

Цифрова обработка на изображения

Уейвлетни преобразувания
Компресия на изображения

Пространствена резолюция

- **Резолюция (*resolution*)**

- разделителна способност

- **Пространствена резолюция**

- определя се от пространствената дискретизация
- най-малкият детайл, различим в изображението

Разделительна способност

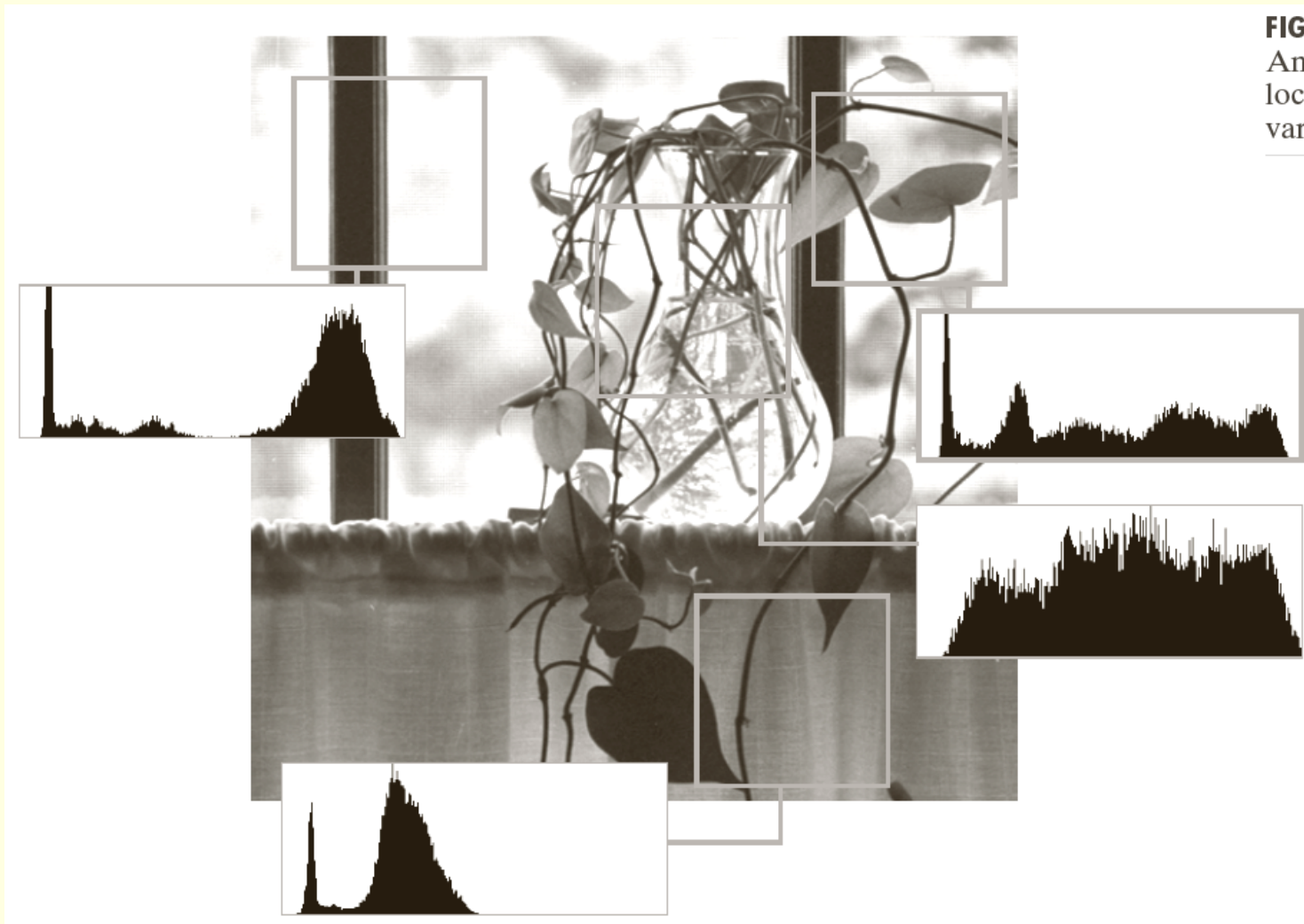


FIGURE 7.1
An image and its
local histogram
variations.

Пространствена резолюция

- ***Мултирезолюция (multiresolution)***
 - разглеждането на различни резолюции за едно и също изображение носи полезна информация за съдържанието на изображението
 - малки или ниско-контрастни обекти
 - изискват висока резолюция
 - големи или контрастни обекти
 - достатъчна е ниска резолюция

Пирамиди

- Представяне на изображенията при различни резолюции
- ***Пирамида***
 - съвкупност от изображения с намаляваща разделителна способност
 - най-висока разделителна способност в основата на пирамидата
 - най-ниска разделителна способност на върха на пирамидата
 - с изкачване нагоре в пирамидата намалява размера и резолюцията на изображението

Пирамиди

- Изображението J в основата на пирамидата има размери $N \times N$, или $2^J \times 2^J$ за $J = \log_2 N$
- Изображенията на междинните нива j имат размери $2^j \times 2^j$ за $0 \leq j \leq J$

- ***Пълна пирамида***

- има $J+1$ нива
 - изображенията са с размери от $2^J \times 2^J$ до $2^0 \times 2^0$

- ***Непълна пирамида***

- има $P+1$ нива
 - $j = J - P, \dots, J - 2, J - 1, J$
 - $1 \leq P \leq J$

Пирамиди

- **Създаване на пирамида**

- итеративен процес с няколко стъпки

- (1) намаляване на резолюцията (***downsampling***)

- (2) увеличаване на резолюцията (***upsampling***)

- (3) изчисляване на разлики (***residuals***)

Пирамиди

■ *Създаване на пирамида*

(1) намаляване на резолюцията (***downsampling***)

- от изображението на ниво j , $0 \leq j \leq J$, се получава изображение с редуцирана резолюция
 - полученото изображение е апроксимация за ниво $(j - 1)$ на входното изображение

Пирамиди

■ *Създаване на пирамида*

(2) увеличаване на резолюцията (*upsampling*)

- от апроксимацията за ниво $(j - 1)$ се получава изображение с увеличени размери (и резолюция)
 - полученото изображение е интерполирана оценка за ниво $(j - 1)$ на входното изображение

Пирамиди

- **Създаване на пирамида**

- (3) изчисляване на разлики (*residuals*)

- разлика на интерполираното изображение за ниво $(j - 1)$ и оригиналното изображение за ниво j
 - полученото изображение съдържа разликите за ниво j

Пирамиди

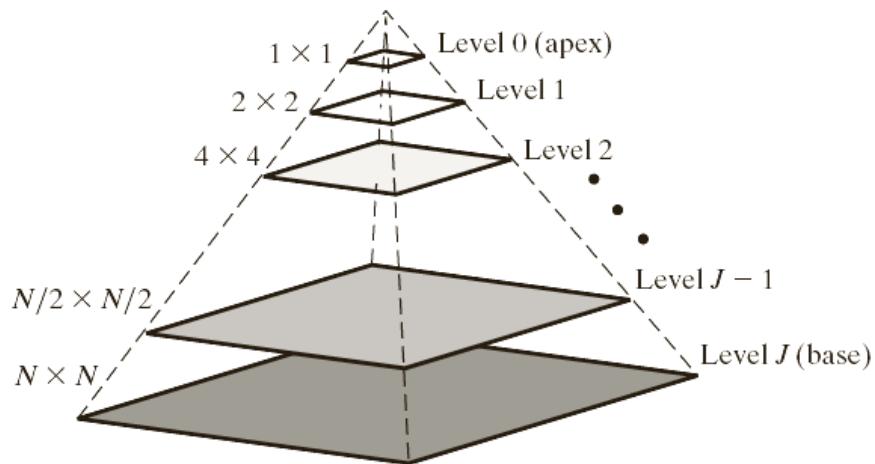
■ *Апроксимации*

- прилага се редуциране на размера на изображението
 - премахване на пиксели
 - ефект aliasing на горните нива на пирамидата
 - филтър по средна стойност
 - Гаусов филтър

■ *Интерполации*

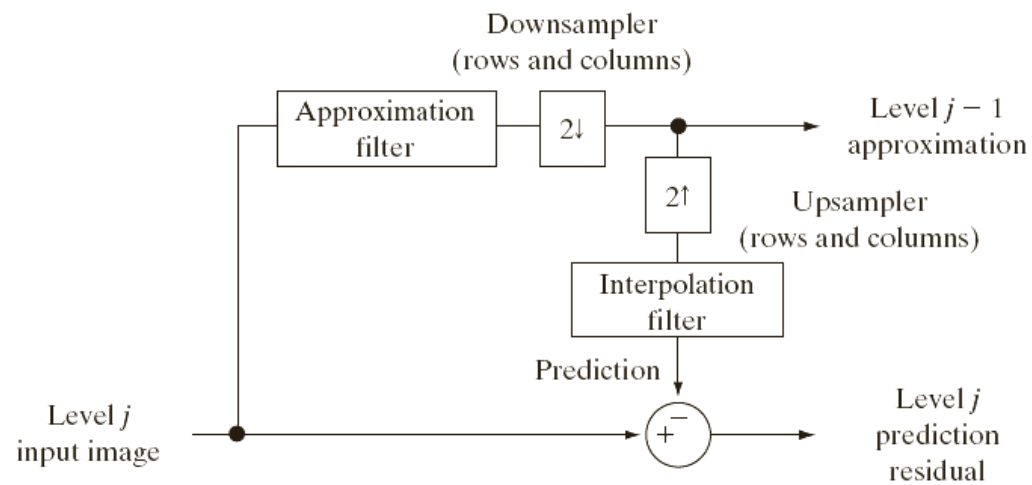
- прилага се увеличаване на размера на изображението
 - повторение на пиксели
 - интерполация
 - най-близки съседни
 - билинейна
 - бикубична

Пирамиди



a
b

FIGURE 7.2
(a) An image pyramid. (b) A simple system for creating approximation and prediction residual pyramids.



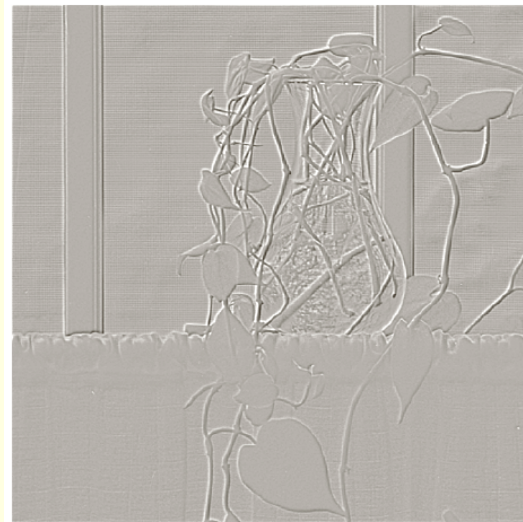
Пирамиди

- Получават се две пирамиди
 - *на апроксимациите*
 - *на разликите*
 - по пирамидата на разликите прогресивно може да се възстанови оригиналното изображение

Пирамиди

■ Гаусова пирамида

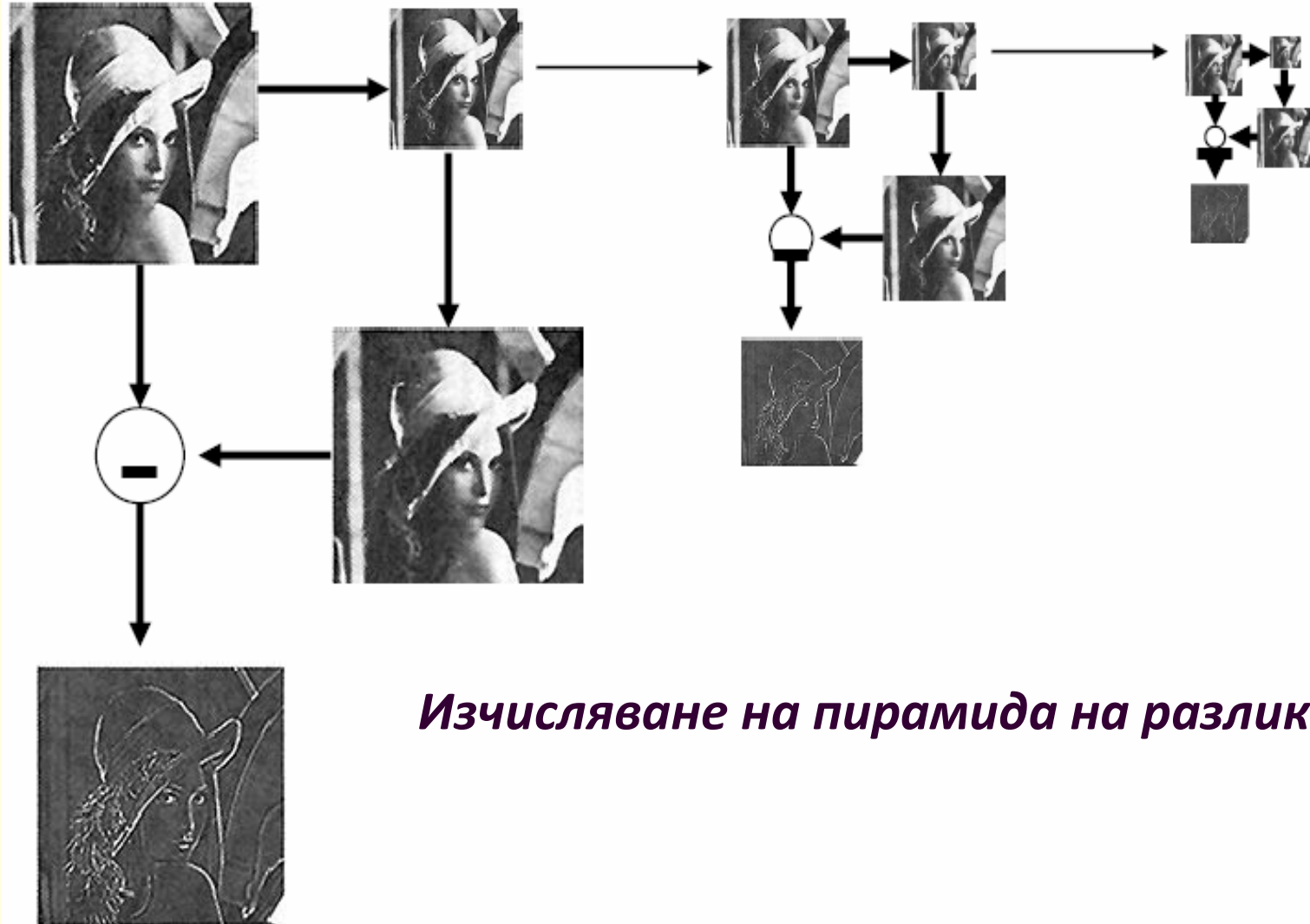
- ниво 0:
- оригинално изображение 512×512



a
b

FIGURE 7.3 Two image pyramids and their histograms: (a) an approximation pyramid; (b) a prediction residual pyramid.

