

## TECHNIKI STOSOWANE W KOMPRESJI SYGNAŁÓW

## TECHNIKI KOMPRESJI SYGNAŁÓW

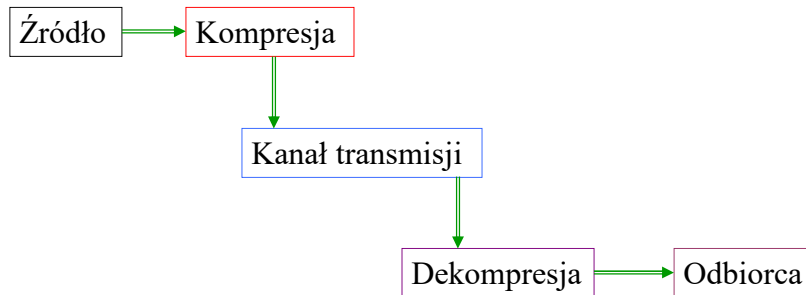
### 1. Metody bezstratne:

- nie dopuszcza się w nich utraty informacji (ani 1 bitu)
- najistotniejsze to tzw. metody entropii

### 2. Metody stratne:

- często stosowane w kodowaniu rzeczywistych sygnałów multimedialnych
- z reguły otrzymujemy dolno-częstotliwościową aproksymację sygnału oryginalnego (bez wysokoczęstotliwościowych detali/szczegółów)

## Standardowe techniki kompresji sygnałów



Motywacja:

1 sekunda wideo – 160 Mbit

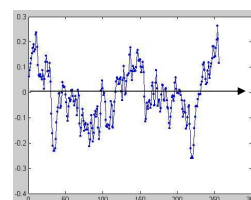
2 minuta muzyki (44 100 próbek/sek, 16 bitów/próbkę) 84 Mbit

## Kodowanie entropowe

EC od ang. Entropy Coding



$$s(n) \in \{s_m\}_{m=1}^M$$



Ilość informacji stowarzyszona z komunikatami  $\{s_m : m = 1, \dots, M\}$

wynosi  $I(m) = -\log_2 p_m$ , gdzie  $p_m$  jest prawdopodobieństwem

wystąpienia  $s_m$ , a jej miarą jest ilość bitów.

Średnia liczba bitów potrzebnych do bezstratnego zakodowania komunikatu (**entropia**) jest wartością oczekiwaną

$$H = -\sum_{m=1}^M p_m \log_2 p_m$$

## Alfabet Morse'a



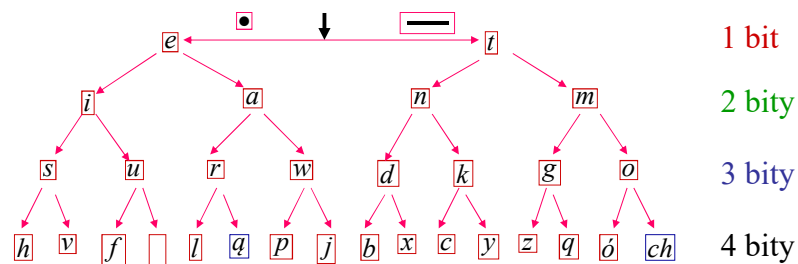
Samuel Finley Breese MORSE (1791-1872) –  
amerykański malarz i wynalazca

1837 – aparat telegraficzny

1840 - alfabet telegraficzny

1844 – pierwsza na świecie linia

Baltimore - Washington



## Sprawność kodowania



Oczekiwana ilość bitów

$$H_w = \sum_{m=1}^M b_m p_m$$

$$H = -\sum_{m=1}^M p_m \log_2 p_m$$

gdzie  $p_m$  jest prawdopodobieństwem komunikatu  
zakodowanego przez symbol posiadający  $b_m$  bitów.

Sprawność kodowania

$$\eta = \frac{H}{H_w} 100\%$$

może być co najwyżej równa 100%, bo entropia jest dolną  
granica średniej liczby bitów wymaganych do reprezentacji  
komunikatów.

