WYKŁAD 9

Liczenie różnic pomiędzy obrazami – 2 sposoby

1. Różnica pomiędzy dwoma **wektorami** reprezentującymi porównywane obrazy (***metryka***)
2. Różnica pomiędzy dwoma **tablicami** reprezentującymi porównywane obrazy (operacja jednopunktowa dwuargumentowa)

1. Obraz jako wektor (punkt w przestrzeni n-wymiarowej):

$$\underline{d} = [d_1, ..., d_n]^T \qquad \underline{d} \in D$$

n – liczba piksli obrazu , np. N^2
D – przestrzeń obrazów
Wartości odpowiednich składowych wektora zależą od sposobu przeglądu obrazu (np. 1,2) dwa sposoby przeglądu linia po linii, 3) krzywa Hilberta.

Metryka - odwzorowanie: $\rho : X \times X \to R_*$ spełniające dla wszystkich wektorów $\underline{x}^\mu \in X \left(\mu = 1,2,\ldots\right)$ założenia (warunki):

$\rho\left(\underline{x}^\mu, \underline{x}^\nu\right) = 0 \Leftrightarrow \underline{x}^\mu \equiv \underline{x}^\nu$ - tożsamość,

$\rho\left(\underline{x}^\mu, \underline{x}^\nu\right) = \rho\left(\underline{x}^\nu, \underline{x}^\mu\right)$ - symetria,

$\rho\left(\underline{x}^\mu, \underline{x}^\nu\right) \le \rho\left(\underline{x}^\mu, \underline{x}^\eta\right) + \rho\left(\underline{x}^\eta, \underline{x}^\nu\right)$ - warunek trójkąta.

gdzie:

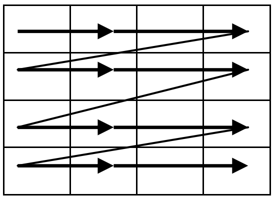
X - przestrzeń wektorów,

R_* - zbiór liczb nieujemnych

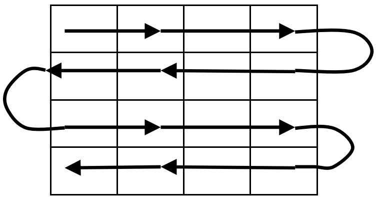
<u>Zastosowanie praktyczne:</u> obliczanie **różnic** pomiędzy poszczególnymi obrazami

Sposoby przeglądu obrazu

1. Linia po linii (wiersz po wierszu) – 1-szy sposób
   Przykład: N=4



2. Linia po linii (wiersz po wierszu) – 2-gi sposób
   Przykład: N=4

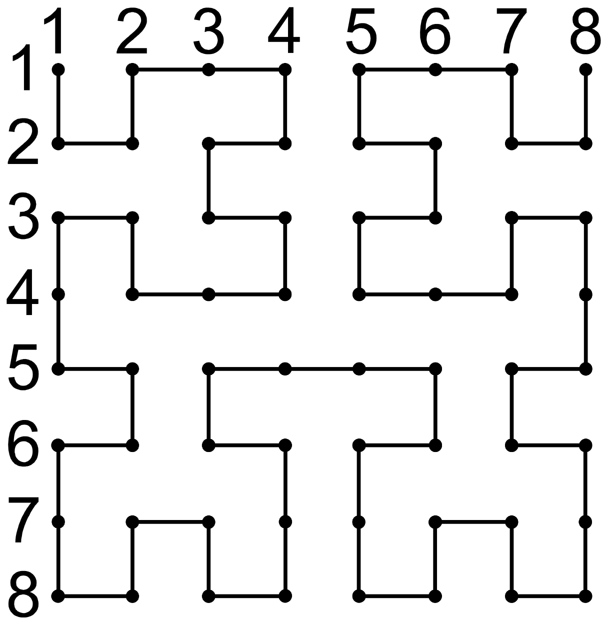


3. Według krzywej Hilberta rzędu k

Rekurencyjny algorytm generowania krzywych Hilberta. Z czterech krzywych rzędu k-1 budowana jest krzywa rzędu k.

Przykład: k=3

Krzywa Hilberta rzędu 3.



