

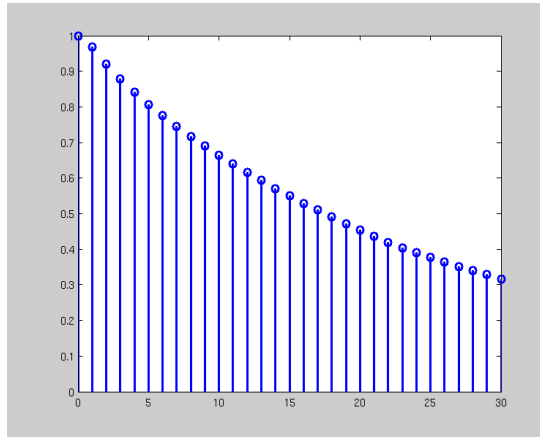
## Zadanie 1

Dla próbek pochodzących z pliku 'lena256.pgm', wczytanych w następujący sposób

```
x = double(imread('lena256.pgm'));    %wczytaj obraz
x = x(:);                             %przekształć na sygnał jednowymiarowy
x = x - mean(x);                       %odejmij składową stałą
```

wyznaczyć:

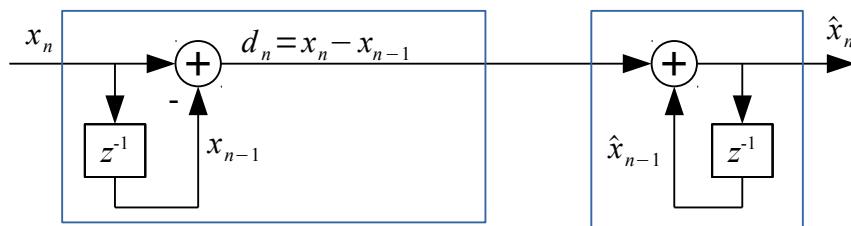
1) unormowaną funkcję autokorelacji sygnału (xcorr z opcją 'coeff');



2) wariancję sygnału: (2.7345e+03);

3) wariancję różnic próbek sygnału (diff(x) = 175.5518);

4) zysk z kodowania DPCM, gdzie jako predykcję próbki bieżącej używa się wartości próbki poprzedniej ( $G=15.5766$ ), (patrz schemat poniżej)



$$G_{DPCM} = \frac{\sum_{n=1}^N x_n^2}{\sum_{n=1}^N d_n^2} = \frac{\sigma_x^2}{\sigma_d^2}.$$

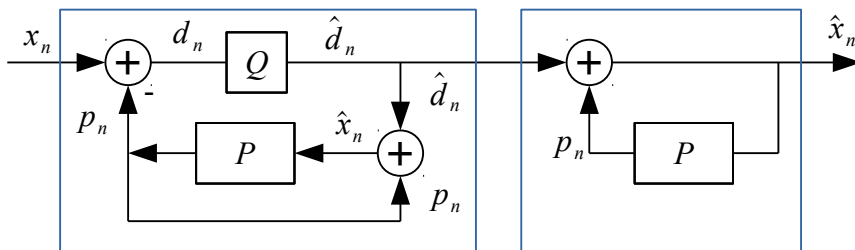
5) zysk z kodowania DPCM, dla predyktora z jednym współczynnikiem ( $a=0.9679$ ,  $G=15.8287$ ), dwoma ( $a=[1.2314, -0.2723]$ ,  $G=17.0960$ ) i trzema ( $a=[1.2729, -0.4598, 0.1523]$ ,  $G=17.5021$ );

Współczynniki predykcji  $a$  wyznacza się korzystając z równań Yulea-Walker'a:

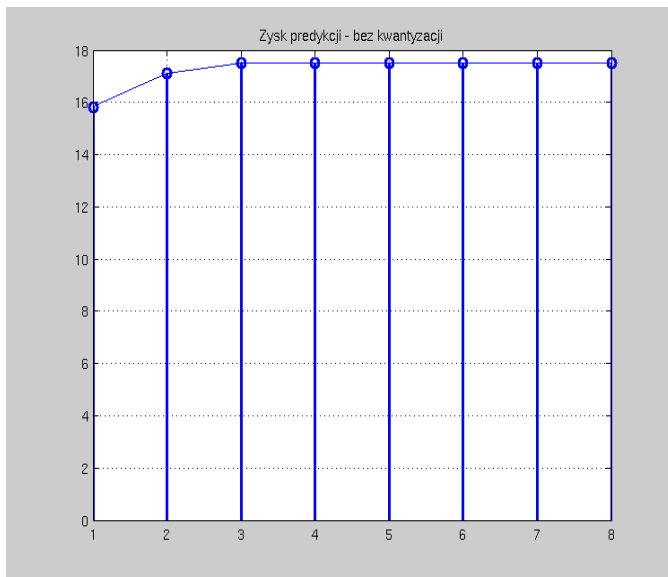
$$\begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(N) \end{bmatrix} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & \cdots & R_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix},$$

gdzie wartości  $R_{xx}$  są wartościami funkcji autokorelacji sygnału.

Schemat algorytmu DPCM z predyktorem  $P$  i opcjonalnym kwantyzatorem pokazano poniżej.



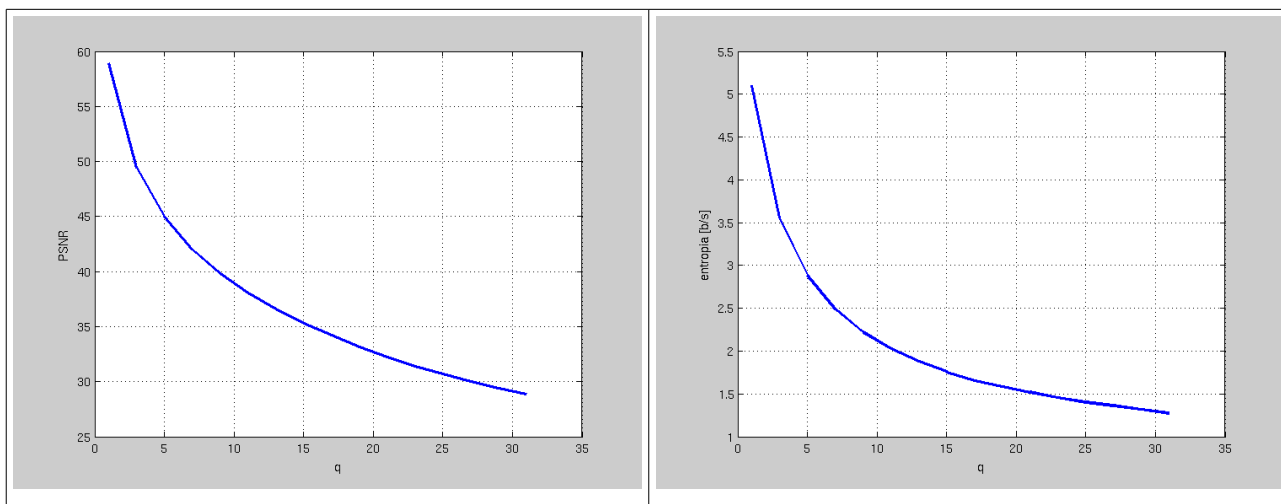
funkcja predykcji  $P$ :  $p_n = \sum_{i=1}^N a_i \hat{x}_{n-i}$



6) błąd rekonstrukcji  $PSNR$  w funkcji stopnia kwantyzacji ( $q=1:32$ )

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \quad MSE = \frac{1}{N} \sum_{n=1}^N (x_n - \hat{x}_n)^2$$

7) entropię sygnału błędu predykcji w funkcji stopnia kwantyzacji ( $q=1:32$ ). Operacja kwantyzacji wykonywana jest w następujący sposób  $xq = q * \text{round}(x/q)$