Пирамиди



Изчисляване на пирамида на разликите

Трансформация

- Входен сигнал: 56; 40; 8; 24; 48; 48; 40; 16
- Трансформация
 - **права**: $(a;b) \rightarrow (d;s)$
 - $s = (a + b)/2$
 - $d = a - s$
 - **обратна**: $(d;s) \rightarrow (a;b)$
 - $a = s + d$
 - $b = s - d$

Трансформация

$(56 + 40)/2$				$56 - 48$			
56	40	8	24	48	48	40	16
48				8			

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16			8	-8		

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48		8	-8	0	

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32		16		8	-8	0	12

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
		16	10	8	-8	0	12

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и осреднената стойност

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

- **Права трансформация:** изчислява се
 - средна стойност на всяка двойка стойности
 - разлика на оригиналната и средната стойност

Трансформация

35	-3	16	10	8	-8	0	12

- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Трансформация

32	38						
35	-3	16	10	8	-8	0	12

- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Трансформация

32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Трансформация

48	16	48	28				
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Трансформация

48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Трансформация

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

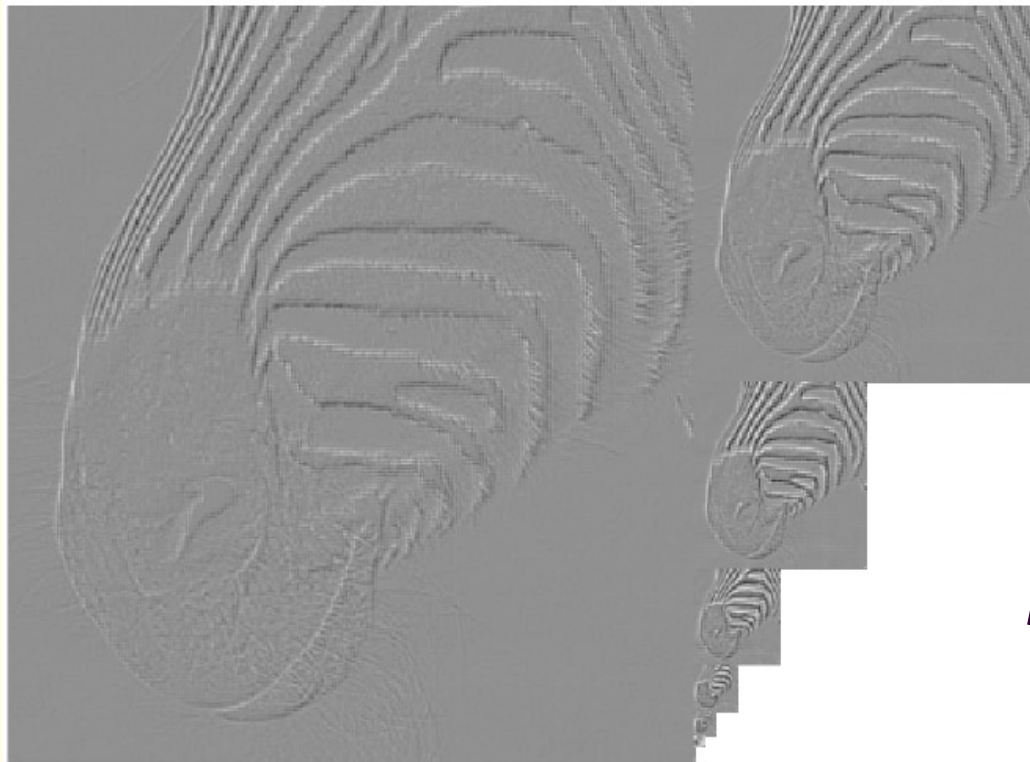
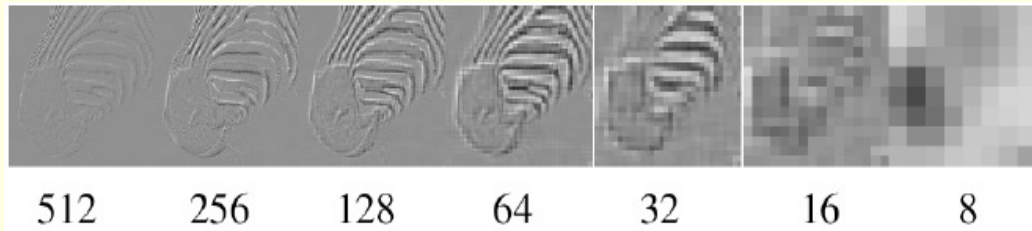
- **Обратна трансформация:** изчислява се
 - сума на средната стойност и разликата
 - разлика на средната стойност и разликата

Пирамиди



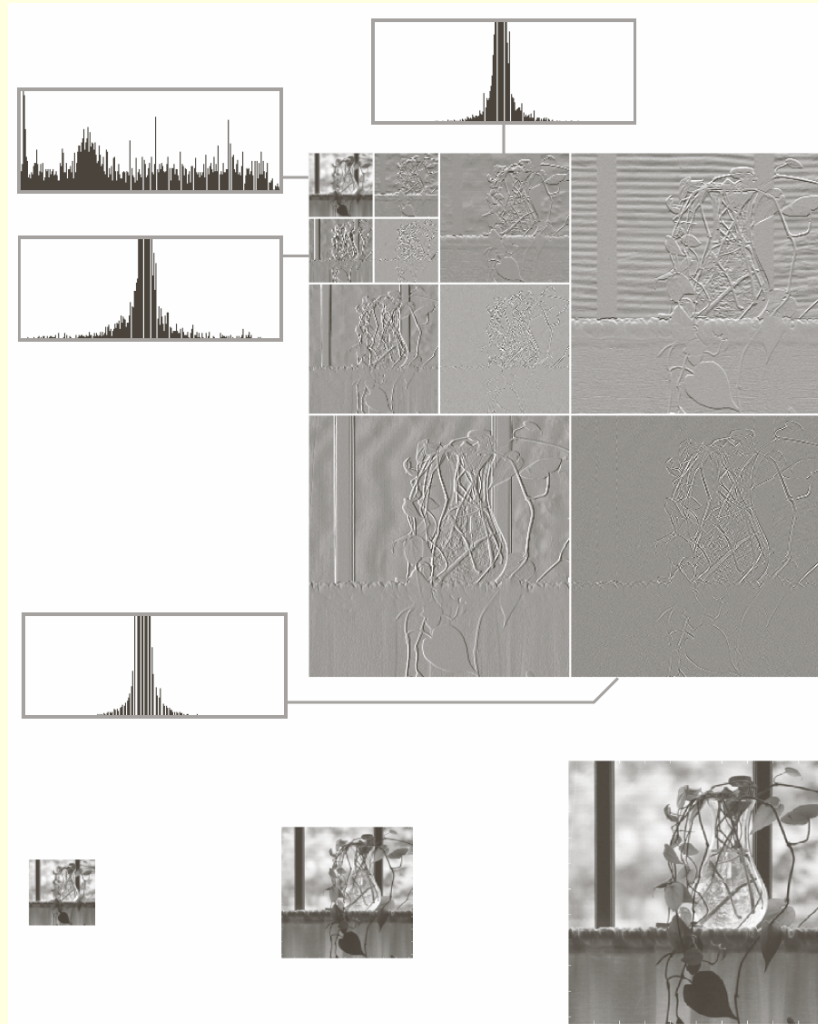
*Пирамида на
апроксимациите*

Пирамиди



*Пирамида на
разликите*

Пирамиди



a
b c d

FIGURE 7.10
(a) A discrete wavelet transform using Haar H_2 basis functions. Its local histogram variations are also shown. (b)–(d) Several different approximations (64×64 , 128×128 , and 256×256) that can be obtained from (a).

Дискретна косинусова трансформация

■ *Discrete Cosine Transform (DCT)*

■ Едномерна

$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \quad f(x) = \sum_{u=0}^{N-1} \alpha(u) C(u) \cos\left[\frac{\pi(2x+1)u}{2N}\right]$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0. \end{cases}$$

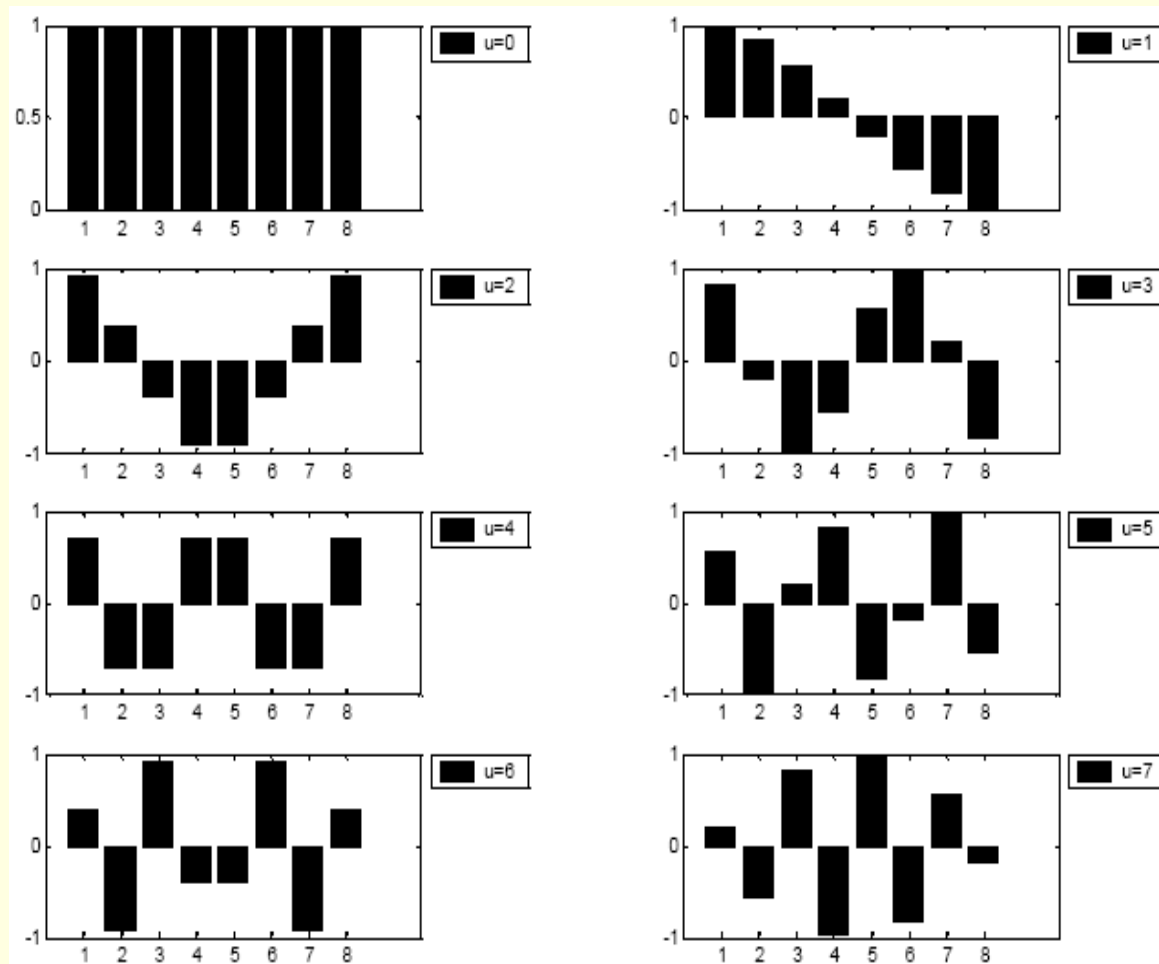
■ Двумерна

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u, v) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