## Kodowanie ze zmienną długością słowa

VLC ang. Variable Length Coding



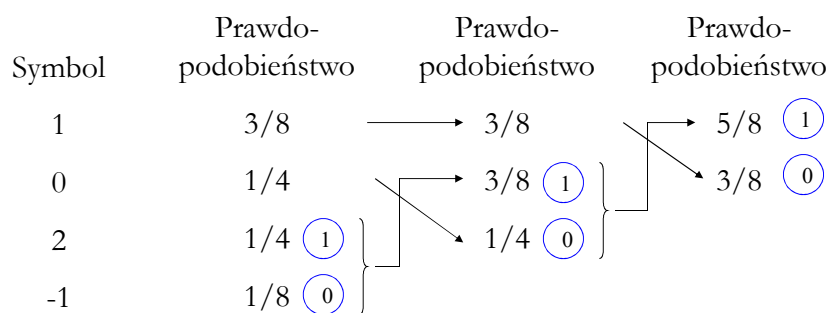
| $s_m$ | $p_m$ |
|-------|-------|
| -1    | 1/8   |
| 0     | 1/4   |
| 1     | 3/8   |
| 2     | 1/4   |

$$H = -\frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{3}{8} \log_2 \frac{3}{8} - \frac{1}{4} \log_2 \frac{1}{4} \approx 1,9$$

## Przykład 1: kodowanie metodą Huffmana



1952 rok



## Przykład 1: kodowanie metodą Huffmana



| Symbol             | -1    | 0     | 1     | 2     |
|--------------------|-------|-------|-------|-------|
| Prawdopodobieństwo | 0,125 | 0,250 | 0,375 | 0,250 |
| Kod binarny        | 110   | 10    | 0     | 111   |

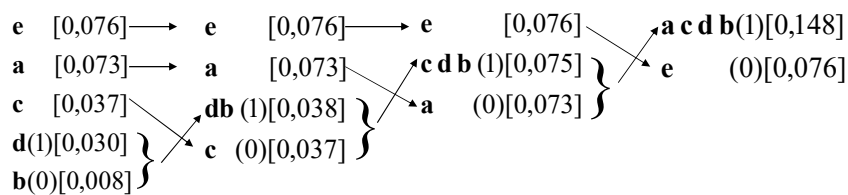
Przeciętna ilość bitów na symbol  $H_w = 1 \cdot \frac{3}{8} + 2 \cdot \frac{1}{4} + 3(\frac{1}{4} + \frac{1}{8}) = 2$

Sprawność kodowania  $\eta = \frac{1,9}{2} 100\% = 95\%$

111|111|110|0|0|10|0|0|110  
2 2 -1 1 1 0 1 1 -1

100 % sprawność kodowania gdy prawdopodobieństwa są potęgami 1/2

## Przykład 2: kodowanie metodą Huffmana



| Symbol             | a     | b     | c     | d     | e     |
|--------------------|-------|-------|-------|-------|-------|
| Prawdopodobieństwo | 0,073 | 0,008 | 0,037 | 0,030 | 0,076 |
| Kod binarny        | 10    | 1110  | 110   | 1111  | 0     |

### Przykład 3: kodowanie metodą Huffmana



$p_m$  gęstość prawdopodobieństwa zmiennej losowej  $s_m$

$$F_m = \sum_{i=1}^m p_i \quad \text{dystrybuanta}$$

| Znak | Prawdopo-<br>dobieństwo | Kod<br>Huffmana |
|------|-------------------------|-----------------|
| ♂    | 0,95                    | 1               |
| ♀    | 0,02                    | 00              |
| 😊    | 0,03                    | 01              |

$$H = 0,335 \text{ bit/symbol}$$

$$H_w = 1,05 \text{ bit/symbol}$$

Sprawność tylko  
 $\eta = 31,9\%$

### Przykład 4: kod Huffmana sekwencji symboli



| Sekwencja<br>symboli | Iloczyn<br>prawdopo-<br>dobieństw | Kod<br>Huffmana |
|----------------------|-----------------------------------|-----------------|
| ♂♂                   | 0,9025                            | 1               |
| ♂♀                   | 0,0190                            | 000             |
| ♂😊                   | 0,0285                            | 011             |
| ♀♂                   | 0,0190                            | 0010            |
| ♀♀                   | 0,0004                            | 001100          |
| ♀😊                   | 0,0006                            | 001110          |
| 😊♂                   | 0,0285                            | 010             |
| 😊♀                   | 0,0006                            | 001101          |
| 😊😊                   | 0,0009                            | 001111          |