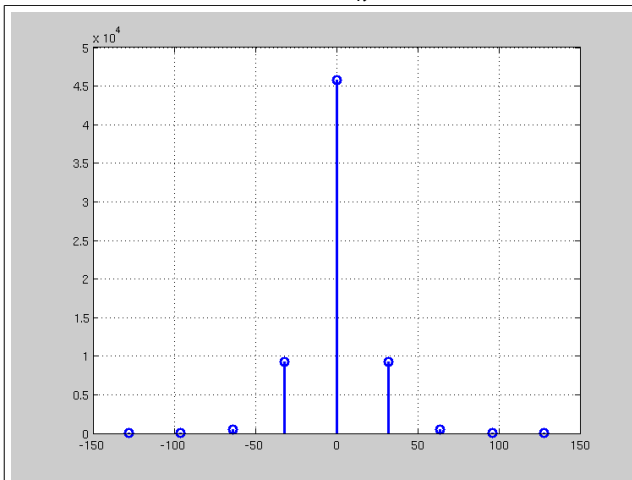
Entropia Shannona dyskretnego źródła o alfabcie  $X = \{x_1, x_2, \dots, x_N\}$  i prawdopodobieństwach wystąpienia symboli  $P = \{p_1, p_2, \dots, p_N\}$  określona jest wzorem

$$H(x) = - \sum p(x_i) \log_2 p(x_i),$$

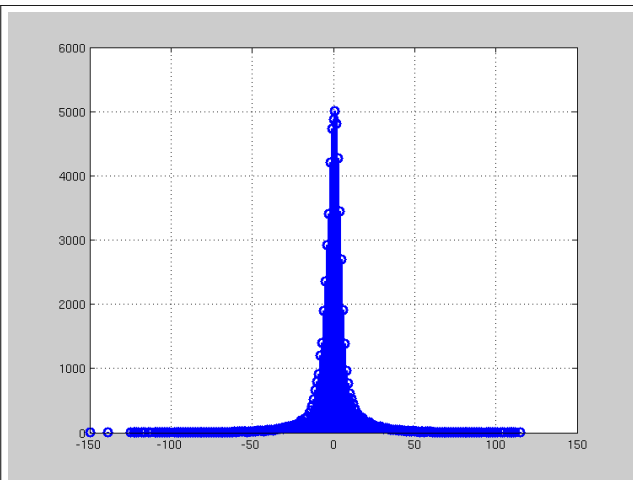
gdzie  $\log_2 p(x_i)$  jest liczbą bitów potrzebnych do reprezentacji symbolu  $x_i$

By zastosować powyższy wzór na entropię należy wyznaczyć częstość wystąpień poszczególnych

wartości próbek sygnału  $\hat{d}_n$ .



Histogram wartości  $\hat{d}_n$  dla kwantyzatora  $q=32$



Histogram wartości  $\hat{d}_n$  dla kwantyzatora  $q=2$

Przykładowo dla kwantyzatora  $q=32$  częstości te wynoszą

| wartość | ilość | częstość (estymata $p(x_i)$ ) |
|---------|-------|-------------------------------|
| -186    | 1     | 0.000015258789062             |
| -155    | 2     | 0.000030517578125             |
| -124    | 23    | 0.000350952148438             |
| -93     | 120   | 0.001831054687500             |
| -62     | 494   | 0.007537841796875             |
| -31     | 8734  | 0.133270263671875             |
| 0       | 46743 | 0.713241577148438             |
| 31      | 8776  | 0.133911132812500             |
| 62      | 530   | 0.008087158203125             |
| 93      | 98    | 0.001495361328125             |
| 124     | 15    | 0.000228881835938             |

## Zadanie 2

Stosując niżej przedstawiony algorytm zaprojektować kod Huffmana dla symboli z algorytmu DPCM z zadania nr 1.

### Algorytm konstrukcji kodu Huffmana

1. utwórz tablicę o  $2N-1$  kolumnach, gdzie  $N$  jest liczbą symboli (patrz przykład poniżej),
2. znajdź dwa symbole  $s_i$  i  $s_j$  o najmniejszej wadze i zerowej wartości pola *Link*
3. w drugiej części tablicy w pierwszej wolnej komórce utwórz nowy symbol  $s_{ij}$  i zapisz sumę wag symboli  $s_i$  i  $s_j$ .
4. w polu *Link* zapisz numer nowo utworzonego symbolu
5. w polu *P* symbolu  $s_i$  wpisz 0, a w pole *P* symbolu  $s_j$  wpisz 1.
6. jeśli są co najmniej dwa symbole z pustym polem *Link* wróć do punktu 2
7. rozpoczynając od symbolu (pozycje:  $1:N$ ) idziemy do korzenia (pozycja  $2N-1$ ) korzystając z pola *Link* i zapisując wartości z pola *P* – w ten sposób tworzymy ścieżkę od symbolu do korzenia drzewa Huffmana.
8. utworzone ciągi zapisujemy w odwróconej kolejności otrzymując zapis ścieżki od korzenia drzewa do symbolu.