Metryka Euklidesowa:

$$\rho_1(\underline{x}^{\mu},\underline{x}^{\eta}) = \sqrt{\sum_{\nu=1}^{n}(x_{\nu}^{\mu} - x_{\nu}^{\eta})^2}$$

<u>Wady i zalety metryki Euklidesowej</u>
- odpowiada obiegowej definicji odległości
- ignorowanie składowych o b. małych wymiarach, długie czasy obliczeń (pierwiastkowanie, podnoszenie do kwadratu)

Uogólniona metryka Euklidesowa:

$$\rho_2(\underline{x}^{\mu},\underline{x}^{\eta}) = \sqrt{\sum_{\nu=1}^{n}\left[\lambda_{\nu}\left(x_{\nu}^{\mu} - x_{\nu}^{\eta}\right)\right]^2}$$

λ_{ν} - mnożniki normalizujące

Metryka uliczna (Manhattan, city block distance):

$$\rho_3\left(\underline{x}^{\mu},\underline{x}^{\eta}\right) = \sum_{\nu=1}^{n}\left|x_{\nu}^{\mu} - x_{\nu}^{\eta}\right|$$

Uogólniona metryka uliczna:

$$\rho_4\left(\underline{x}^{\mu},\underline{x}^{\eta}\right) = \sum_{\nu=1}^{n}\lambda_{\nu}\left|x_{\nu}^{\mu} - x_{\nu}^{\eta}\right|$$

Metryka Czebyszewa (maksymalna):

$$\rho_5\left(\underline{x}^{\mu},\underline{x}^{\eta}\right) = \max_{1\le\nu\le n}\left|x_{\nu}^{\mu} - x_{\nu}^{\eta}\right|$$

<u>Oznaczenia:</u>

$$\underline{x}^{\mu} = [x_1^{\mu},\dots x_{\nu}^{\mu},\dots x_n^{\mu}]^T\ ,\quad \underline{x}^{\eta} = [x_1^{\eta},\dots x_{\nu}^{\eta},\dots x_n^{\eta}]^T$$

Przykład

Obliczyć różnicę (stosując metrykę euklidesową) pomiędzy dwoma obrazami trzypikslowymi

$$\underline{d}^1 = [d_1^1, d_2^1, d_3^1]^T = [2,3,5], \quad \underline{d}^2 = [d_1^2, d_2^2, d_3^3]^T = [1,4,5]$$

$$\rho(\underline{d}^1, \underline{d}^2) = \sqrt{\sum_{i=1}^{3} (d_i^1 - d_i^2)^2} = \sqrt{(2-1)^2 + (3-4)^2 + (5-5)^2} = \sqrt{2}$$

2. Obraz jako tablica

Obliczanie wartości bezwzględnych różnic pomiędzy odpowiednimi pikslami dwóch obrazów

$$[\, c_{ij}] = [\, |\, a_{ij} - b_{ij} \,|\,]$$

$$[a_{ij}] \qquad [\, b_{ij}]$$

$$\begin{bmatrix} 0 & 12 & 142 & 255 \\ 1 & 6 & 40 & 254 \\ 24 & 0 & 20 & 255 \\ 30 & 2 & 10 & 240 \end{bmatrix} \quad \begin{bmatrix} 14 & 11 & 9 & 253 \\ 3 & 5 & 39 & 254 \\ 11 & 1 & 19 & 255 \\ 18 & 2 & 11 & 256 \end{bmatrix}$$

$$a_{ij} - b_{ij} \qquad c_{ij} = |\, a_{ij} - b_{ij} \,|$$

$$\begin{bmatrix} -14 & 1 & 133 & 2 \\ -2 & 1 & 1 & 0 \\ 13 & -1 & 1 & 0 \\ 12 & 0 & -1 & -16 \end{bmatrix} \quad \begin{bmatrix} 14 & 1 & 133 & 2 \\ 2 & 1 & 1 & 0 \\ 13 & 1 & 1 & 0 \\ 12 & 0 & 1 & 16 \end{bmatrix}$$

Kompresja obrazów

Cele kompresji:
- archiwizacja,
- przesyłanie.

Stopień kompresji obrazu

Definicja:

$$SK = \frac{KP}{KW}$$

SK - stopień kompresji obrazu.