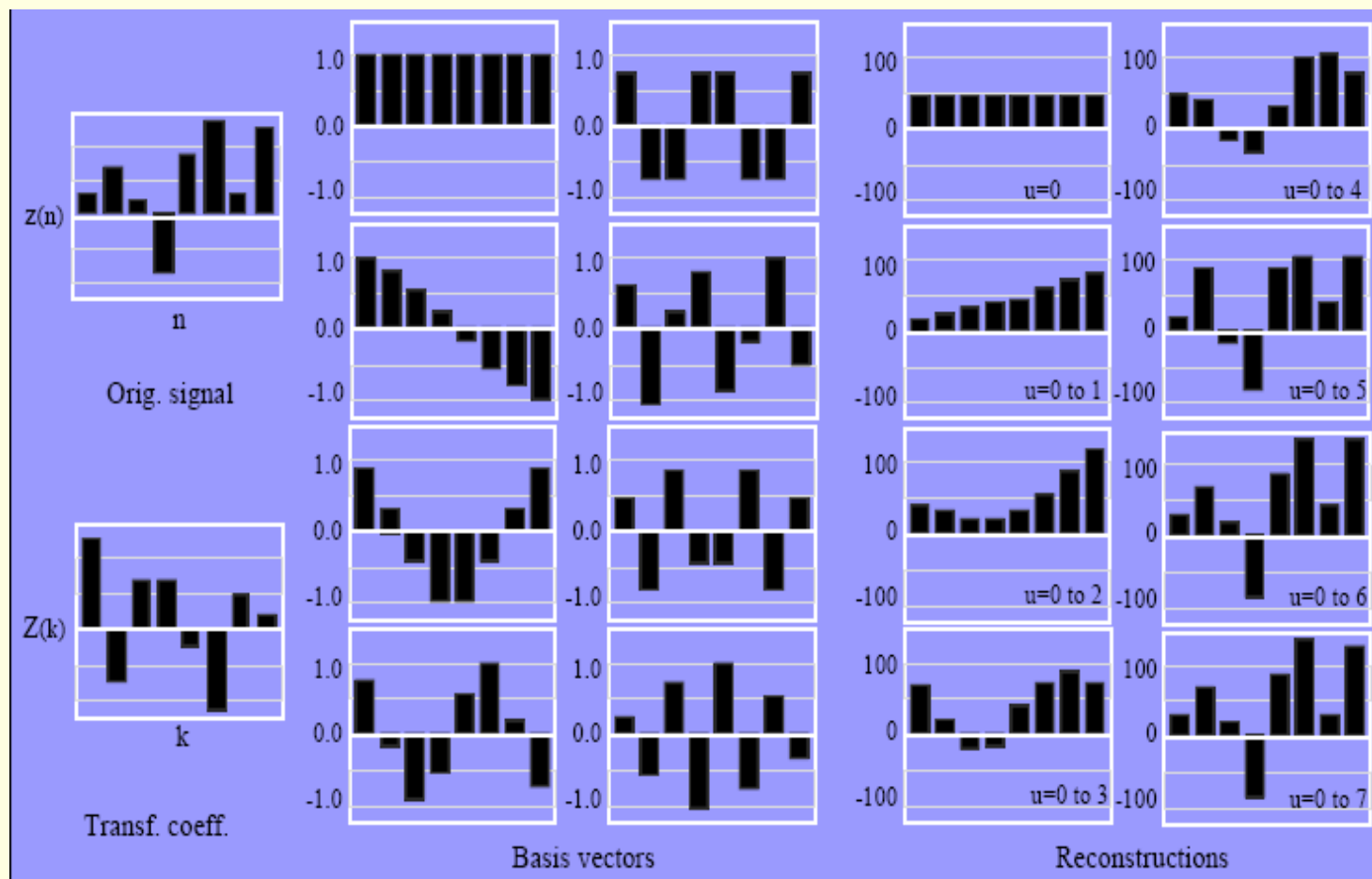
Дискретна косинусова трансформация

■ 1D DCT за $N=8$



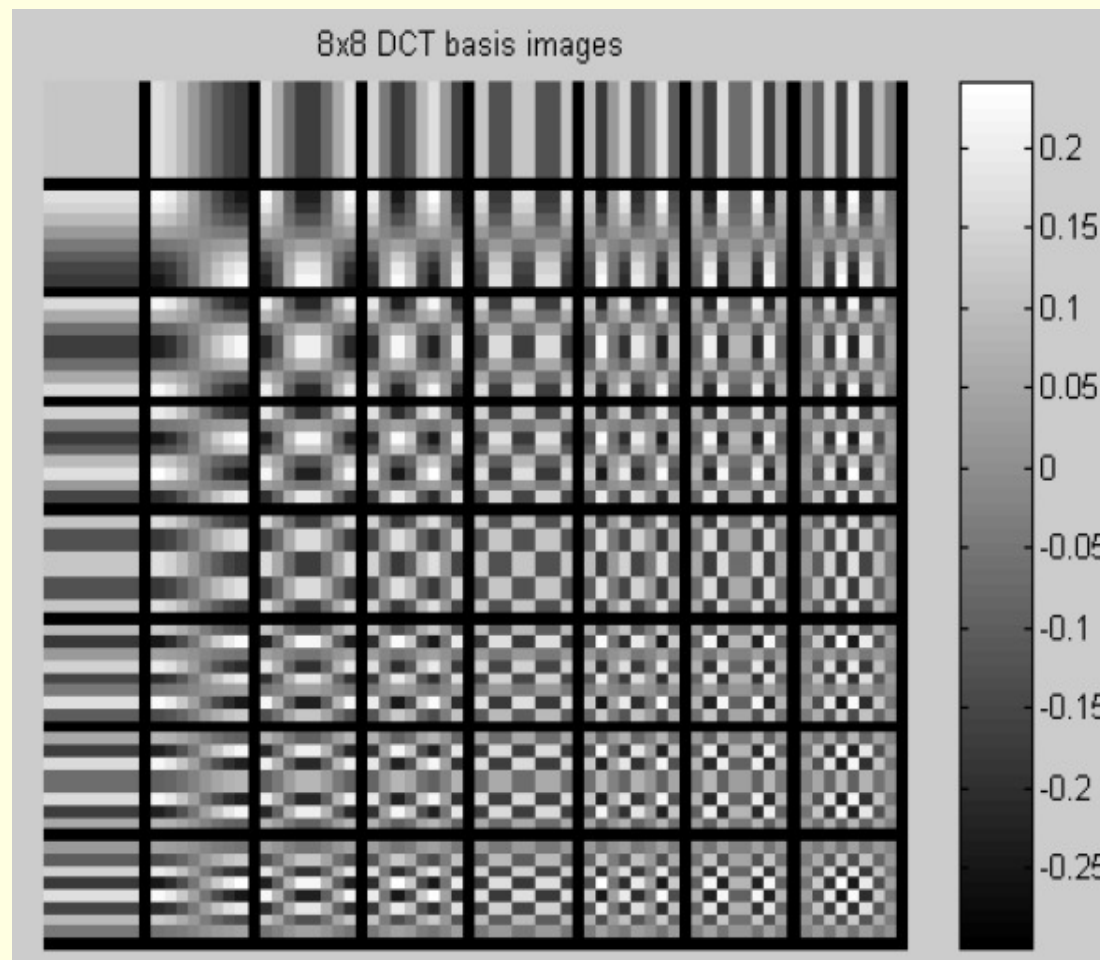
Дискретна косинусова трансформация

■ 1D DCT за $N=8$



Дискретна косинусова трансформація

■ *2D DCT – базові функції*



Уейвлетна трансформация

- ***Wavelet transform***

- Декомпозиция на сигнал в множество базови функции

- базовите функции се наричат ***уейвлети***
- получават се от основен уейвлет чрез мащабиране и изместване

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

където

- a е мащабиращ параметър
- b е изместващ параметър

Уейвлети

■ *Едномерна уейвлетна трансформация*

■ *Непрекъснатата*

■ права

$$\underline{W_f(a,b)} = \int_{-\infty}^{\infty} x(t)\psi_{a,b}(t)dt$$

■ обратна

$$\underline{x(t)} = \frac{1}{C} \int_0^{\infty} \int_{-\infty}^{\infty} W_f(a,b)\psi_{a,b}(t)db \frac{da}{a^2}$$

$$\underline{\text{where } C} = \int_{-\infty}^{\infty} \frac{|\psi(\omega)|^2}{\omega} d\omega < \infty$$

Уейвлети

■ *Едномерна уейвлетна трансформация*

■ *Дискретна*

- преобразува последователност от стойности $\{x_0, x_1, \dots, x_m\}$ в последователност от високо-честотни H_i и ниско-честотни L_i уейвлетни коефициенти

$$H_i = \sum_{m=0}^{k-1} x_{2i-m} \cdot s_m(z)$$

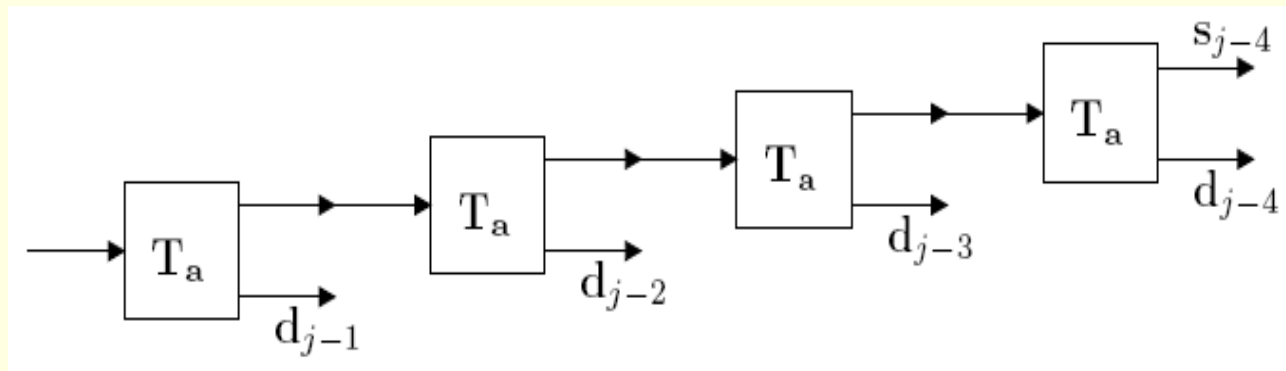
$$L_i = \sum_{m=0}^{k-1} x_{2i-m} \cdot t_m(z)$$

където

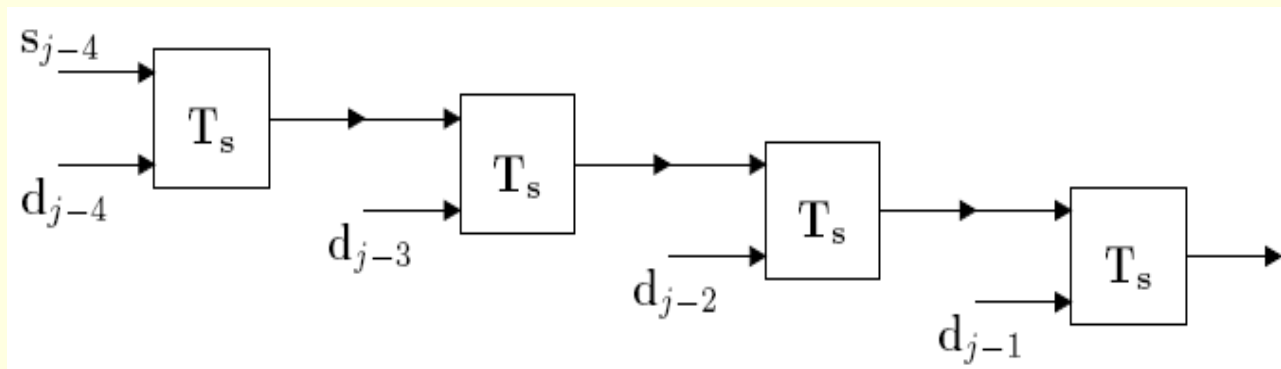
- $s_m(z)$ и $t_m(z)$ са уейвлетни филтри
- K е размера на филтъра
- $i = 1, 2, \dots, (n/2)-1$