$$H = 0,611 \text{ bit/symbol}$$

$$H_w = 1,222 \text{ bit/symbol}$$

sprawność  $\eta = 50\%$

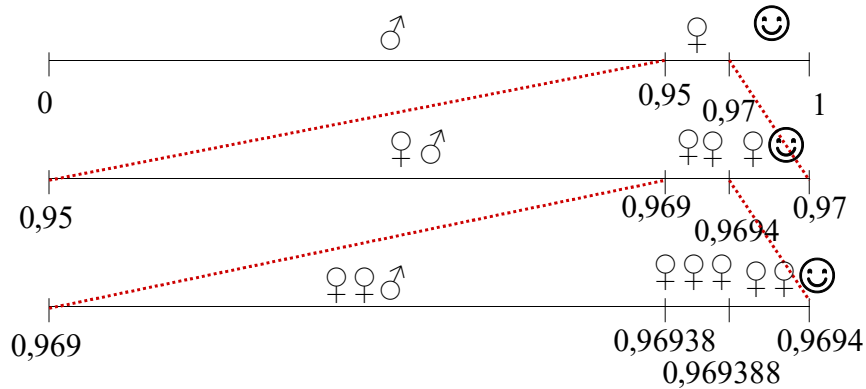
**Poprzednio:**

$$H = 0,335 \text{ bit/symbol}$$

$$H_w = 1,05$$

$$\eta = 31,9\%$$

## Kodowanie arytmetyczne



Liczba bitów potrzebna do zakodowania komunikatu jest częścią całkowitą  $1 - \log_2 p_n$

## Przykład 5: kodowanie arytmetyczne



| Sekwencja symboli | Iloczyn prawdopodobieństw | Granice przedziału | Wartość środkowa w kodzie dziesiętnym | Wartość środkowa w kodzie binarnym        | $1 - \log_2 p_n$ |
|-------------------|---------------------------|--------------------|---------------------------------------|---|------------------|
| ♂♂                | 0,9025                    | 0 – 0,9025         | 0,45125                               | 0,0111001110000101<br><b>0</b>            | 1,15             |
| ♂♀                | 0,0190                    | 0,9025 – 0,9215    | 0,912                                 | 0,1110100101111000<br><b>111010</b>       | 6,72             |
| ♂😊                | 0,0285                    | 0,9215 – 0,95      | 0,93575                               | 0,1110111110001101<br><b>111011</b>       | 6,13             |
| ♀♂                | 0,0190                    | 0,95 – 0,969       | 0,9595                                | 0,1111010110100001<br><b>111101</b>       | 6,72             |
| ♀♀                | 0,0004                    | 0,969 – 0,9694     | 0,9692                                | 0,1111100000111101<br><b>111110000011</b> | 13,29            |
| ♀😊                | 0,0006                    | 0,9694 – 0,97      | 0,9697                                | 0,1111100000111110<br><b>11111000001</b>  | 11,70            |
| 😊♂                | 0,0285                    | 0,97 – 0,9985      | 0,98425                               | 0,1111101111110111<br><b>111110</b>       | 6,13             |
| 😊♀                | 0,0006                    | 0,9985 – 0,9991    | 0,9988                                | 0,1111111111010001<br><b>11111111101</b>  | 11,70            |
| 😊😊                | 0,0009                    | 0,9991 – 1         | 0,99955                               | 0,1111111111100010<br><b>1111111111</b>   | 11,12            |

$H_w = 1,5008$   
 $\eta = 40,7\%$

## Kodowanie ciągów

RLC ang. Run Length Coding

$(v, r)$  gdzie  $v$  (od ang. value) powtarzający się symbol  
 $r$  (od ang. run) liczba powtórzeń

1111000111111111000111100000011111111111

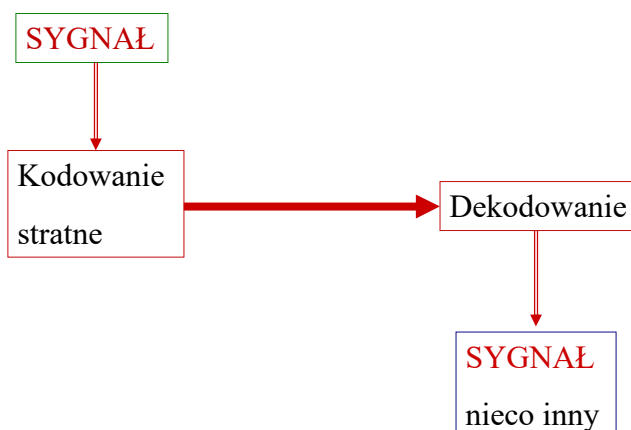
(1,1) (0,1) (1,2) (0,1) (1,1) (0,2) (1,3)