KP [bajt] - obszar pamięci zajmowany przez ***kod pierwotny*** obrazu.

Kod pierwotny - reprezentacja rastrowa lub wektorowa.

KW [bajt] - obszar pamięci zajmowany przez ***kod wynikowy*** obrazu.

Kompresja bezstratna (*lossless compression*)

Własność: $\rho\left(\underline{x}^{\mu}, \underline{x}^{\nu}\right) = 0$

$\underline{x}^{\mu}$ - wektor reprezentujący obraz pierwotny

$\underline{x}^{\nu}$ - wektor reprezentujący obraz odtworzony (zrekonstruowany)

1. Kompresja obrazów z obszarami o jednolitej jasności:

- *Kodowanie ciągów identycznych symboli* (ciąg identycznych symboli - para zawiera 1 symbol i liczbę jego powtórzeń)

Przykład:

7,7,8,9,10,10,10,10,9,9,9,8,7,7,7

7(2), 8(1), 9(1), 10(4), 9(3), 8(1), 7(3).

Uwaga: dla obrazów o dużych obszarach o jednolitej jasności przegląd według krzywej Hilberta daje z reguły dłuższe ciągi identycznych symboli niż w przypadku przeglądu "linia po linii"

Praktyczna realizacja metody kodowania ciągów identycznych symboli:

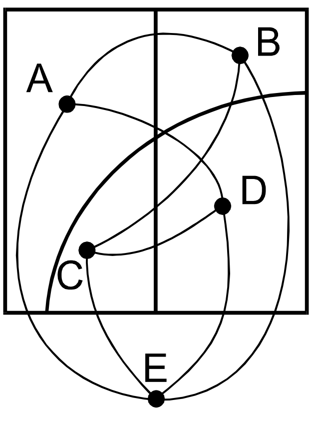
- formaty obrazowe: *.PCX, *.PIC

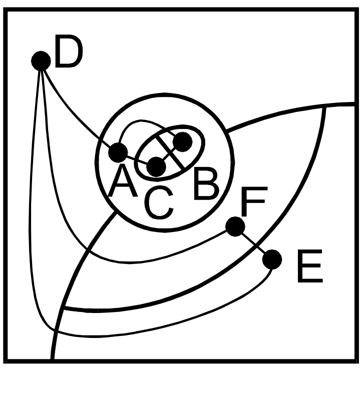
- Kodowanie *drzewiaste* - odwzorowanie obrazu w drzewo (szczególny przypadek grafu), czyli odwzorowanie piksli lub podzbiorów piksli w węzły (wierzchołki) drzewa. Przegląd siatki dyskretnej obrazu realizowany poprzez przegląd drzewa będącego odwzorowaniem tego obrazu.

Powszechnie stosowane są *drzewa czwórkowe* i *drzewa binarne*

Terminologia z zakresu teorii grafów

Graf - zbiór punktów lub ***wierzchołków (węzłów)*** połączonych liniami lub ***gałęziami***





Stopień wierzchołka - liczba gałęzi spotykających się w nim.

Graf ***zorientowany*** - część lub wszystkie gałęzie są ukierunkowane.

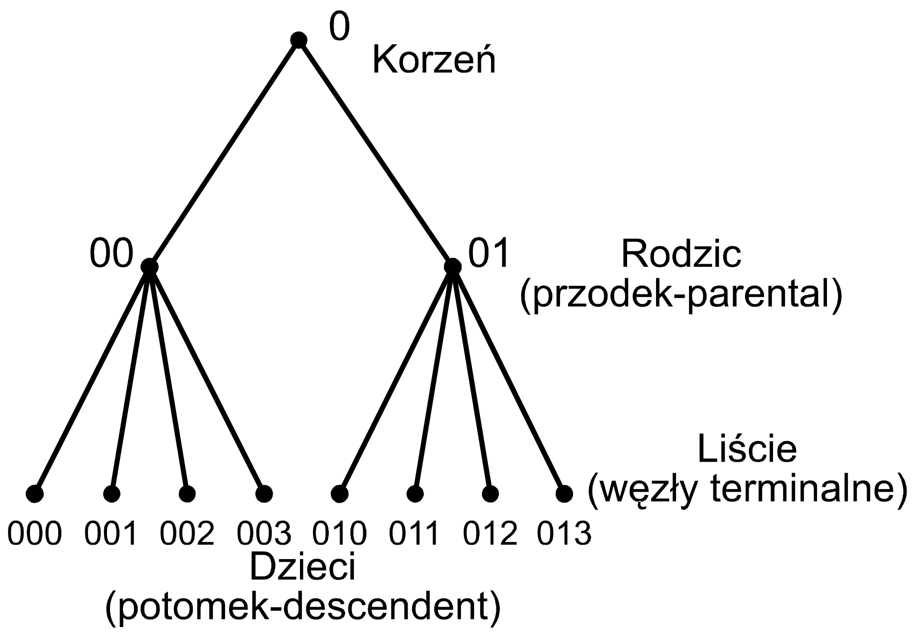
Stopień ***zewnętrzny*** lub ***wewnętrzny*** wierzchołka - liczba gałęzi skierowanych od i do wierzchołka.