Уейвлети

■ Уейвлетна трансформация

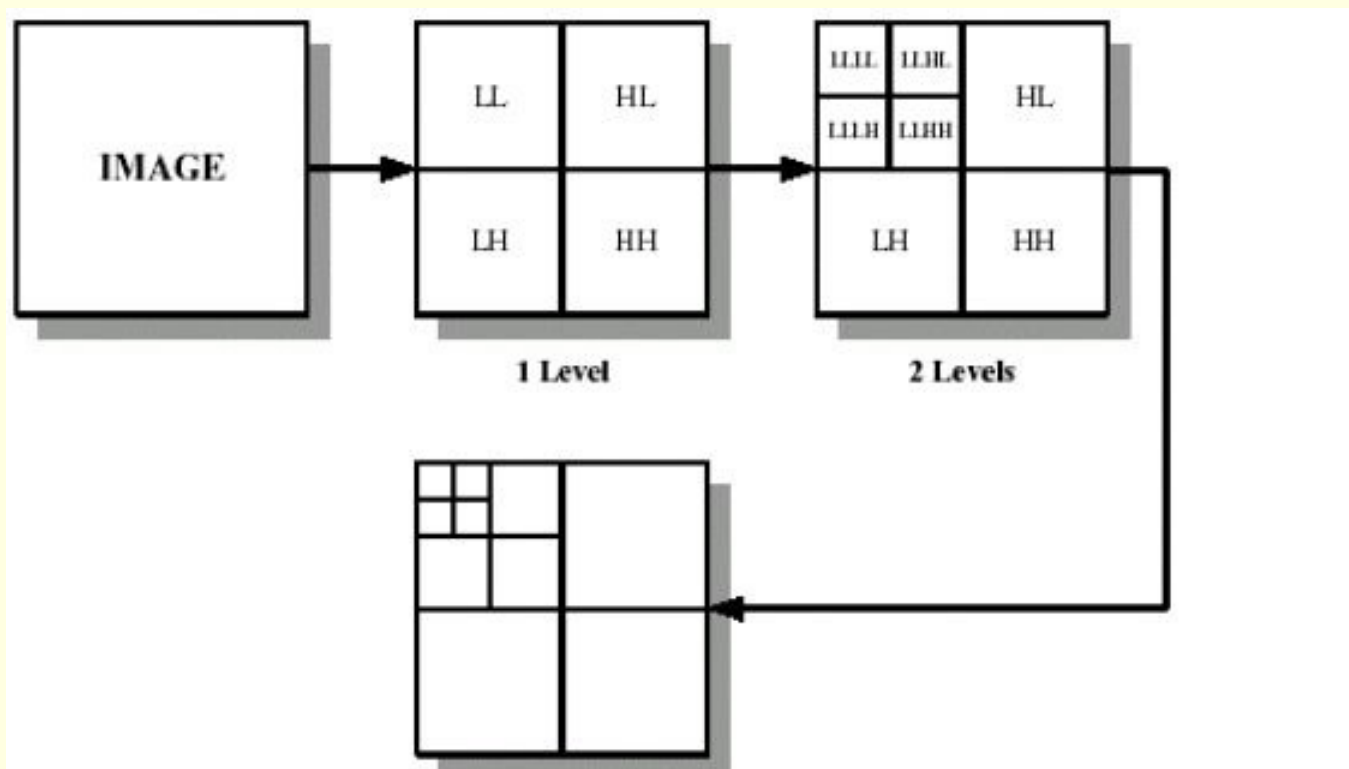


■ Обратна уейвлетна трансформация



Уейвлети

■ *Двумерна уейвлетна трансформация*



Уейвлети

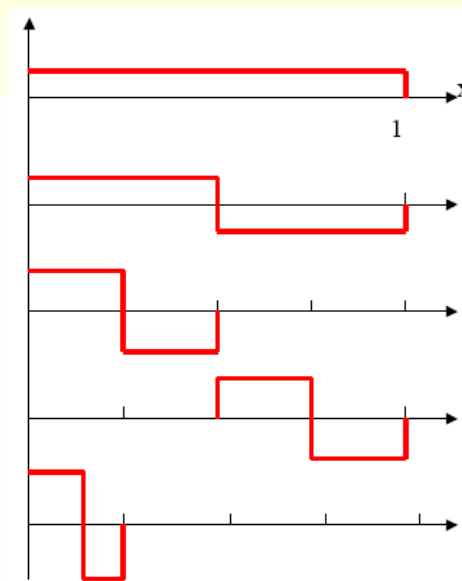
■ Хаар функция

$$k = 2^p + q - 1$$

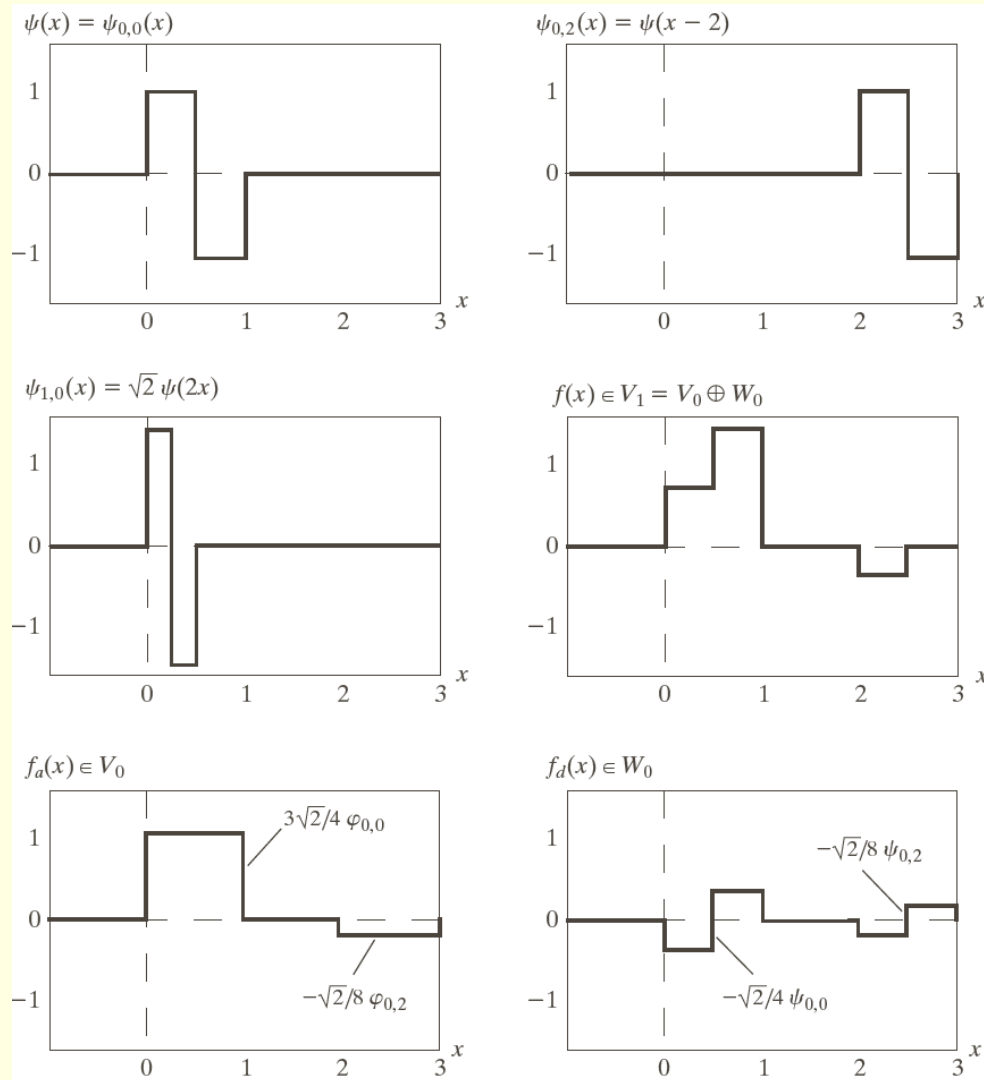
- уникална декомпозиция на цяло число k в двойка (p, q)
 - $k = 0, \dots, N-1, N = 2^n, 0 \leq p \leq n-1$
 - $q = 0, 1$ (за $p=0$)
 - $1 \leq q \leq 2^p$ (за $p>0$)
 - например $k=0 \Leftrightarrow (0,0), k=1 \Leftrightarrow (0,1); k=2 \Leftrightarrow (1,1), k=3 \Leftrightarrow (1,2)$
- $h_k(x) = h_{p,q}(x)$ за $x \in [0,1]$

$$h_0(x) = h_{0,0}(x) = \frac{1}{\sqrt{N}} \text{ for } x \in [0,1]$$

$$h_k(x) = h_{p,q}(x) = \begin{cases} \frac{1}{\sqrt{N}} 2^{p/2} & \text{for } \frac{q-1}{2^p} \leq x < \frac{q-\frac{1}{2}}{2^p} \\ \frac{1}{\sqrt{N}} 2^{p/2} & \text{for } \frac{q-\frac{1}{2}}{2^p} \leq x < \frac{q}{2^p} \\ 0 & \text{for other } x \in [0,1] \end{cases}$$



Уейвлети



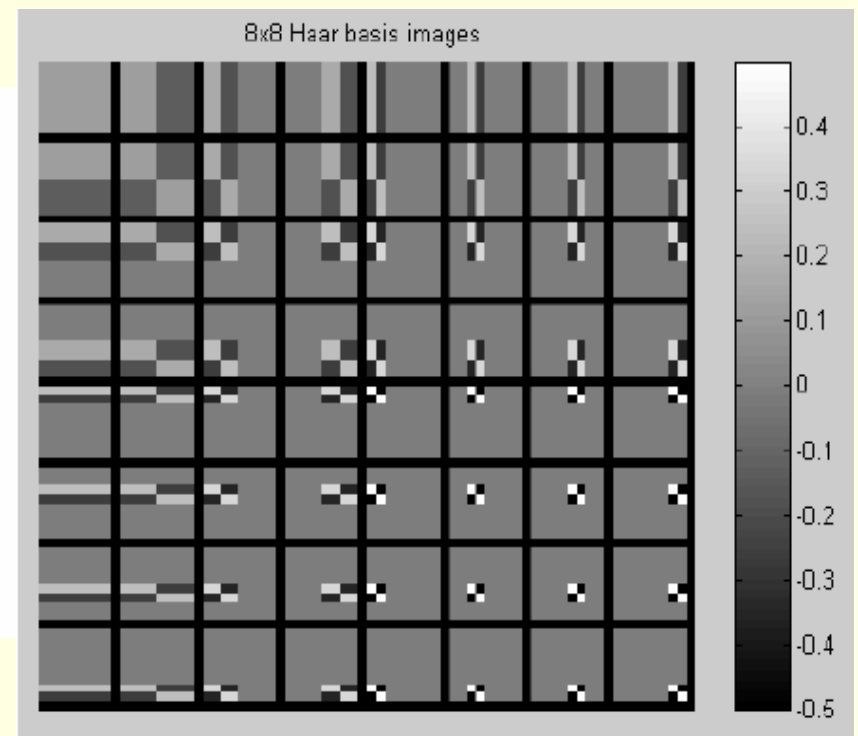
a b
c d
e f

FIGURE 7.14
Haar wavelet functions in W_0 and W_1 .

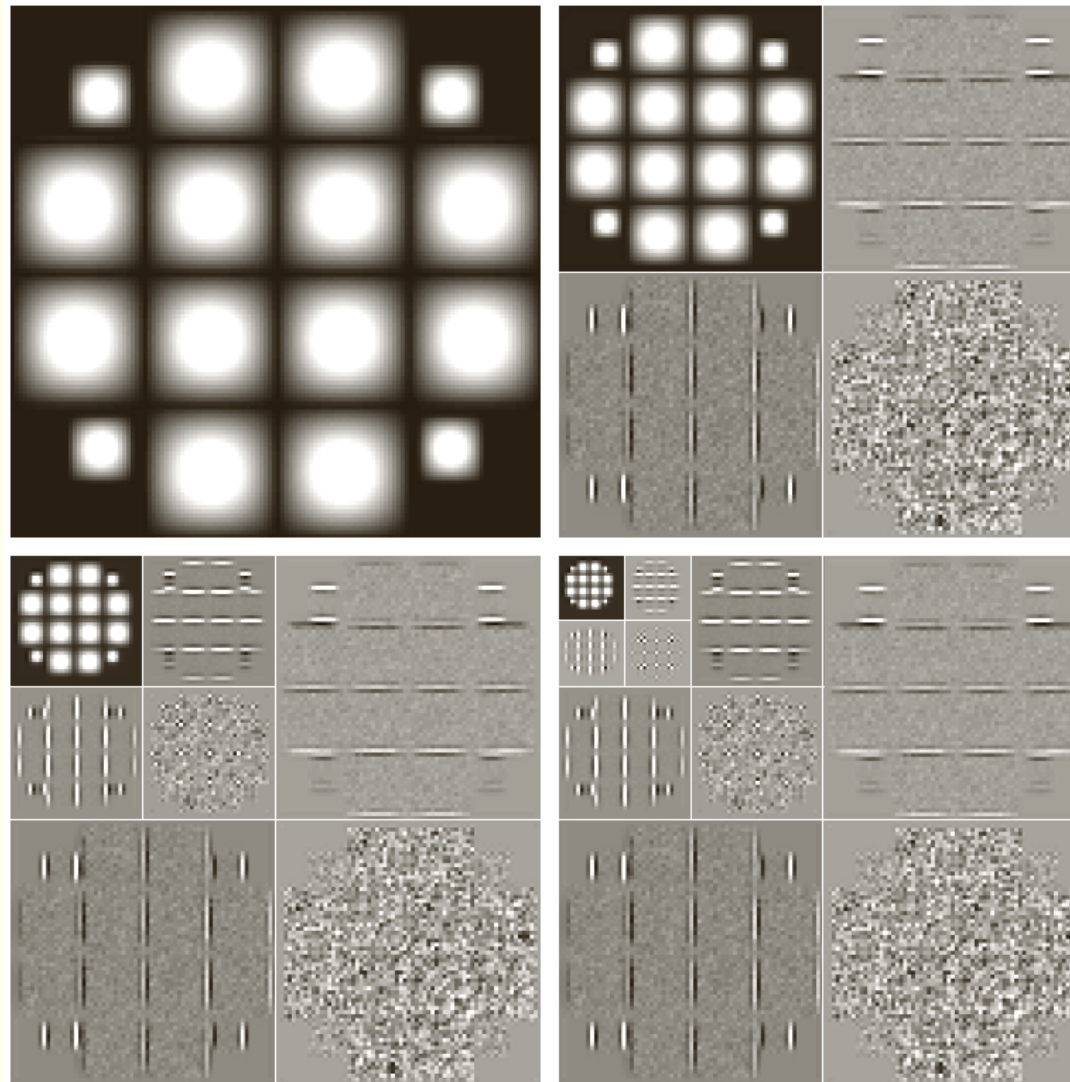
Уейвлети

■ Трансформация на Хаар

$$\frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$



Уейвлети



a	b
c	d

FIGURE 7.25
Computing a 2-D three-scale FWT: (a) the original image; (b) a one-scale FWT; (c) a two-scale FWT; and (d) a three-scale FWT.

