

**Labolatorium PTI – ćwiczenie 1, przykład 5**

# **Ucyfrowienie sygnałów**

## **Kwantyzacja sygnałów**

# 1 POMIAR SYGNAŁÓW: kwantyzacja i tw. Lloyd-Maxa

## 1.1 Teoria

**Kwantyzacja** to nieodwracalne nieliniowe odwzorowanie statyczne zmniejszające dokładność danych przez ograniczenie ich zbioru wartości. Zbiór wartości wejściowych dzielony jest na rozłączne przedziały. Każda wartość wejściowa wypadająca w określonym przedziale jest w wyniku kwantyzacji odwzorowana na jedną wartość wyjściową przypisaną temu przedziałowi, czyli tak zwany **poziom reprezentacji**. W rozumieniu potocznym proces kwantyzacji można przyrównać do "zaokrąglania" wartości do określonej skali.

Wartości wejściowe muszą zostać jednoznacznie skojarzone z poziomami reprezentacji, dlatego przedział dopuszczalnych wartości wejściowych jest dzielony na podprzedziały; punkty podziału są nazywane **poziomami decyzyjnymi**, ich liczba jest o jeden mniejsza od liczby poziomów reprezentacji. Każda wartość należąca do danego podprzedziału jest zastępowana przez poziom reprezentacji przypisywany do danego przedziału.

**Twierdzenie 1.1** (Lloyda-Maxa). *Przedziały  $\{b_i\}_{i \leq n}$  i reprezentanci  $\{y_i\}_{i \leq n-1}$  optymalnej kwantyzacji spełniają następujące warunki:*

$$y_i = \frac{\int_{b_{i-1}}^{b_i} x f(x) dx}{\int_{b_{i-1}}^{b_i} f(x) dx}$$
$$b_i = \frac{y_i + y_{i+1}}{2}$$

Twierdzenie daje podstawy do iteracyjnej metody poprawiania parametrów kwantyzacji.

## 1.2 Przykład matlabowy

W poniższym przykładzie kwantyzujemy sinusa na przedziale  $[0, 2\pi]$ . Sinus został wygenerowany co 0.1. Początkowo jest kwantowany równomiernie co 0.2. Następnie przy użyciu algorytmu Lloyd dobieramy lepsze poziomy decyzyjne co pozwala zmniejszyć błąd średniokwadratowy. Początkowo wynosił Początkowo wynosił on 0.0148, po optymalizacji 0.0022. Na rysunku również widać poprawę względem początkowo dobranych wartości.

```
1 t = [0:.1:2*pi];%time
  sig = sin(t);
3 partition = [-1:.2:1];
  codebook = [-1.2:.2:1];
5 % Now optimize, using codebook as an initial guess.
  [partition2, codebook2] = lloyds(sig, codebook);
7 [index, quants, distor] = quantiz(sig, partition, codebook);
  [index2, quant2, distor2] = quantiz(sig, partition2, codebook2);
9 % Compare mean square distortions from initial and optimized
  [distor, distor2] % parameters.
11
```

```

13 plot(t, sig, '- ', t, quants, '.', t, quant2, 'x')
    legend('Original signal', 'Quantized signal', 'Optimized quantized
        signal');
    axis([-0.2 7 -1.2 1.2])

```

sinus.m

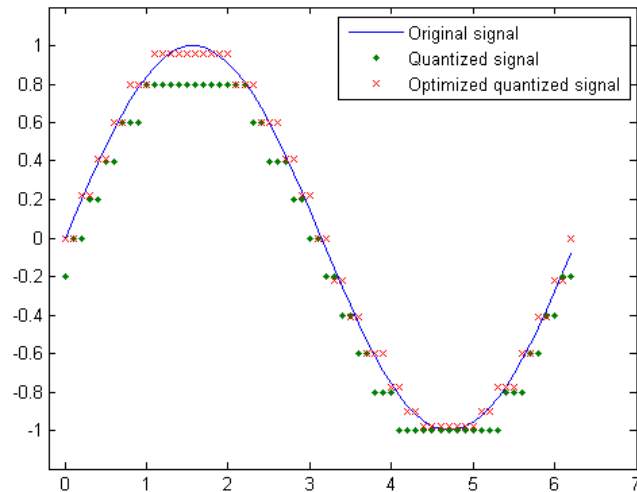


Figure 1: Kwantyzacja sinusa

### 1.3 Przykład praktyczny

W tym przykładzie przeprowadzimy kwantyzację obrazka. Z 256 zmniejszono do 8 poziomów szarości.

```

RGB = imread('..\mietus.png');
2 I = rgb2gray(RGB);
  imshow(I);
4 I = double(I);
  step=32;
6 partition = [step:step:256];
  r = [0:step:256];
8 [m,n] = size(I); % get size of your image
  I = reshape(I, m*n, 1);
10 min=10000;
  res=1:9;
12 for i=1:11
    [partition, codebook] = lloyds(I, r);
14    [index, quant, distor] = quantiz(I, partition, codebook);
    res(i)=distor;
16    if distor < min
        min=distor;
18        bestP=partition;
        bestC=codebook;
20        bestR=r;
    end
22    r = [0:step:256];

```

```

    r=round(sort(randn(1,9)*step+r));
24 end
    [index, quant, distor] = quantiz(I, bestP, bestC);
26 res
    bestR
28 bestC
    quant = uint8(reshape(quant, m, n));
30 imshow(quant);

```

obrazek.m

Jak widać optymalizacja pozwoliła na zachowanie dość dobrej jakości obrazka, przy stosunkowo małej liczbie poziomów szarości.



Figure 2: Oryginał



Figure 3: Kwantyzacja równomierna



Figure 4: Kwantyzacja zoptymalizowana

#### 1.4 Modyfikacja - losowanie startowych poziomów decyzyjnych

Błąd średniokwadratowy po zastosowaniu algorytmu Lloyd'a wynosił początkowo 37.5949. Sprawdzone zostały dwie modyfikacje: z losowaniem poziomów decyzyjnych z przedziału  $[0,256]$  oraz modyfikacja gaussowska (z wartością oczekiwaną 0 i odchyleniem standardowym 32) poziomów równomiernych. Ekperyment powtórzono po 10 razy uzyskując kolejno następujące wyniki:

losowane	37.4387	37.6599	44.1507	44.1507	44.1507
	44.1507	44.1507	40.3692	41.3335	38.7121
Gauss	37.6599	37.3715	38.1602	37.6726	37.8819
	37.1689	37.6599	37.1689	42.8068	44.1507

Z obserwacji wyników widać, że start algorytmu z równomiernego rozkładu daje dobre wyniki. Delikatne zaburzenie może spowodować poprawę działania algorytmu jednak wylosowanie wszystkich poziomów najczęściej pogarsza wynik.