Droga od wierzchołka A do B - ciąg gałęzi.

Droga ***cykliczna*** - droga, której początek i koniec pokrywają się.

Graf jest ***spójny*** $\Leftrightarrow$ istnieje droga między dowolną parą wierzchołków.

Drzewo - graf spójny bez dróg cyklicznych; poziomy: górne, dolne; stopień wierzchołka: górny i dolny.



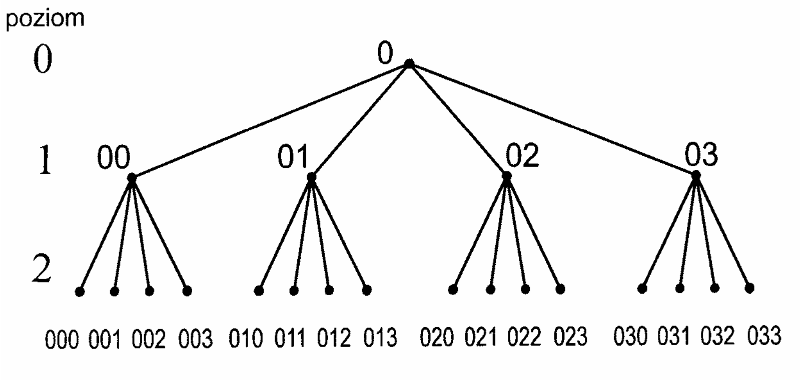
Podgraf grafu G - graf zawierający część lub wszystkie wierzchołki G oraz część lub wszystkie jego gałęzie i nie ma innych wierzchołków lub gałęzi.
Drzewo ***rozpinające*** grafu spójnego G - podgraf G zawierający wszystkie wierzchołki G oraz liczbę gałęzi wystarczającą aby był spójny bez dróg cyklicznych.

<u>Odpowiedniki w odwzorowaniu obraz - graf</u>
Siatka dyskretna - graf
Wierzchołki grafu - elementy obrazu. Sruktura do przechowywania wierzchołków - ***stos.***
Gałęzie - łączą wierzchołki odpowiadające sąsiednim elementom.
Dane: współrzędne x, y oraz ich jasność.
Problem: znalezienie wierzchołków przyległych do danego wierzchołka.

<u>Typowy algorytm przeglądania grafu</u> - odpowiada algorytmowi przeglądania obrazu
- Manipulacja stosem.
- Usuwanie węzłów (manipulowanie kolejką).

Piramidy albo drzewa czwórkowe
(odwzorowanie obrazu w drzewo czwórkowe)

Obraz - postać macierzy kwadratowej A o wymiarach 2^n x 2^n.
Powtarzany rekursywnie *n* razy proces podziału A na **4 macierze** kwadratowe aż do osiągnięcia poziomu pojedynczego elementu obrazu. Przedstawienie podziału w postaci drzewa, którego wierzchołki (węzły) odpowiadają kwadratom.



Wszystkie wierzchołki oprócz liści (wierzchołków stopnia 1) są ***stopnia dolnego 4,*** stąd: drzewo czwórkowe.