**Przykład:** Stosując powyższy algorytm wyznaczyć kod dla symboli o następujących częstościach wystąpień:  $p(s_1)=0.1$ ,  $p(s_2)=0.1$ ,  $p(s_3)=0.15$ ,  $p(s_4)=0.2$ ,  $p(s_5)=0.45$ ,

Etap 1.

|      |     |     |      |      |      |   |   |   |   |
|------|-----|-----|------|------|------|---|---|---|---|
| Id   | 1   | 2   | 3    | 4    | 5    | 6 | 7 | 8 | 9 |
| P    | 0.1 | 0.1 | 0.15 | 0.20 | 0.45 |   |   |   |   |
| Link |     |     |      |      |      |   |   |   |   |

Etap 2: symbole 1 i 2, utworzono 6

|      |   |   |      |      |      |     |   |   |   |
|------|---|---|------|------|------|-----|---|---|---|
| Id   | 1 | 2 | 3    | 4    | 5    | 6   | 7 | 8 | 9 |
| P    | 0 | 1 | 0.15 | 0.20 | 0.45 | 0.2 |   |   |   |
| Link | 6 | 6 |      |      |      |     |   |   |   |

Etap 2: symbole 3 i 6, utworzono 7

|      |   |   |   |      |      |   |      |   |   |
|------|---|---|---|------|------|---|------|---|---|
| Id   | 1 | 2 | 3 | 4    | 5    | 6 | 7    | 8 | 9 |
| P    | 0 | 1 | 0 | 0.20 | 0.45 | 1 | 0.35 |   |   |
| Link | 6 | 6 | 7 |      |      | 7 |      |   |   |

Etap 2: symbole 4 i 7, utworzono 8

|      |   |   |   |   |      |   |   |      |   |
|------|---|---|---|---|------|---|---|------|---|
| Id   | 1 | 2 | 3 | 4 | 5    | 6 | 7 | 8    | 9 |
| P    | 0 | 1 | 0 | 0 | 0.45 | 1 | 1 | 0.55 |   |
| Link | 6 | 6 | 7 | 8 |      | 7 | 8 |      |   |

Etap 2: symbole 5 i 8, utworzono 9 – koniec konstrukcji drzewa

|      |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|
| Id   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| P    | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |   |
| Link | 6 | 6 | 7 | 8 | 9 | 7 | 8 | 9 |   |

Słowa kodu otrzymane w wyniku wykonania powyższego algorytmu

| Symbol | Utworzony ciąg | Kod  |
|--------|----------------|------|
| A      | 0111           | 1110 |
| B      | 1111           | 1111 |
| C      | 011            | 110  |
| D      | 01             | 10   |
| E      | 0              | 0    |

### Zadanie 3

1) Wyznaczyć funkcję autokorelacji sygnału 'lena256.pgm'; i zbudować macierz  $R_{xx}$

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \cdots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & \cdots & R_{xx}(N-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & \cdots & R_{xx}(0) \end{bmatrix}$$

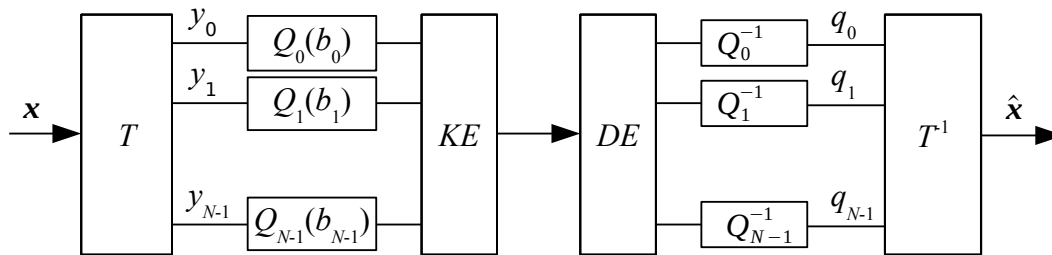
2) Znaleźć wektory własne i zbudować z nich macierz  $T$  korzystając z funkcji  $\text{eig}$ :  $[V, D] = \text{eig}(R_{xx})$ ;  $R_{xx} * V = V * D$ ;  $T = V'$ ;