Компресия

■ *Необходимост от компресия*

- капацитет за съхраняване на данни
- предаване на данни

■ *Компресия на изображения*

- обикновено са големи
 - съхраняват се в памет с обем $M \times N \times \text{брой канали} \times \text{брой битове}$ за канал
- редуциране на данните за представяне на изображение
 - с премахване на излишната информация

Компресия

- Например изображение с размери 352x288 се представя с 304 128 Bytes
- **Без компресия**
 - видео с 15 кадъра/сек ще се съхрани в 4MB, 1 минута – 260MB, 24 часа – 367GB
- **С компресия**
 - 24 часа видео с 15 кадъра/сек ще изискват 2GB
 - видео за период 187 дни ще изисква същия капацитет за съхраняване, колкото некомпесираната версия за 1 ден

Компресия

■ **Излишък на данни (data redundancy)**

- данни, които не носят нова информация
- данни, които повтарят известна информация

■ **Относителен излишък данни** в едно множество спрямо друго множество данни

- двете множества данни представят една и съща информация

$$R_D = 1 - \frac{1}{C_R} \quad C_R = \frac{n_1}{n_2}$$

където

- n_1 и n_2 са броя единици информация в двете множества
- C_r е **коэффициент на компресия**

Компресия

- Ако $n_2 = n_1$, $C_r = 1$, $R_D = 0$
 - второто множество данни не съдържа излишни данни по отношение на първото множество
- Ако $n_2 \ll n_1$, $C_r \rightarrow \infty$, $R_D = 1$
 - коефициента на компресия е значителен и излишъка на данни е голям
- Ако $n_2 \gg n_1$, $C_r \rightarrow 0$, $R_D \rightarrow -\infty$
 - второто множество съдържа много повече данни от оригиналното
 - нежелателен случай на разширяване, а не компресиране на данните
- **Обикновено $C_r \in (0, \infty)$ и $R_D \in (-\infty, 1)$**
- Коефициент на компресия 10 (или 10:1) означава
 - оригиналното множество данни съдържа 10 инф. единици (например байт) за всяка инф. единица от второто множество данни
 - излишъка на данни е 0.9, т.е 90% от данните в първото множество са излишни

Компресия

■ *Компресия и декомпресия на данни*



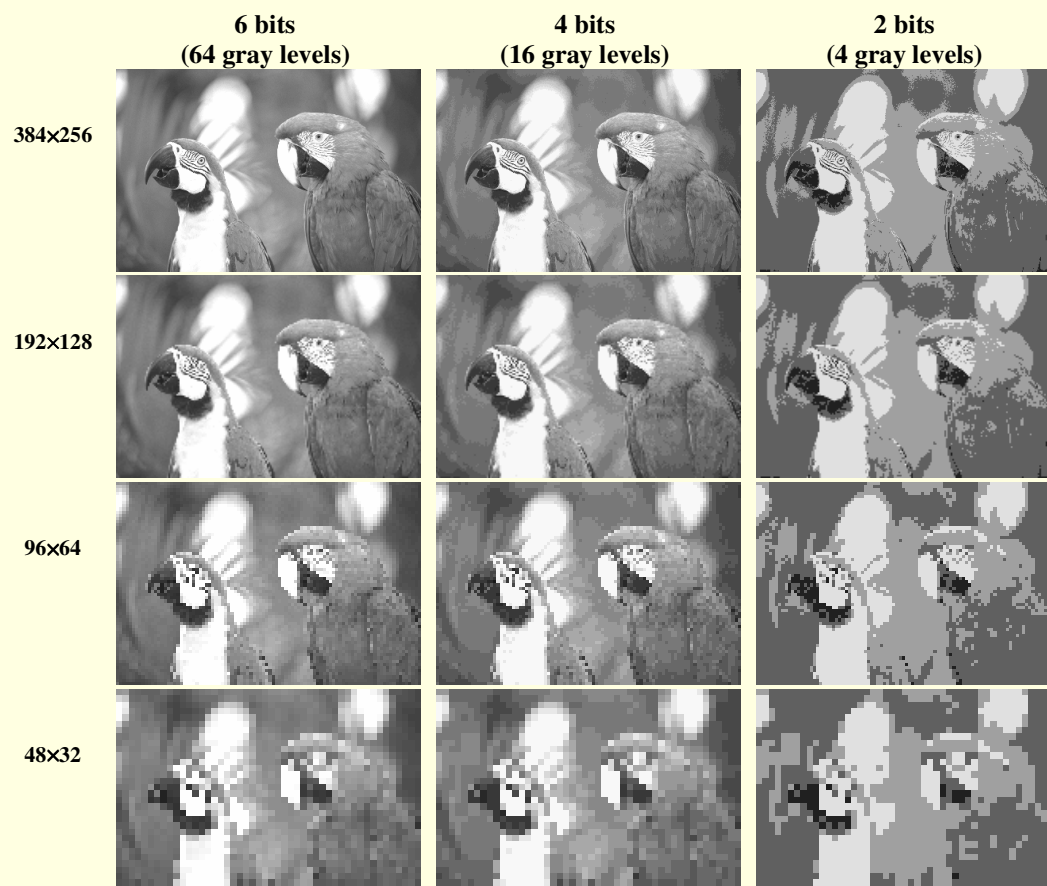
Параметри на изображение

■ *Пространствена дискретизация*

- размери на изображението

■ *Квантоване по нива*

- брой стойности или битове за съхраняване на стойността на всеки пиксел



Стеганография

■ Приложение на квантоването

- изображение се преобразува от 8 bit/pixel/band в 6 bit/pixel/band
- разликите са незабележими



оригинално
изображение

8 бита за пиксел

3 канала (RGB)

Стеганография

■ Приложение на квантуването

- изображение се преобразува от 8 bit/pixel/band в 6 bit/pixel/band
- разликите са незабележими



оригинално
изображение

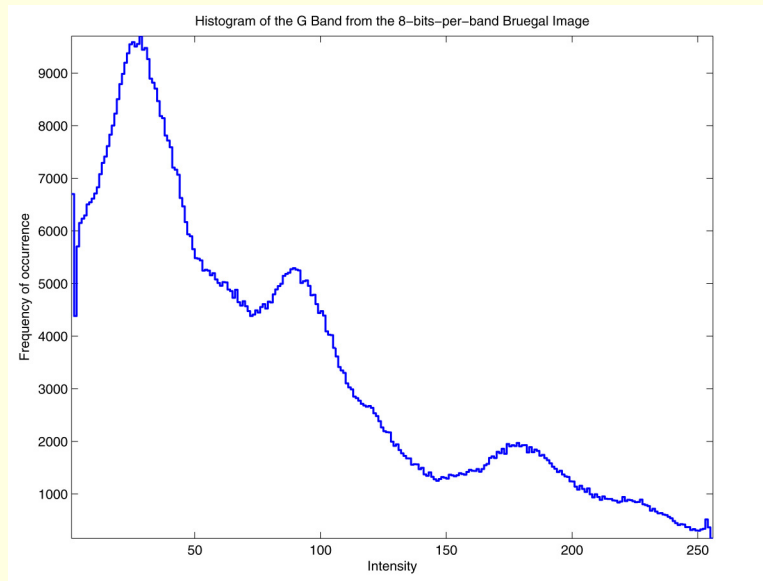
6 бита за пиксел

3 канала (RGB)

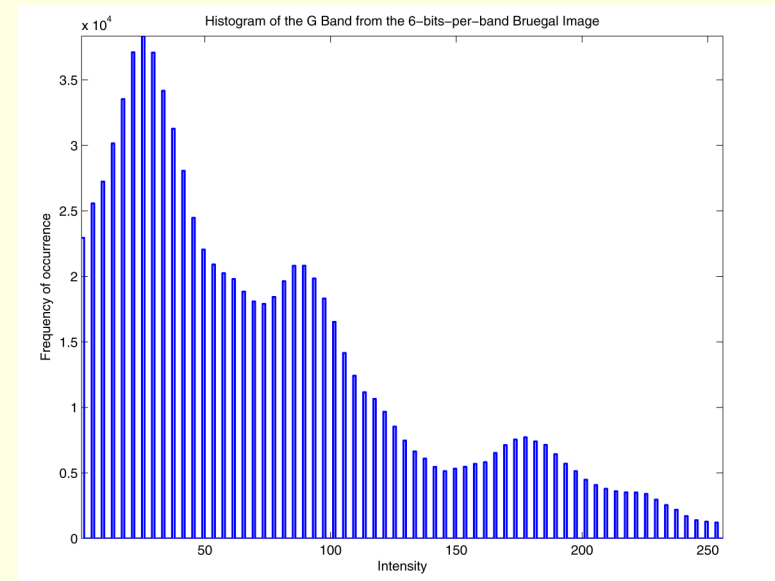
Стеганография

- Ако 6 битовото изображение се представи като 8 битово, то ще съдържа само стойностите 0, 4, 8, ... , 252

хистограма на G-канала за
8-битовото изображение

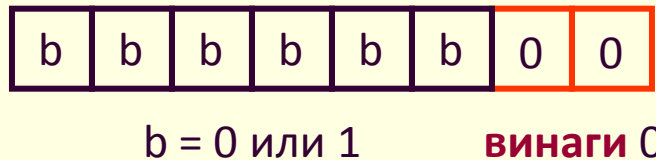


хистограма на G-канала за
6-битовото изображение



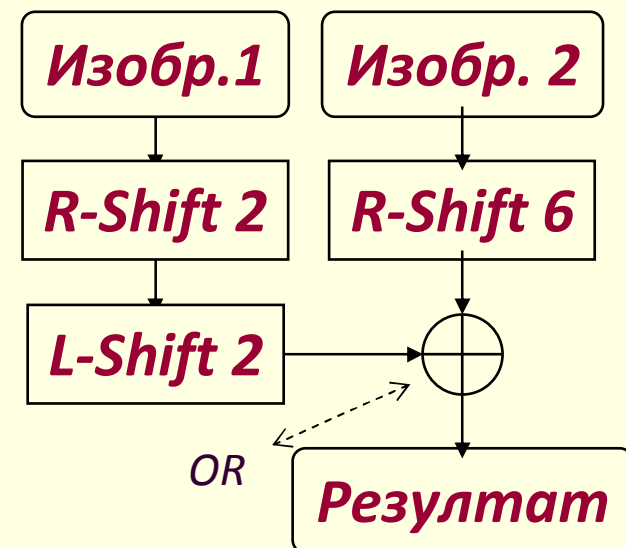
Стеганография

Ако 6 битовото изображение се представи като 8 битово, то 8 битовите стойности ще имат нули в младшите 2 бита



Това позволява друга информация да се кодира в младшите битове на всеки байт

Другата информация може да бъде текст или друго изображение



X-Shift n = изместване наляво (L) или надясно (R) с n бита

Стеганография

- Второто изображение е “невидимо”
 - стойностите на пикселите му са между 0 и 3
 - стойността на всеки пиксел от 2^{то} изображение се добавя към стойността на съответния пиксел от 1^{то} изображение, която е от множеството {0, 4, ... , 252}
- Второто изображение е “шум” за първото изображение

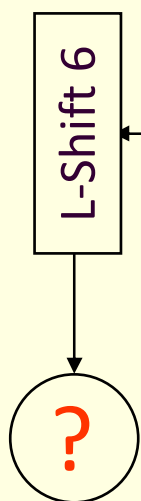


Pieter Bruegel (the Elder, ca. 1525-69), *The Peasant Dance*, 1568, Oil on oak panel, 114x164 cm, Kunsthistorisches Museum Wien, Vienna

Стеганография

- Как да възстановим второто изображение от резултата
 - 2^{то} изображение се съдържа в младшите 2 бита на всеки пиксел за всеки канал
 - изместване наляво с 6 бита възстановяват второто изображение

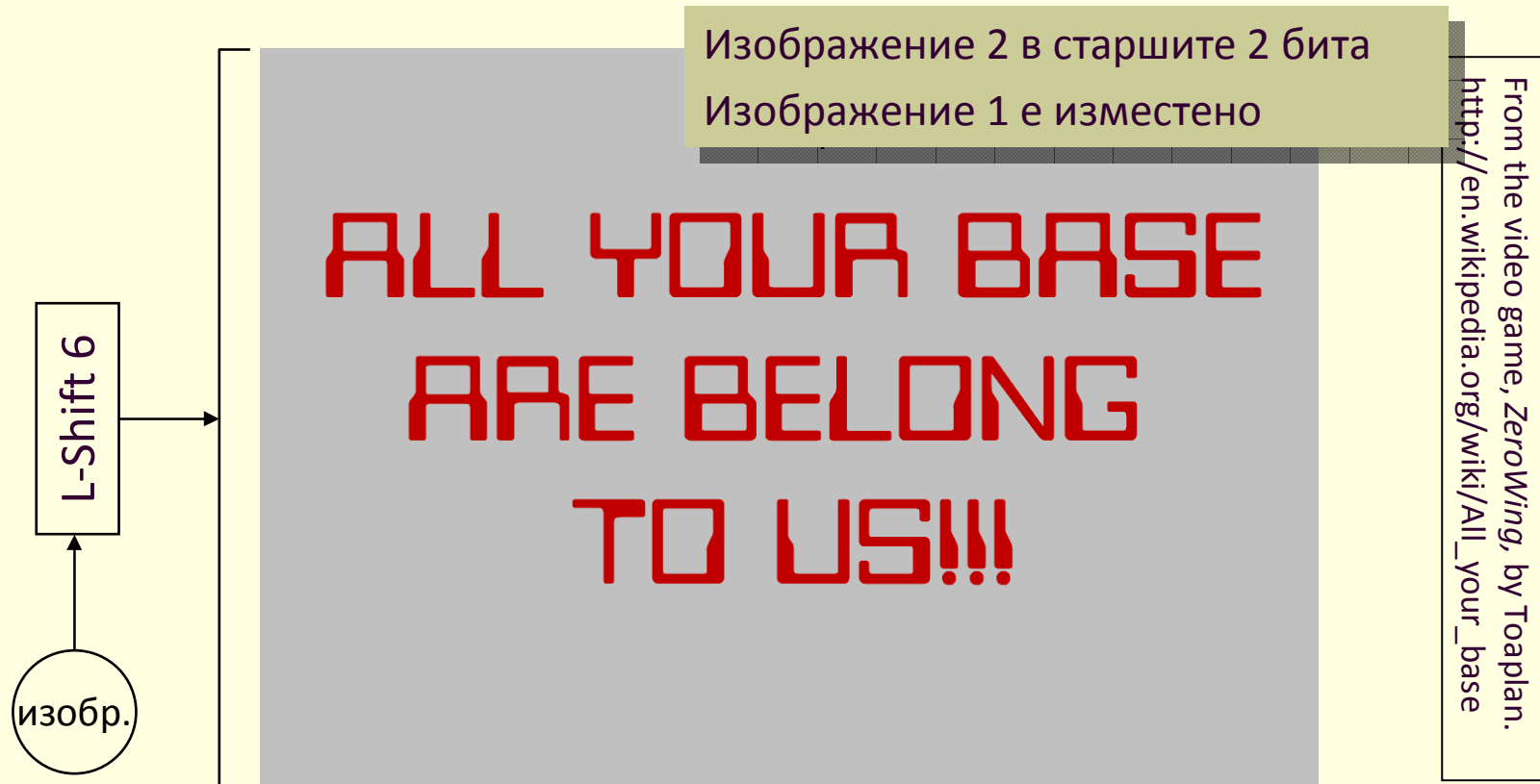
Изображение 1 в старшите 6 бита
Изображение 2 в младшите 2 бита