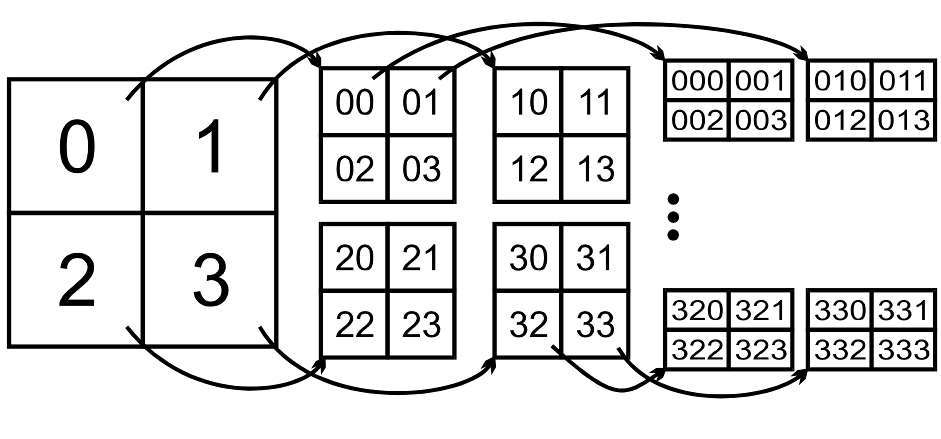
Długość etykiety pojedynczego elementu wynosi *n* (np. dla *n*=3 obraz 8x8)

Poziom k zawiera 4^k kwadratów. Stąd liczba wierzchołków drzewa:

$$N = \sum_{k=0}^{n} 4^k = \frac{4^{n+1} - 1}{3} \cong \frac{4}{3} \cdot 4^n$$

tzn. ok. 1/3 więcej wierzchołków niż elementów.
Tak więc w przypadku, gdy w odwzorowaniu obrazu w drzewo jeden wierzchołek drzewa odpowiada jednemu pikslowi, tzn. gdy nie ma obszarów (złożonych z więcej niż jednego piksla) o takiej samej jasności, występuje ***ekspansja*** obrazu - przeciwna do kompresji.

System adresowania kwadratów:



Typowe algorytmy

- Algorytm tworzenia drzewa czwórkowego z obrazu przeglądanego wiersz po wierszu
- Rekonstrukcja obrazu na podstawie drzewa czwórkowego.
  Wyświetlanie **zgrubne** np. w czasie $T_c/2$, gdzie T_c - całkowity czas odtworzenia obrazu. W reprezentacji macierzowej w czasie $T_c/2$ wyświetli się połowa obrazu.
- Kompresja obrazu za pomocą drzewa czwórkowego (przy dostatecznie dużych obszarach o jednolitej jasności).

Drzewa binarne

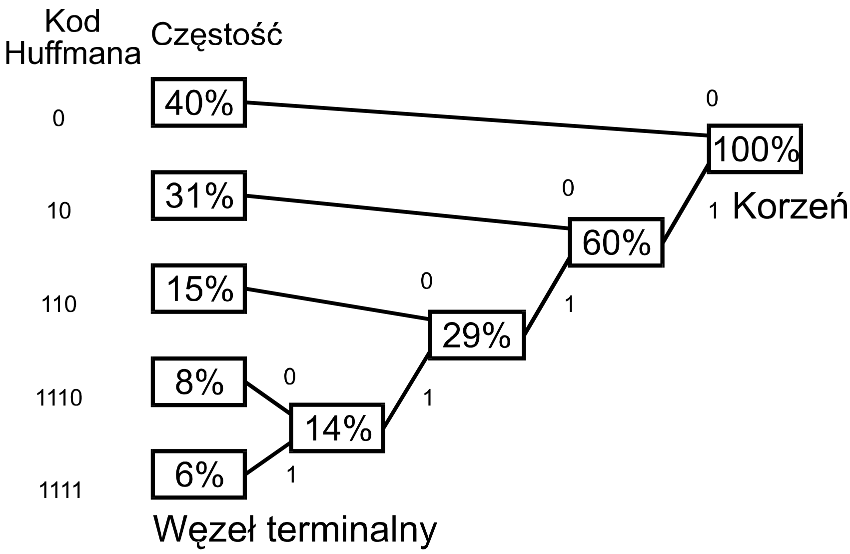
Zaleta: mniejsza zmiana rozdzielczości przy zmianie poziomu (dwukrotna) niż w przypadku drzewa czwórkowego (czterokrotna).
Jeżeli przy drzewach czwórkowych każda transmisja czterokrotnie zwiększa rozdzielczość, to dla drzew binarnych jedynie ją podwaja. Stąd, przy tej samej całkowitej ilości danych w obydwu metodach, dla drzew binarnych powstaje wcześniej obraz bardziej odpowiadający oryginałowi.

