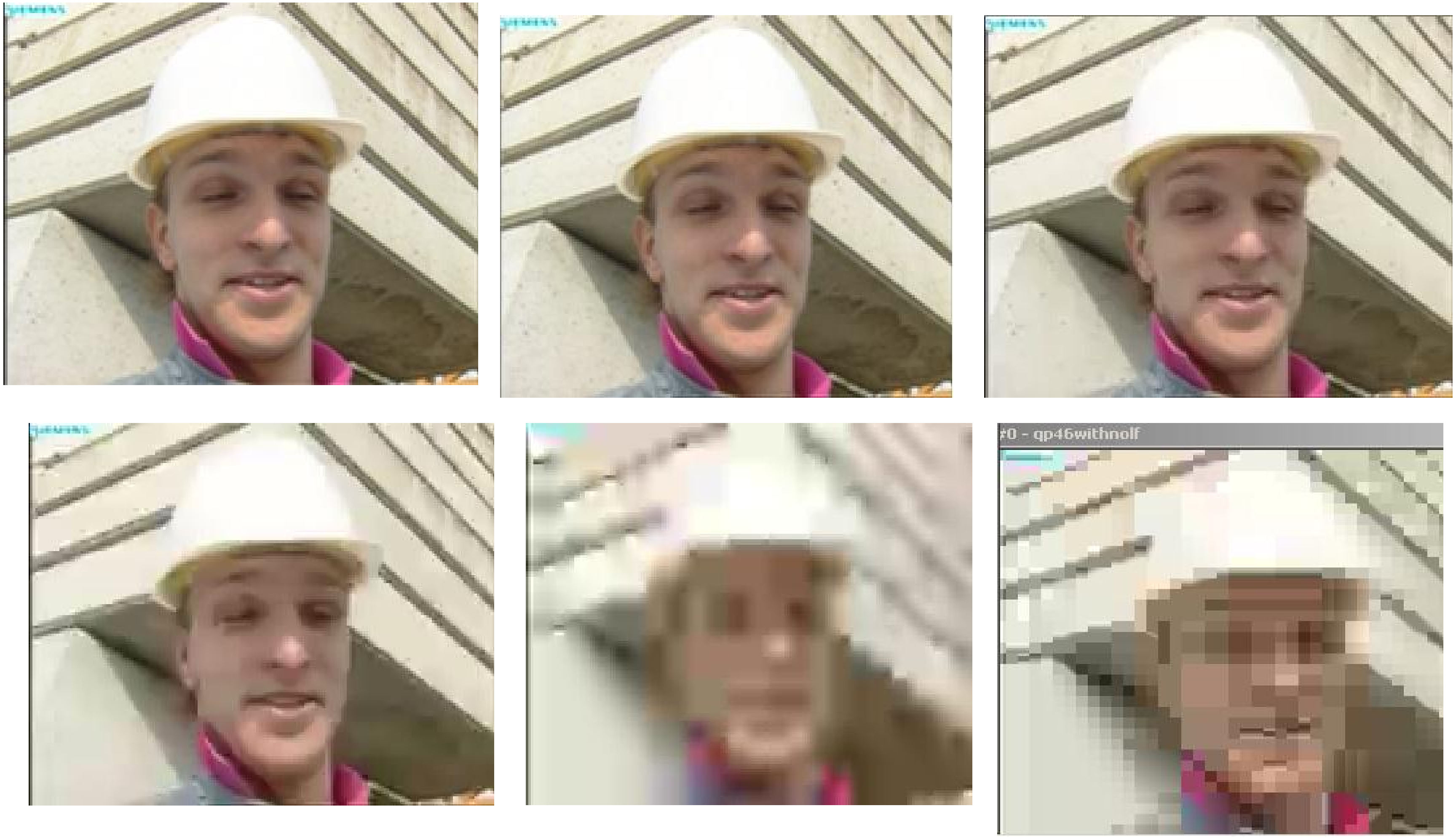


Figure 2-3 Residual (without MC) showing optimum choice of partitions

quantization



QP＝5， 15， 25， 35， 50

deblocking

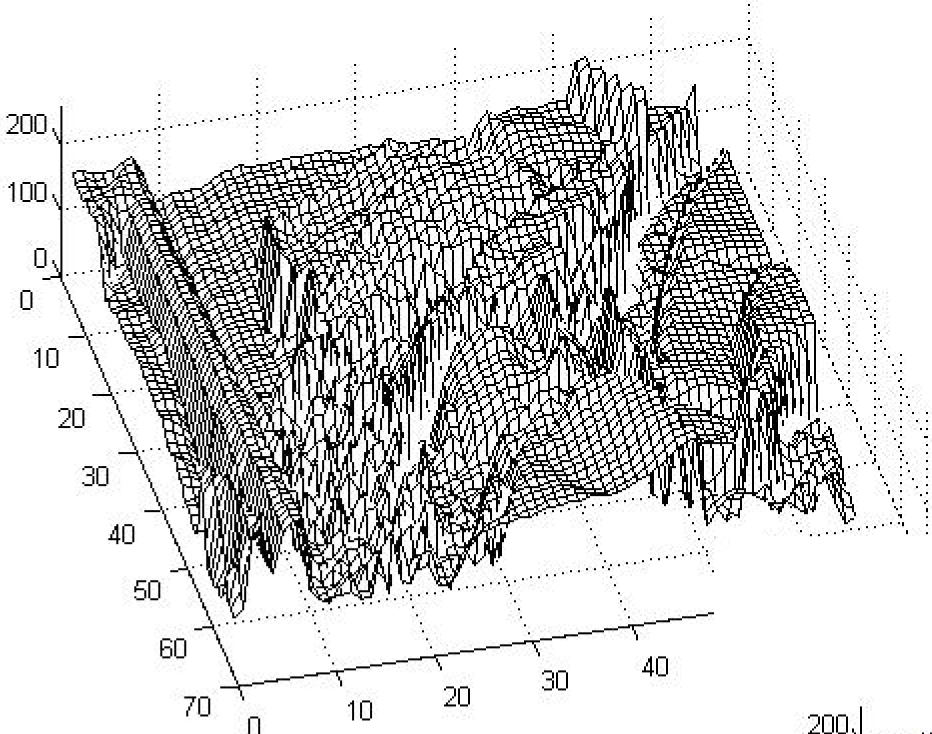
fig6 the comparing of deblocking and non-deblocking（QP=36）

Homework

1、 (1)请说明是否能用变长变码法压缩1幅已直方图均衡化的具有
2^n级灰度的图？