Pieter Bruegel (the Elder, ca. 1525-69), *The Peasant Dance*, 1568, Oil on oak panel, 114x164 cm, Kunsthistorisches Museum Wien, Vienna

Стеганография

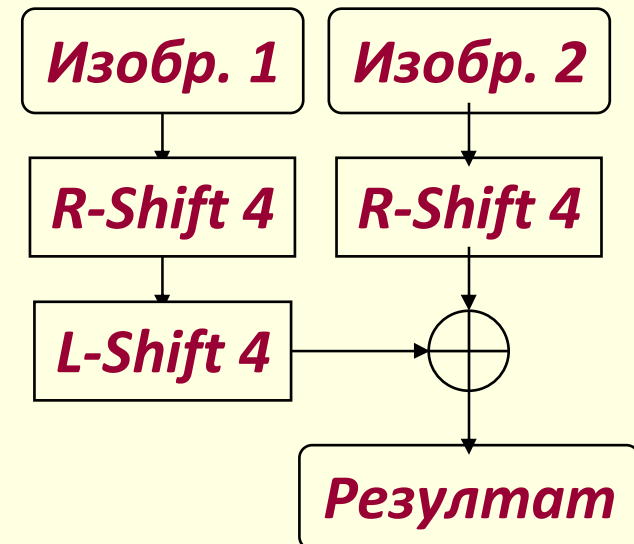
- Как да възстановим второто изображение от резултата
 - 2^{то} изображение се съдържа в младшите 2 бита на всеки пиксел за всеки канал
 - изместване наляво с 6 бита възстановяват второто изображение



Стеганография

- Две изображения представени с 4 бита/пиксел/канал могат да се комбинират
- При това само изображението в старшите битове ще се “вижда”
- двете изображения съдържат еднакво количество информация
 - едното има стойности между 0 и 15
 - другото има стойности в интервала {16, 32, 48, ... , 240}
 - стойностите се сумират
 - изображението в младшите битове е на практика невидимо

Двете изображения са представени с 4 бита/пиксел/канал преди да се комбинират



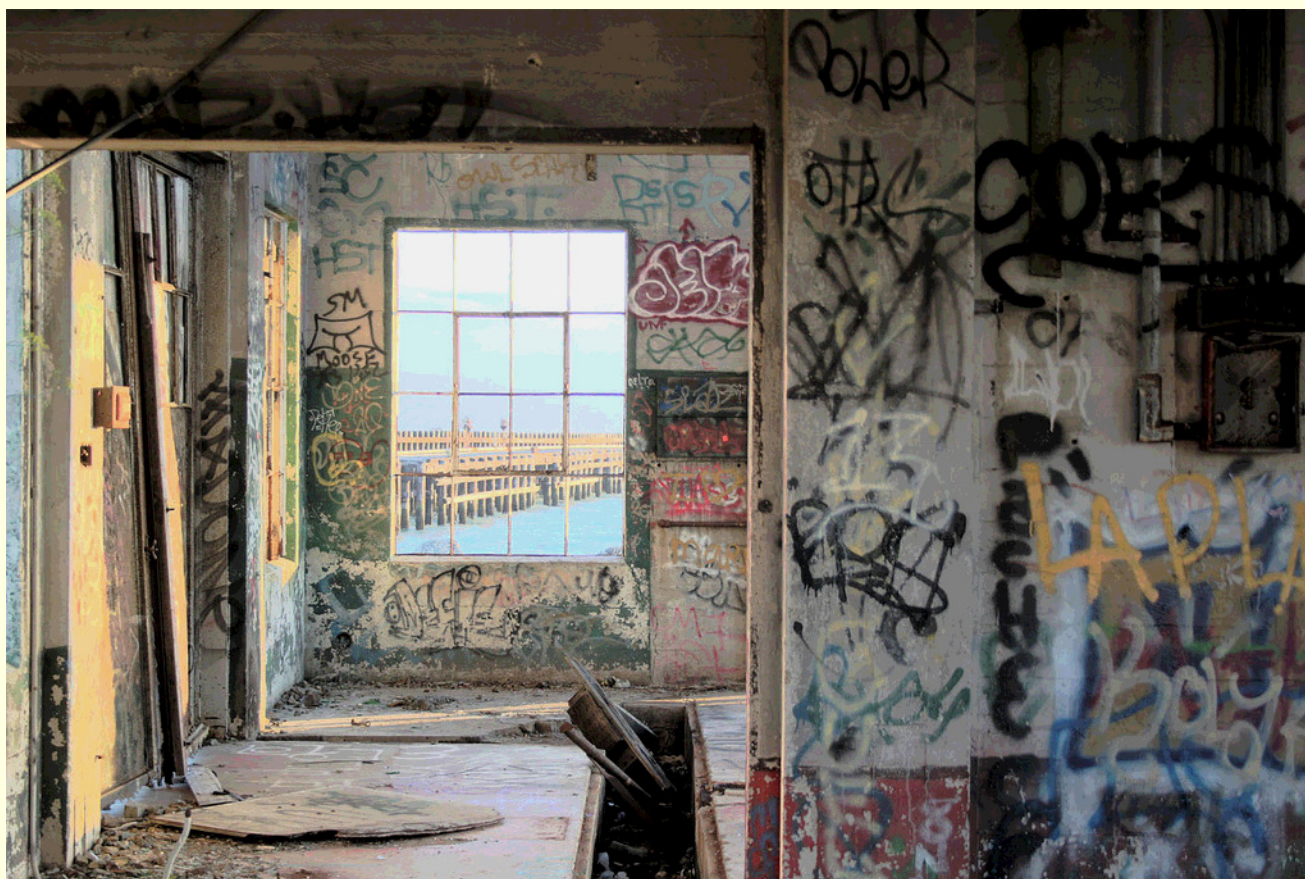
Стеганография

Изображение 1



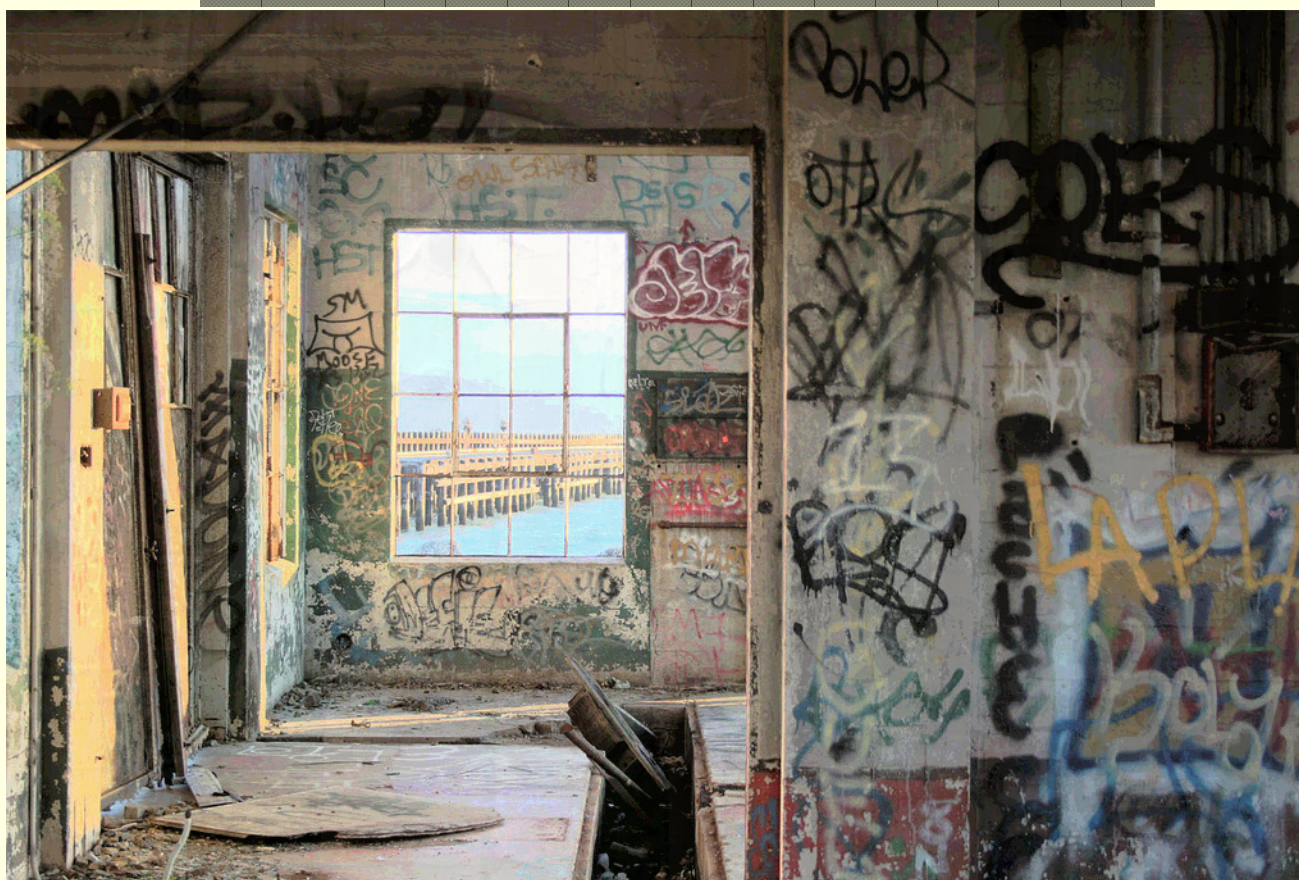
Стеганография

Изображение 1 дискретизировано в 4 бита



Стеганография

Изображение 1 в старшите 4 бита
Изображение 2 в младшите 4 бита



Стеганография

Изображение 2 в старшите 4 бита
Изображение 1 изместено



Принципи на компресия

- Преобразуване на оригиналното множество данни в некорелирано множество данни
- Източници на излишък на данни
 - *Статистически излишък*
 - *Визуално-физиологичен излишък*
- Компресията се постига с редуциране или елиминиране на един или няколко от източниците на излишни данни

Статистически излишък

■ **Пространствена корелация**

- излишък на данни в стойностите на съседни пиксели
 - *interpixel redundancy*

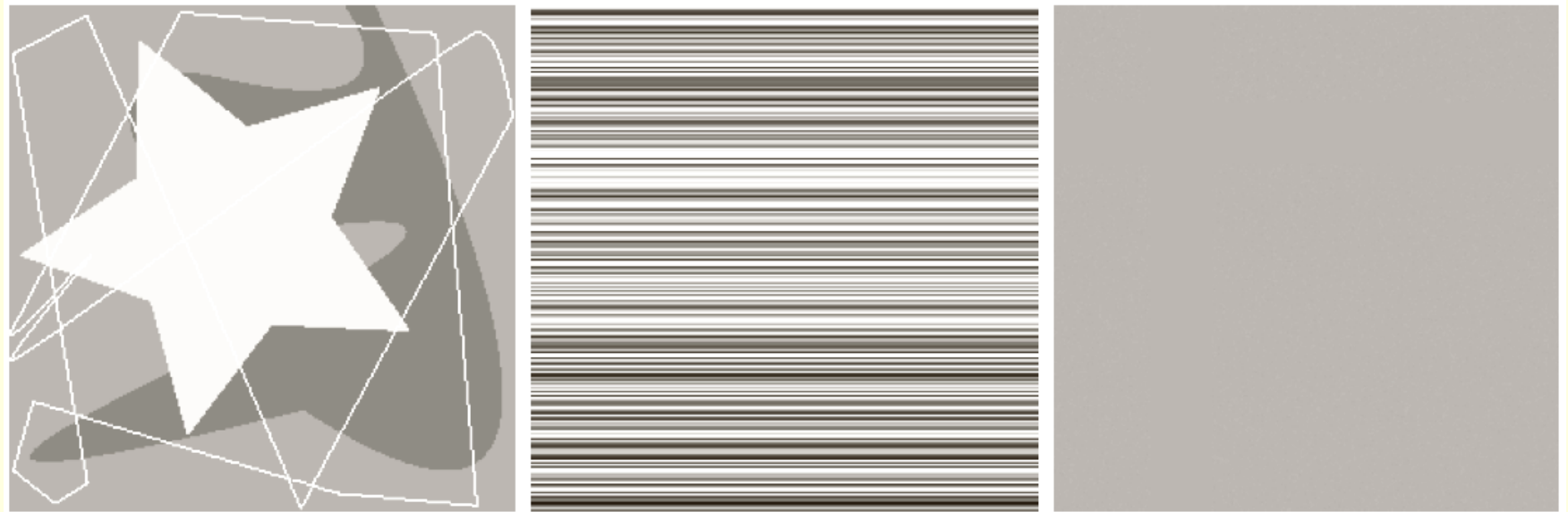
■ **Спектрална корелация**

- излишък на данни в различните цветови равнини
 - *coding redundancy*

■ **Времева корелация**

- излишък на данни в съседни кадри на последователност от изображения
 - *temporal redundancy*

Принципи на компресия



a b c

FIGURE 8.1 Computer generated $256 \times 256 \times 8$ bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

Визуално-физиологичен излишък

- ***Psychovisual redundancy***

- Възприеманата от човешката визуална система информация зависи от много фактори

- Различните видове визуална информация при различни условия се възприемат по-различен начин

- част от визуалната информация има по-малка важност от друга
 - именно тази информация е визуално-физиологично излишна
 - такава информация може да бъде премахната без това да промени съществено възприемането на изображението

Визуално-физиологичен излишък

■ ***Интензитет/контраст***

- човешката визуална система има праг за възприемане на светъл обект върху светъл фон