2. Kompresja obrazów z obszarami o niejednolitej jasności

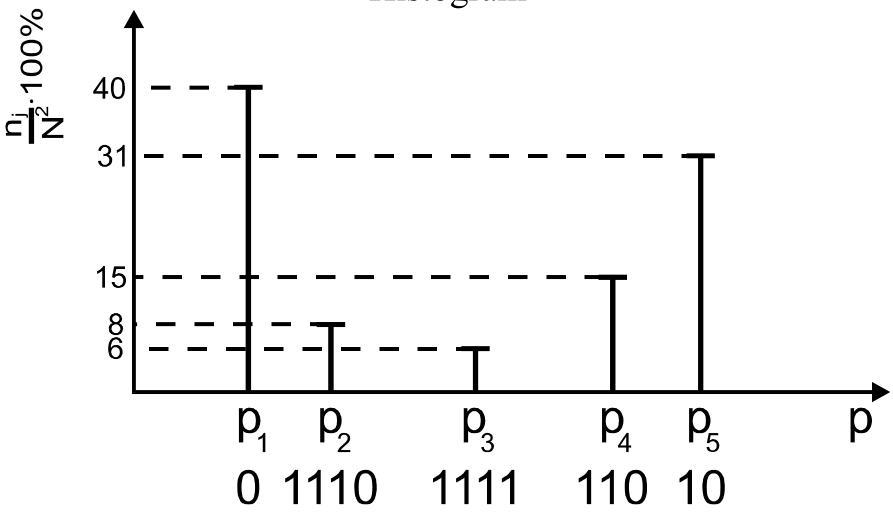
Kod Huffmana - kod o *zmiennej długości słowa*.
Sposób postępowania:

- Przypisanie każdemu poziomowi jasności częstości występowania pikseli o tym poziomie jasności (utworzenie histogramu).
- Wyszukanie 2 poziomów o najmniejszej częstości występowania i połączenie w jeden o częstości występowania równej sumie tych poziomów.

Schemat łączenia - *drzewo Huffmana.*



Histogram



n_j - liczba piksli o jasności (poziomie jasności, wartości) p_j,
N^2 - liczba piksli obrazu
p - poziom jasności

<u>Praktyczna realizacja</u> kodu Huffmana - format obrazowy *.TGA

<u>Wada kodu Huffmana</u> - konieczność przyłączania do zakodowanego obrazu biblioteki użytych kodów (odpowiadających zadanym poziomom jasności). Rozmiar biblioteki może przewyższyć redukcję rozmiaru obrazu.

<u>Ominięcie problemu:</u>

Sposób 1 - szy:
– Metoda adaptacyjna - wykorzystanie do kodowania każdego piksla biblioteki zbudowanej na podstawie częstości wystąpień wszystkich **już zakodowanych** pikseli; program dekodujący **zna** te piksle, stąd nie ma potrzeby przyłączania tablicy przekodowań.
Wada - duży koszt obliczeń - wyznaczanie kodów Huffmana NxN razy, tzn. *złożoność obliczeniowa* wynosi $O(N^2)$.

Sposób 2 - gi:
– Metoda wykorzystania *standardowej tablicy przekodowań* (tylko w przypadkach, gdy charakter kodowanych danych można dobrze przewidzieć). Krótszy proces kodowania (jednakrotny przegląd obrazu - tylko w trakcie rzeczywistego kodowania). Pierwszy krok - polegający na przeglądzie obrazu dla znalezienia częstości występowania poszczególnych poziomów jasności (histogram) zostaje pominięty.