(2) 这样的图像中包含像素间冗余吗？

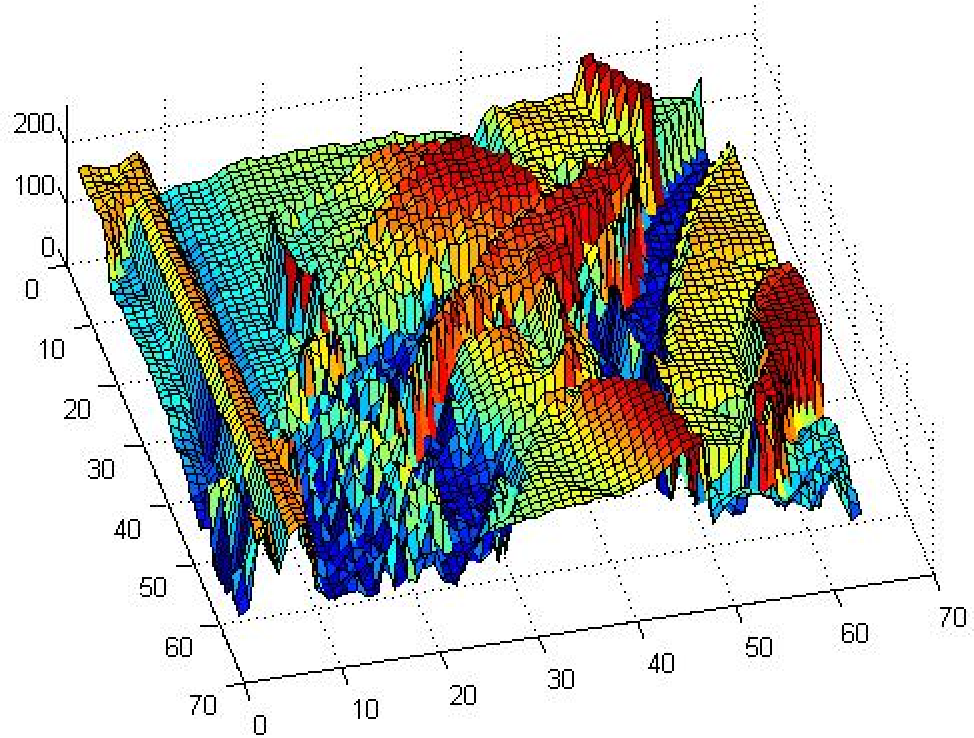
2、
(1) 对一个具有3个符号的信源，有多少唯一的Huffman码？
(2)构造这些码。

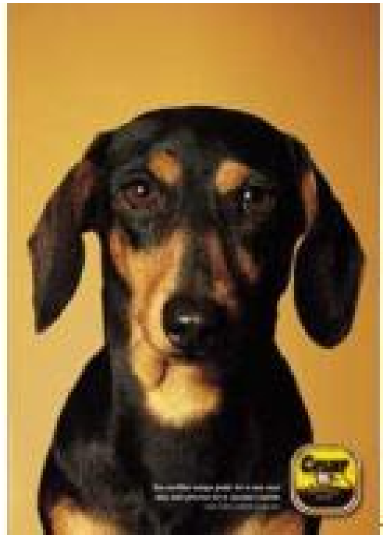
3、 已知符号a,e,i,o,u,?的出现概率分别是0.2，0.3，0.1，0.2，0.1
0.1，0.23355进行解码，解码长度为6。



difficult in DIP :

low feature extraction





difficult in IU: Semantic Problem

科大西区

West campus of USTC