■ ***Текстура***

- дискриминационните способности се увеличават с увеличаване на детайлите

■ ***Frequency sensitivity***

- дискриминационните способности се увеличават с увеличаване на честотата

■ ***Color sensitivity***

- човешката визуална система е много по-чувствителна към яркостта отколкото към цветността

■ ***Temporal masking***

- човешката визуална система не е чувствителна към детайлите при голямо изменение на сцената

Видове компресия

■ ***Без загуби (lossless compression)***

- позволява възстановяване на изображението без грешка (разлика) на оригиналните стойности

■ ***Със загуби (lossy compression)***

- възстановеното изображение се различава от оригиналното

■ ***При компресия на изображения***

- рядко е необходима компресия без загуби
- компресираното изображение не трябва да се различава ***МНОГО*** от оригиналното

Параметри за оценяване

■ **Bitrate** $\frac{\text{размер на компресирания файл}}{\text{брой пиксели в изображението}} = \frac{C}{N}$ бит/пиксел

■ **Коефициент на компресия**

$$\frac{\text{размер на оригиналния файл}}{\text{размер на компресирания файл}} = \frac{N \cdot k}{C}$$

■ **Субективни критерии**

■ 5 степенна скала за оценяване на Bell Labs

1. Разликата след компресия не се забелязва
2. Разликата след компресия малко се забелязва
3. Разликата след компресия ясно се забелязва, но не е неприятна
4. Разликата след компресия е неприятна и нежелателна
5. Разликата след компресия е изключително неприятна и нежелателна

Параметри за оценяване

- **Mean average error (MAE)**
$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - x_i|$$

- **Mean square error (MSE)**
$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - x_i)^2$$

- **Signal-to-noise ratio (SNR)** [децибели]

$$\text{SNR} = 10 \cdot \log_{10} [\sigma^2 / \text{MSE}]$$

- **Pulse-signal-to-noise ratio (PSNR)** [децибели]

$$\text{PSNR} = 10 \cdot \log_{10} [A^2 / \text{MSE}]$$

- A е амплитудата на сигнала: $A = 2^8 - 1 = 255$ за 8-битов сигнал

Излишък при кодиране

- Хистограма на изображение

- вероятността за поява на стойност r_k е $p_r(r_k)$

$$p_r(r_k) = \frac{n_k}{n}, \quad k = 0, 1, 2, \dots, L-1$$

- където

- L е броя полутонови стойности в изображението
- n_k е броя срещания на полутоновата стойност k
- n е общия брой пиксели в изображението

- Стойността r_k се представя с $l(r_k)$ бита

- Всеки пиксел се представя средно с L_{avg} бита

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

- Изображение с размери $M \times N$ се представя с $M \times N \times L_{avg}$ бита

Излишък при кодиране

- **Информационно съдържание**

$$I = \log_2 \frac{1}{n} \text{ бита}$$

- **Ентропия**

- **среден брой битове за стойност**

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

- **мярка за случайността**
- **ентропията е максимална когато всички стойности са еднакво вероятни**

- **Минималният брой битове за представяне на дадена стойност в определен информационен източник е равна на ентропията**

Излишък при кодиране

■ *Пример*

- Изображение съдържа 8 полутонове стойности
- $l_1(r_k)$ е естественото бинарно кодиране
 - m -битова последователност от стойности (2^m)
 - за кодиране на 8 различни стойности $m = 3$
- Всеки пиксел се представя с 3 бита
 - $Ll_{avg} = 3$ бита

Излишък при кодиране

■ Пример

Table 6.1 Variable-Length Coding Example

r_k	$p_k(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

- ако се използва код $l_2(r_k)$

$$L2_{avg} = \sum_k l_2(r_k) \cdot p_k(r_k) =$$

$$2 \times (0.19) + 2 \times (0.25) + 2 \times (0.21) + 3 \times (0.15) + 4 \times (0.08) + 5 \times (0.06) + 6 \times (0.03) + 7 \times (0.02)$$

- $L2_{avg} = 2.7$ бита

Излишък при кодиране

■ Пример

- $L1_{avg} = 3$ бита

- $L2_{avg} = 2.7$ бита

- $C_r = L1_{avg}/L2_{avg} = 3/2.7 = 1.1$

- $R_d = 1 - 1/1.1 = 0.099$

- Приблизително 10% от данните в $L1$ са излишни

Кодиране с променлива дължина

- **Variable Length Coding**
- Използва различен брой битове за кодиране на различните стойности
 - по-малко битове се използват за по-вероятните стойности
- Компресия без загуби
- Недостатъци
 - еднакво вероятни стойности
 - няма полза от компресирането
 - как да се определят стойностите за прекодиране

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

TABLE 8.1
Example of
variable-length
coding.

Huffman кодиране

- ***Huffman Encoding***
- Използва най-малкия възможен брой кодове за стойностите
 - кодирането е с променлива дължина
- Оптимален подход при фиксиран брой стойности
- Стойностите се кодират една по една
 - подреждат се оригиналните стойности според вероятността за появата им в информационния източник
 - в няколко итерации се редуцира информацията чрез обединяване на стойности
 - най-малко вероятния символ се кодира с 1 битов код
 - останалите – с уникални кодове с нарастваща дължина
- Декодирането е уникално

Huffman кодиране – редуциране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4				
a6	0.3				
a1	0.1				
a4	0.1				
a3	0.06				
a5	0.04				

Huffman кодиране – редуциране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4			
a6	0.3	0.3			
a1	0.1	0.1			
a4	0.1	0.1			
a3	0.06	0.1			
a5	0.04				

Huffman кодиране – редуциране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4	0.4		
a6	0.3	0.3	0.3		
a1	0.1	0.1	0.2		
a4	0.1	0.1			
a3	0.06	0.1			
a5	0.04				

The diagram illustrates the reduction step of Huffman coding. It shows a table of original and reduced values for symbols a1 through a6. Arrows indicate the merging of nodes: a3 (0.06) and a5 (0.04) merge into a new node with probability 0.1; a1 (0.1) and a4 (0.1) merge into a new node with probability 0.2. The value 0.2 is highlighted in a red box.

Huffman кодиране – редуциране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4	0.4	0.4	
a6	0.3	0.3	0.3	0.3	
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman кодиране – редуциране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6
a6	0.3	0.3	0.3	0.3	0.4
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman кодиране – кодиране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4	0.4	0.4	0.6 0
a6	0.3	0.3	0.3	0.3	0.4 1
a1	0.1	0.1	0.2	0.3	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman кодиране – кодиране

оригинални стойности		редуцирани стойности			
символ	вероятност	1	2	3	4
a2	0.4	0.4	0.4	0.4 1	0.6 0
a6	0.3	0.3	0.3	0.3 00	0.4 1
a1	0.1	0.1	0.2	0.3 01	
a4	0.1	0.1	0.1		
a3	0.06	0.1			
a5	0.04				

Huffman кодиране – кодиране

оригинални стойности		редуцирани стойности				
символ	вер. код	1	2	3	4	
a2	0.4	1	0.4 1	0.4 1	0.4 1	0.6 0 0.4 1
a6	0.3	00	0.3 00	0.3 00	0.3 00	
a1	0.1	011	0.1 011	0.2 010	0.3 01	
a4	0.1	0100	0.1 0100	0.1 011		
a3	0.06	01010	0.1 0101			
a5	0.04	01011				

Huffman кодиране

- *Модификации и подобрения*
 - *блоково кодиране*
 - кодират се групи стойности
 - *аритметично кодиране*
 - *LZW кодиране*

LZW кодиране

- ***Lernpel-Ziv-Welch coding***
- Използва кодове с фиксирана дължина за последователности с променлива дължина
- ***Речник*** (codebook, dictionary)
 - съдържа стойностите, които ще се кодират
 - за 8 битово изображение първите 256 думи в речника са полутоновите стойности на пикселите 0, 1, 2, ... ,255
 - след това се сканира изображението и към речника се добавят последователности от стойности и позицията в речника определя кода
 - например ако първите два пиксела в изображението са бели последователността "255-255" се записва на позиция 256 в речника (първата свободна след 255)

LZW кодиране

- Речникът се изгражда едновременно с кодирането на изображението
 - когато дадена последователност вече е записана в речника, то тя се заменя с кода
 - например ако речникът се съхранява в 9 бита (512 думи), вместо 16 бита за "255-255" се използва 9 битов код
- Размерът на речника е важен параметър
 - ако е много малък последователностите, които се кодират са малко
 - малък коефициент на компресия
 - ако е много голям кода също е голям
 - кодирането може да се окаже неефективно

LZW кодиране

оригинално изображение

```

39  39  126  126
39  39  126  126
39  39  126  126
39  39  126  126
    
```

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

речник преди началото на кодирането

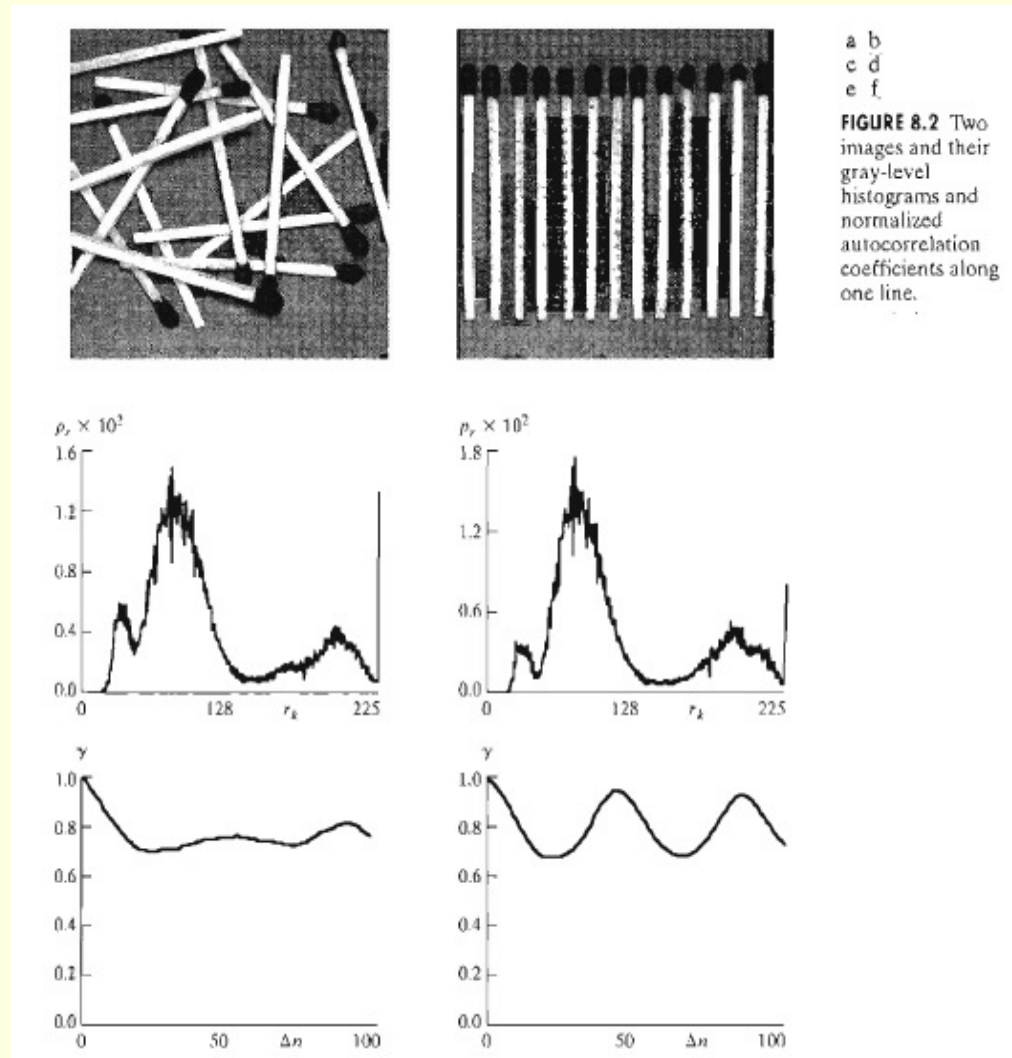
Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

попълване на речника

Пространствен излишък

- ***Spatial Redundancy, Geometric Redundancy, Interframe Redundancy, linterpixel Redundancy***
- Стойността на даден пиксел може да бъде определена по стойностите на съседните пиксели
- Представянето на изображението като двумерен масив от стойности на пиксели се преобразува в друго, не-визуално представяне
 - ***tapping***
 - преобразуването е обратимо ако оригиналните стойности могат да се получат от трансформираните стойности

Пространствен излишък



Run-length encoding

- Всеки ред от изображението се кодира като последователност от 0 и 1
 - последователност: '0', '10', '110', '1110', ...
 - run-length: 0 1 2 3