Stopień kompresji wzrasta, gdy różnica częstości n_j wystąpień poszczególnych poziomów jasności p_j też wzrasta.
Zwiększenie nierówności częstości wystąpień poprzez zastąpienie poziomów jasności piksli $f_{m,n}$ odpowiednimi *wyrażeniami różnicowymi:*

$$\varepsilon_{m,n} = f_{m,n} - f_{m,n-1} \qquad (1)$$
$$\varepsilon_{m,n} = f_{m,n} - f_{m\text{-}1,n} - (f_{m,n-1} - f_{m\text{-}1,n\text{-}1}) \qquad (2)$$

W przypadku (2) nastąpiło zwiększenie nierówności częstości wystąpień w porównaniu z przypadkiem (1)

Położenie piksli:

$$\begin{array}{c|cc} & & \xrightarrow{\quad m \quad} \\ \hline & f_{m\text{-}1,n\text{-}1} & f_{m,n\text{-}1} \\ & f_{m\text{-}1,n} & f_{m,n} \\ n \downarrow & & \end{array}$$

Metody słownikowe

Kodowanie *ciągów symboli* (piksli) za pomocą odwołań do *słownika* zawierającego takie ciągi.
Stopień kompresji (SK) rośnie, gdy **długość ciągu piksli** możliwych do zastąpienia *indeksem* do słownika rośnie.
Podział metod słownikowych:
- *statyczne (*słownik nie zmienia się w trakcie kodowania),
- *adaptacyjne* (słownik zmienia się w trakcie kodowania)

Przykład metody statycznej: słownik zawiera wszyskie możliwe ciągi o długościach równych liczbie piksli w **całym obrazie**; wtedy:

$$\boxed{L=2^{B(NxN)}}$$

gdzie: L - liczba możliwych ciągów (tzn. liczba indeksów słownika),
B - liczba bitów określajacych wartość piksla,
NxN - liczba piksli w całym obrazie

Wtedy: KP = B(NxN), KW = L $\Rightarrow$ SK < 1 , (ekspansja)
(np. dla B=8, N=512: $L = 2^{8(512x512)} \approx 10^{600000}$ >> 8 (512x512), tzn. SK<<1)
Wniosek: należy stosować ciągi o mniejszej liczbie wyrazów.
Problem: Dla obrazu o parametrach : B=8, N=512 znaleźć taką wartość n<NxN, dla której SK=1

Kompresja stratna (*lossy compression*)

Własność: $\rho\left(\underline{x}^{\mu}, \underline{x}^{\nu}\right) \neq 0$

$\underline{x}^{\mu}$ - wektor reprezentujący obraz pierwotny

$\underline{x}^{\nu}$ - wektor reprezentujący obraz odtworzony (zrekonstruowany)

Kompresja stratna: - uzyskiwany jest większy stopień kompresji (SK) niż przy stosowaniu kompresji bezstratnej (*lossless compression*).

Wybrane metody kompresji stratnej:
- *kodowanie różnic*,
- *kodowanie blokowe*

- Kodowanie różnic: $\varepsilon_{m,n} = f_{m,n} - f_{m,n-1}$

gdzie:

$f_{m,n}$ - poziom jasności piksla o współrzędnych *m,n,*

$f_{m,n-1}$ - poziom jasności piksla (**kolejnego**) o współrzędnych *m,n-1.*

Położenie piksli:

$$\begin{array}{l|ll} & & \xrightarrow{\quad m \quad} \\ & f_{m-1,n-1} & f_{m,n-1} \\ & f_{m-1,n} & f_{m,n} \\ n \downarrow & & \end{array}$$

Przykład:

Obraz pierwotny : L = 255 (M = 256)