3) Wyznaczyć współczynniki transformacji sygnału względem  $T$ , DCT, FFT. Przy FFT pamiętać o tym, że  $|y|^2 = y * \text{conj}(y)$

4) Wyznaczyć  $G_{TC}$  dla powyższych transformacji.

$$G_{TC} = \frac{\frac{1}{N} \sum_{k=0}^{N-1} \sigma_{yk}^2}{\left( \prod_{k=0}^{N-1} \sigma_{yk}^2 \right)^{1/N}}$$

gdzie  $\sigma_{y_k}^2$  są wariancjami sygnałów  $y_k$  (patrz rysunek poniżej)



#### Zadanie 4 (kodowanie transformacyjne sygnałów 2D)

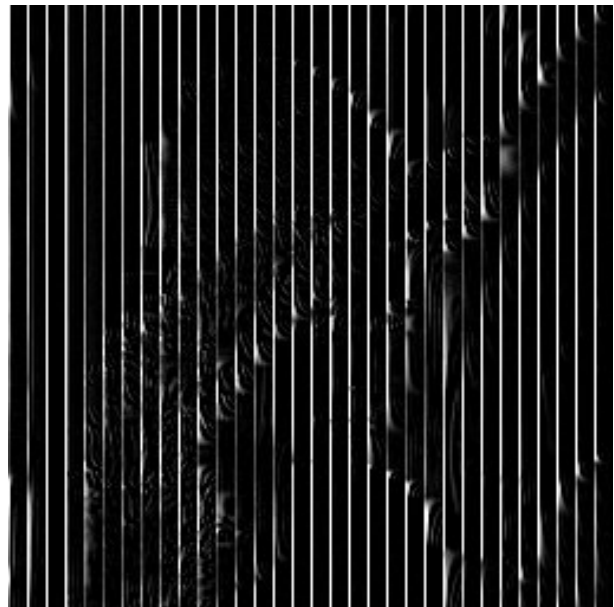
- 1) Podzielić obraz Lena na rozłączne bloki o wymiarach 8x8 i wyznaczyć ich transformatę kosinusową, skorzystać z funkcji DCT pakietu matlab.
- 2) Przeprowadzić rekwantyzację skalarną współczynników uzyskanych w punkcie 1 przy użyciu funkcji round w następujący sposób

$$x = sf * \text{round}(x/sf); \text{ gdzie } sf \text{ jest współczynnikiem kwantyzacji.}$$

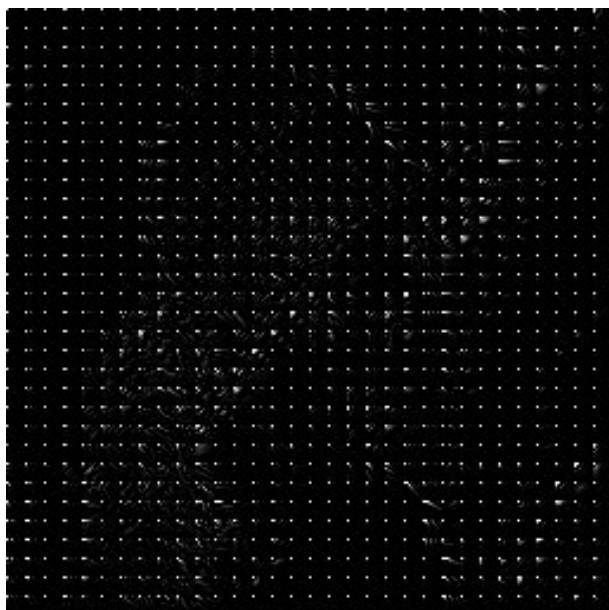
- 3) Wyznaczyć procent współczynników niezerowych
- 4) Dokonać rekonstrukcji sygnału poprzez wyznaczenie transformacji odwrotnej. Skorzystać z funkcji IDCT.
- 5) Wyznaczyć wartość wskaźnika jakości PSNR.



Obraz oryginalny



Wynik transformacji DCT wzdłuż wierszy



Wynik transformacji DCT po wierszach i po kolumnach (2D)