- Алтернативен вариант

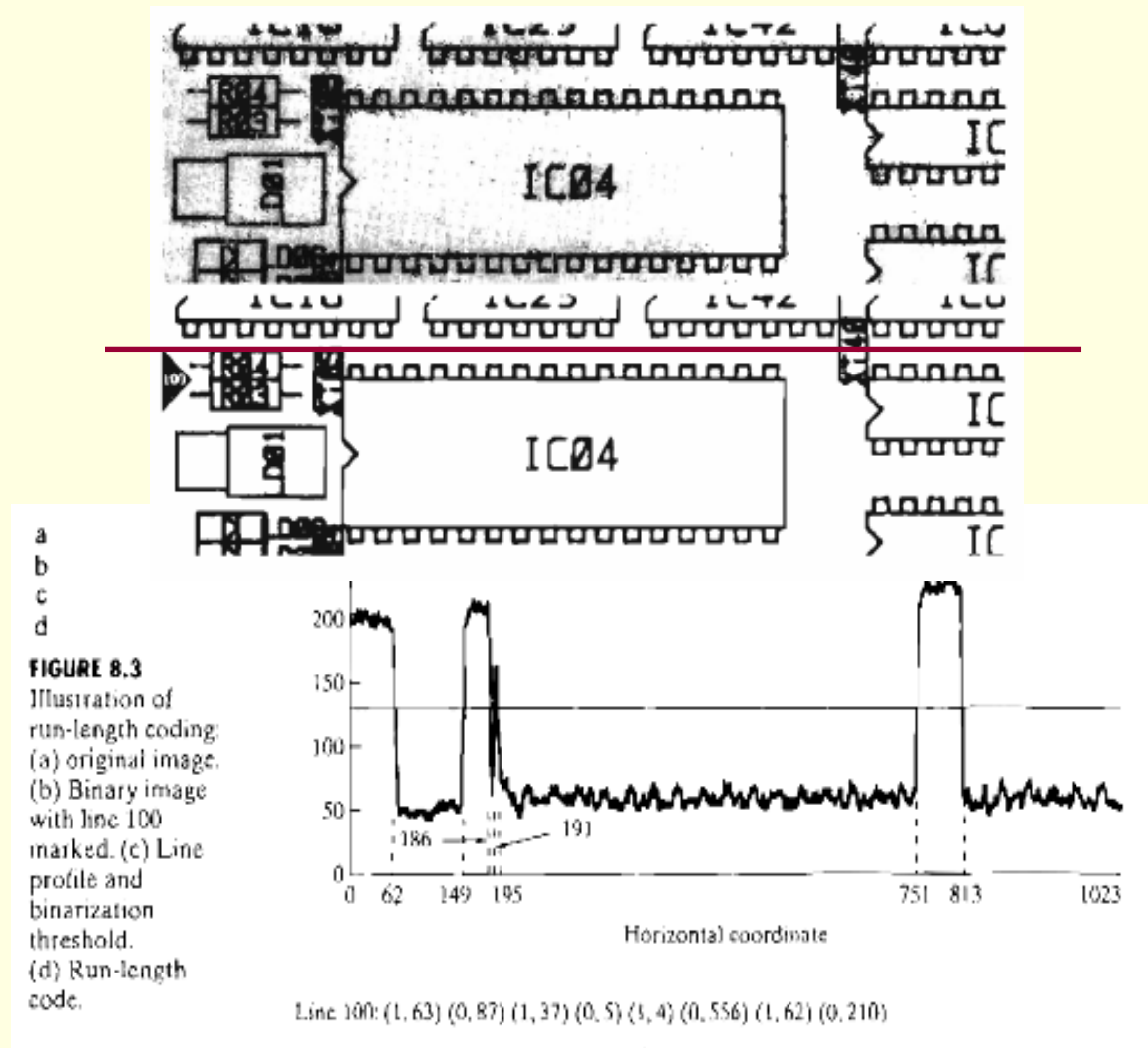
- кодиране с последователност от двойки

$$(g_1, r_1) (g_2, r_2) \dots (g_i, r_i) \dots (g_n, r_n)$$

- където

- g_i е стойността (0 или 1)
- r_i е броя пиксели със стойност g_i

Run-length encoding



Run-length encoding

- *Пример*
- Оригинално изображение
 - размери 1024×343 пиксела (по 8 bits/pixel)
- Бинаризирано изображение
 - след прагова операция
 - размери 1024×343 пиксела (по 1 bit/pixel)
- Run-length encoding на изображението
 - необходими са 12 166 runs
 - всеки run се кодира с 11 bits

Run-length encoding

- *Пример*
- Бинаризирано изображение
 - $N_1 = 1024 \times 343 \times 1$
- Изображение с run length encoding
 - $N_2 = 12\ 166 \times 11$
- $C_r = (1024 \times 343) / (12\ 166 \times 11) = 2.63$
- $R_d = 1 - 1/2.63 = 0.62$
- *Излишната информация в бинаризираното изображение е около 62%*

Визуално-физиологичен излишък

- Премахването на визуално-физиологичен излишък
 - не влияе на реално възприеманата визуална информация
 - променя количествено информацията
- Компресията, която се базира на този подход, е със загуби
 - нарича се **quantification**
 - заради загуба в количествената оценка на информацията (quantifiable information)

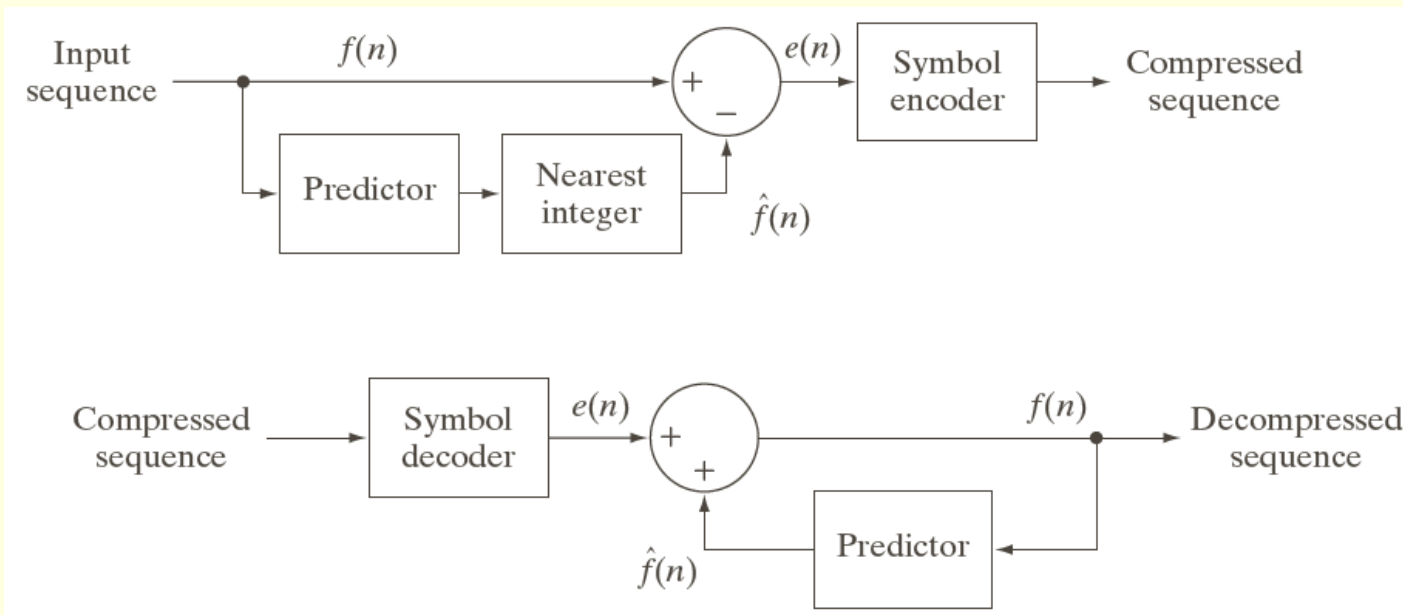
Компресия с квантуване

- Необратим процес
 - загуби на източника на информация
- Използва се често за компресия на изображения и видео
- Два типа компресия с квантуване
 - ***с предсказване***
 - намалява вариациите на кодирания сигнал
 - ***с трансформация***
 - намалява корелацията на стойностите

Компресия с квантуване

■ *Компресия с предсказване (differential coding)*

- кодира се не директно сигнала, а разликата между стойностите му и техните оценки
 - оценките (предсказани стойности) се определят като разлика в оригиналните стойности

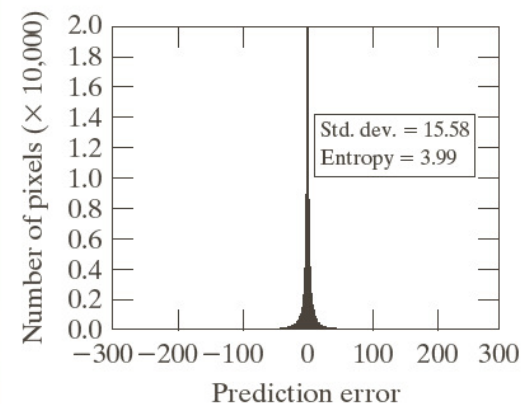
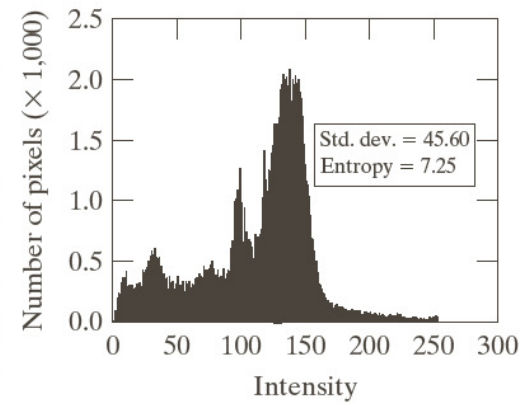
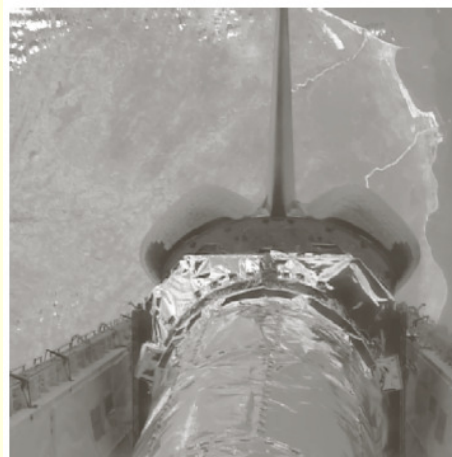


a
b

FIGURE 8.33
A lossless predictive coding model:
(a) encoder;
(b) decoder.

Компресия с квантуване

■ *Компресия с предсказване*



a b
c d

FIGURE 8.34 (a) A view of the Earth from an orbiting space shuttle. (b) The intensity histogram of (a). (c) The prediction error image resulting from Eq. (8.2-34). (d) A histogram of the prediction error. (Original image courtesy of NASA.)

Компресия с квантуване

■ **Компресия с трансформация (*transform coding*)**

■ **компресия**

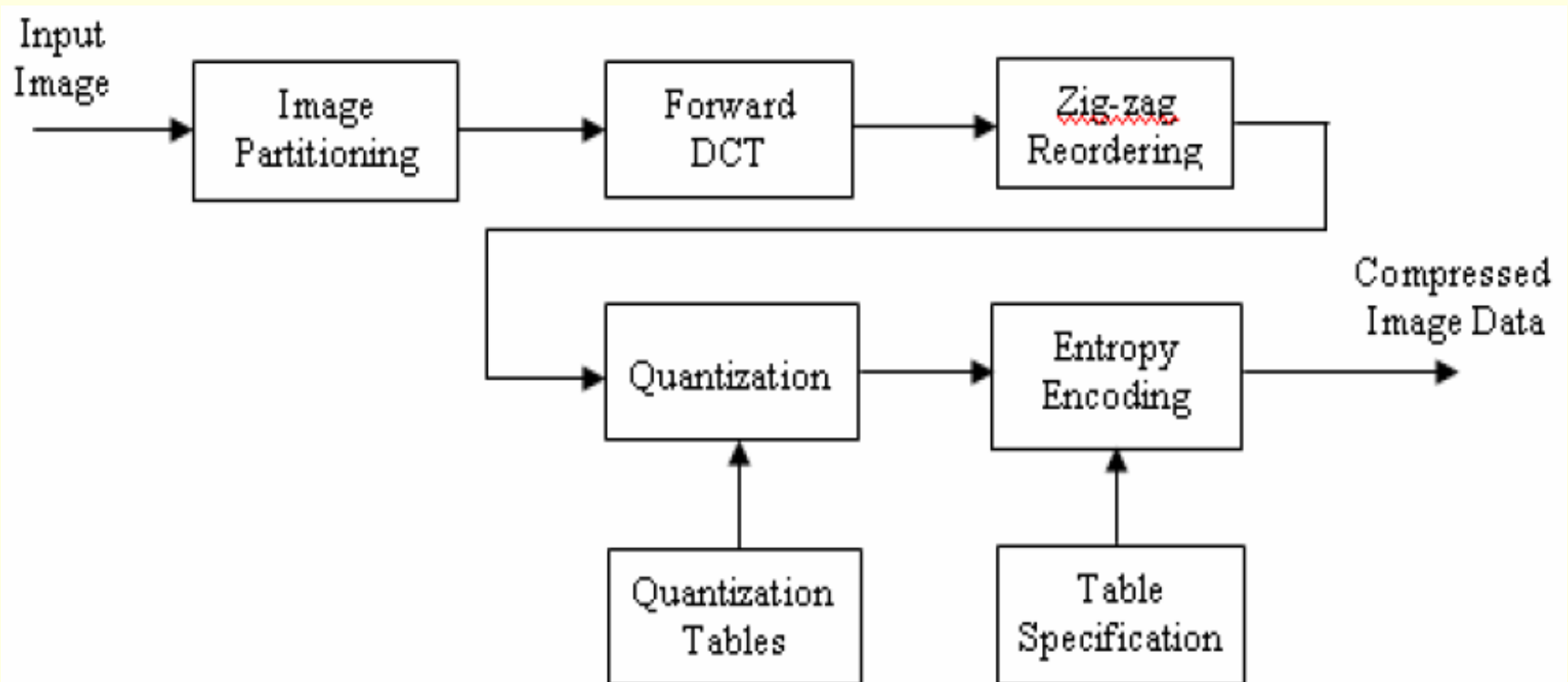
- преди кодирането изображението се разделя на блокове и се прилага права трансформация
 - Дискретни трансформации на Фурие, Уолш, Адамар
 - DCT
 - Wavelet
- кодира се трансформираното изображение

■ **декомпресия**

- прилага се обратна трансформация върху кодираното изображение
- обединяват се блоковете в резултантно изображение

Компресия в JPEG

- **Компресия с последователно прилагане на DCT**
 - lossy компресия базирана на DCT/wavelet
 - lossless компресия базирана на ентропия



КРАЙ

Следваща тема:

Морфологични операции