1 kod 1111

0 kod 000

15

## Metody kompresji stratnej



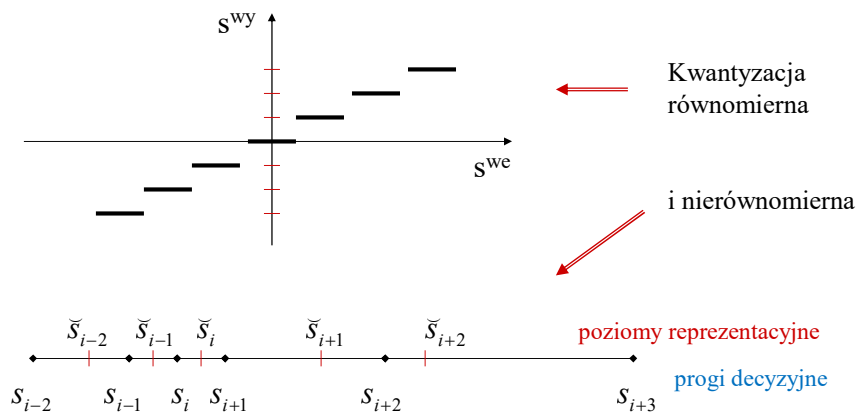
Z reguły sygnał pozbawiony jest informacji wysoko-częstotliwościowej

16



## Kwantyzacja skalarna

SQ ang. Scalar Quantization



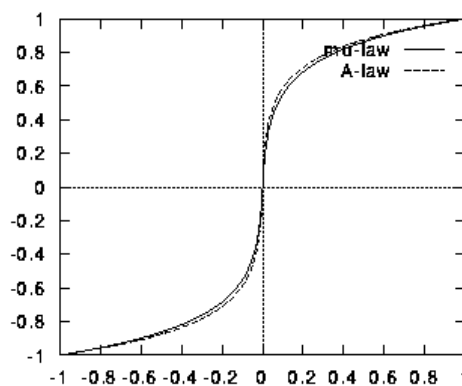
## Kwantyzacja skalarna



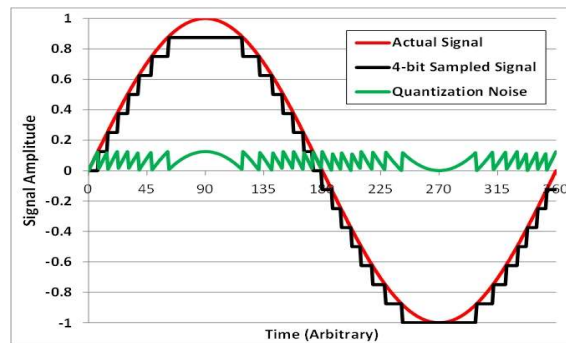
*Kwantyzacja* jest procesem **nieodwracalnym**, a *kwantyzacja odwrotna* polega na przeskalowaniu do oryginalnego zakresu.

*Kwantyzacja nierównomierna*

- A-law
- $\mu$ -law

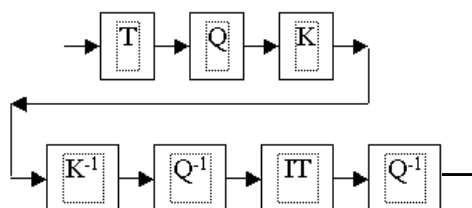


## Błąd kwantyzacji



- *Szum kwantyzacji* – różnica pomiędzy sygnałem oryginalnym a skwantowanym, jest przypadkowy i szerokopasmowy, lecz nie jest szumem białym gdyż jest skorelowany z sygnałem oryginalnym.
- Im mniej bitów (mniej poziomów kwantyzacji) tym silniejszy szum kwantyzacji

## Kodowanie transformatowe



**T** - transformacja

**Q** - kwantowanie

**K** – kodowanie bezstratne

**K<sup>-1</sup>** - dekodowanie

**Q<sup>-1</sup>** - dekwantyzacja

**IT**- transformacja odwrotna

## Poznane transformacje częstotliwościowe



Krótko-czasowa Transformacja Fouriera (STFT)

$$\hat{s}(f) = \int_{-\infty}^{+\infty} w(t) s(t) e^{-2\pi j f t} dt = \int_{-\infty}^{+\infty} w(t) s(t) (\cos(2\pi f t) - j \sin(2\pi f t)) dt$$

Transformacja Falkowa

$$\tilde{s}_{\psi}(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t) \psi\left(\frac{t-b}{a}\right) dt$$

Transformacja Kosinusowa

$$\hat{s}(f) = \int_{-\infty}^{+\infty} w(t) s(t) \cos(2\pi f t) dt$$

## Dyskretna transformacja kosinusowa



DCT od ang. Discrete Cosine Transform

Transformacja DCT:

$$\hat{s}(k) = \sqrt{\frac{2}{N}} c(k) \sum_{n=0}^{N-1} s(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad k = 0, 1, \dots, N-1$$

$$\text{gdzie } c(k) = \begin{cases} 1/\sqrt{2} & \text{dla } k = 0 \\ 1 & \text{dla } k \neq 0 \end{cases}$$

Odwrotna transformacja DCT (IDCT):

$$s(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c(k) \hat{s}(k) \cos\left(\frac{(2n+1)k\pi}{2N}\right)$$

## Dyskretna transformacja kosinusowa

DCT od ang. Discrete Cosine Transform



Analogicznie do DFT, transformację DCT możemy obliczyć macierzowo

$$\hat{\mathbf{s}} = \mathbf{W}_{DCT} \mathbf{s}$$

**Przykład:** korzystając z równania z poprzedniego slajdu, wyznacz 4-elementową macierz transformacji DCT. Dla  $N=4$

$$n = 0, 1, \dots, N-1 \quad k = 0, 1, \dots, N-1$$

$$\mathbf{W}_{DCT} = \begin{bmatrix} \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) \\ \frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{5\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{7\pi}{8}) \\ \frac{1}{\sqrt{2}}\cos(\frac{2\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{6\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{10\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{14\pi}{8}) \\ \frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{9\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{15\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{21\pi}{8}) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) \\ \frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) & -\frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) & -\frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) \\ \frac{1}{\sqrt{2}}\cos(\frac{\pi}{4}) & -\frac{1}{\sqrt{2}}\cos(\frac{\pi}{4}) & -\frac{1}{\sqrt{2}}\cos(\frac{\pi}{4}) & \frac{1}{\sqrt{2}}\cos(\frac{\pi}{4}) \\ \frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) & -\frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) & \frac{1}{\sqrt{2}}\cos(\frac{\pi}{8}) & -\frac{1}{\sqrt{2}}\cos(\frac{3\pi}{8}) \end{bmatrix}$$

23

## Dyskretna transformacja kosinusowa

DCT od ang. Discrete Cosine Transform



**Zalety:**

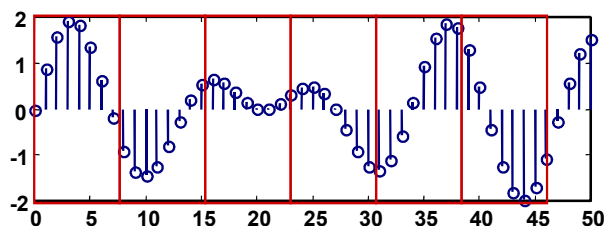
- współczynniki po transformacji DCT są bliskie wartości 0, więc wiele z nich wyzeruje się po kwantyzacji
  - => mniej bitów bez dużego błędu kwantyzacji
- Transformata N-wymiarowa jako N transformat 1-wymiarowych
- Po transformacji DCT sygnał jest zdekorelowany
  - => często stosowany w kompresji

24

## Przykład: DCT dla bloków po 8 próbek

Dzieląc sygnał na bloki po 8 próbek posługujemy się transformacją

$$s(k) = 0,5 c(k) \sum_{n=0}^7 s(n) \cos((2n+1)k\pi/16)$$



W celu uniknięcia efektów brzegowych w DCT, np. w kompresji JPEG oraz MPEG stosuje się **Zmodyfikowaną DCT** (MDCT), która bazuje na nachodzących na siebie blokach.

25

## Sygnał audio CD (Compact Disc-Audio):

- Sony vs Philips (13 vs 16 bitów)
- Dynamika na poziomie 96 dB bo kwantyzacja równomierna (PCM) gdzie każdy bit to podwojenie liczby poziomów kwantyzacji czyli wzrost parametru SNR o 6 dB
 
$$6 \text{ dB} / \text{bit} * 16 \text{ bitów} = 96 \text{ dB}$$
- $F_p = 44.1\text{kHz}$  stąd mamy 44 100 próbek/sekundę po 16 bitów co daje 705 600 bitów/s w jednym kanale

26

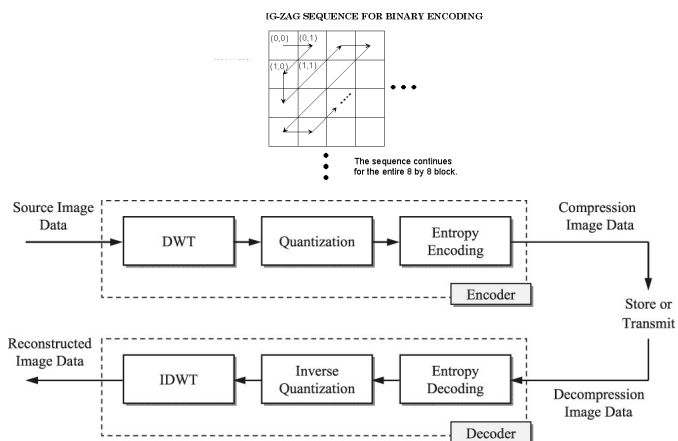
## Kompresja obrazów



### ■ Schemat blokowy - kompresja obrazu JPEG 2000



R,G,B → Y (składowa luminancji),Cb,Cr

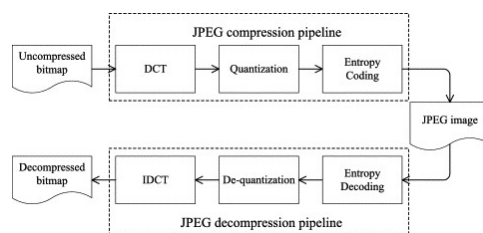


27

## Kompresja obrazów



### ■ Schemat blokowy - kompresja obrazu JPEG



### ■ Przykład

Obraz oryginalny



Obraz skompresowany



28

## Kompresja audio

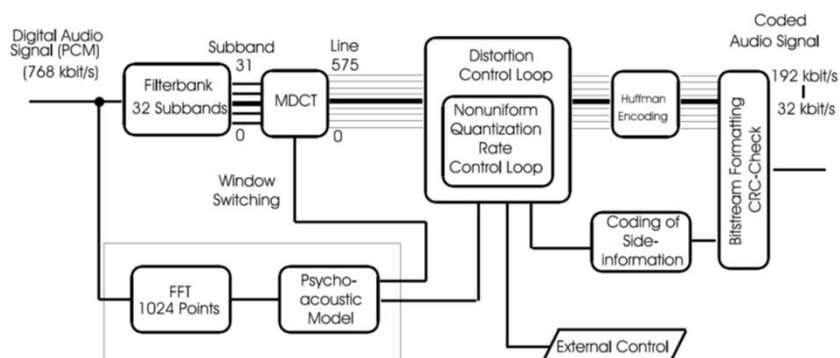


- Kompresja audio jako wsparcie aplikacji:  
Digital Audio Broadcasting, Internet audio, DVD
- Percepcyjne kodowanie dźwięku:
  - Optymalizacja subiektywnej jakości, a nie obiektywnych parametrów
  - Wykorzystanie psychoakustyki
  - Utrzymanie zakłóceń kodowania poniżej limitów psychoakustycznych  
(nietransmitowanie danych nieistotnych)
  - Brak jednego modelu źródła (jak w przypadku mowy)

## Kompresja audio – mp3



Schemat blokowy MPEG-1 warstwa 3 (mp3)

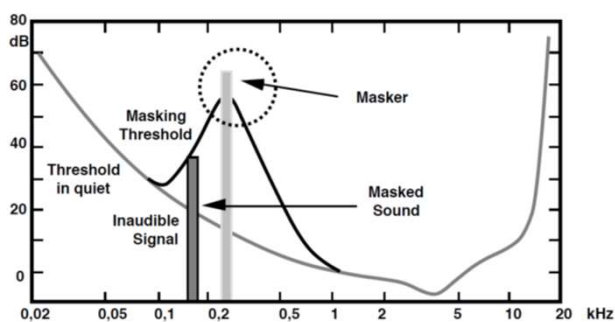


## Kompresja audio - psychoakustyka



### ■ Efekt maskowania sygnałów w sąsiednich pasmach:

- Asymetria maskowania
- Sygn. zbliżone do szumu lepiej maskują niż sygn. tonowe
- Zależne od poziomu sygnału i szumu



## Przykład kompresji audio



Spektrogram + audio :



■ Oryginał



■ mp3



■ Różnica między  
oryginałem a mp3