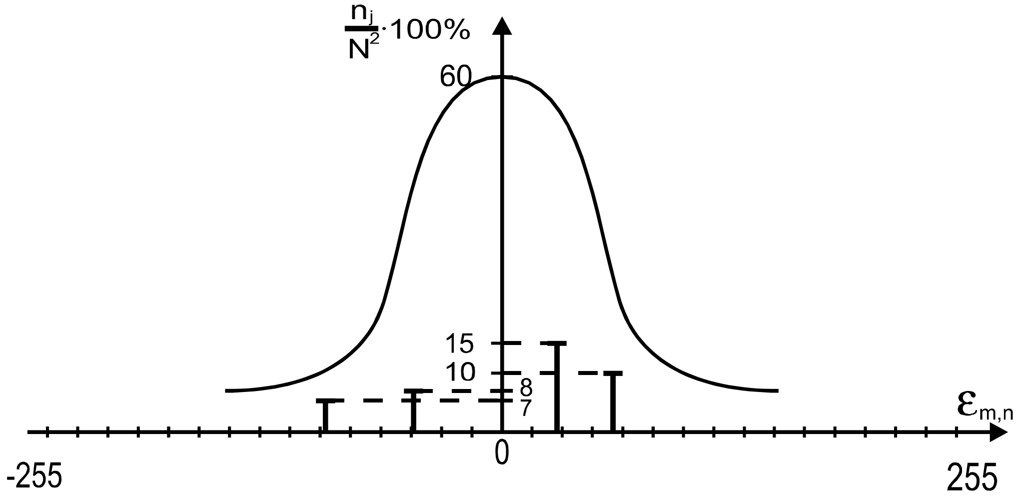
- kodowanie wartości piksli: $l = 0,1,2,3,.....,255$;

zajętość pamięci: 8 bitów/piksel,

- kodowanie różnic pomiędzy wartościami kolejnych piksli:

$\varepsilon_{m,n} = -255, -254, \ldots, 0, \ldots, 254, 255$; zajętość pamięci: 9 bitów /piksel.

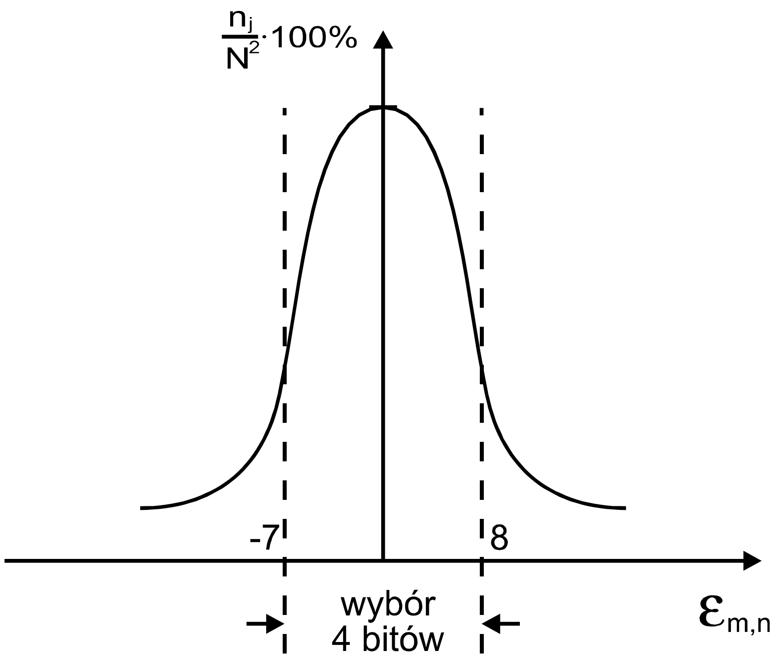
Histogram różnic $\varepsilon_{m,n}$ pomiędzy wartościami (poziomami jasności) **kolejnych** piksli:



Histogram różnic ma charakter *krzywej Gaussa* z maksimum w okolicy zera.

Realizacja kompresji: zakodowanie **najczęściej** występujących różnic.

Histogram różnic z zaznaczonym obszarem najczęściej występujących różnic:



Kryteria wyboru obszaru:
- wymagania na wielkość stopnia kompresji (SK),
-wymagania na *dokładność rekonstrukcji* (określoną wielkością "ρ")

KP = 9, KW = 4

SK = KP/KW = 9/4 =
= 2,25

Obraz zakodowany: $\varepsilon_{m,n} = -7, -6, \ldots, 7, 8$ - 16 symboli zamiast 512

Niepożądany efekt: zależnie od rodzaju obrazu - większe lub mniejsze zamazywanie (*blurring*) ostrych krawędzi.

• <u>Kodowanie blokowe</u>

Podział obrazu na jednakowe bloki, najczęściej 4x4 piksele. Obliczenie dla każdego bloku *średniej arytmetycznej* jasności. Podział pikseli na dwie grupy: a) o jasności większej lub równej jasności średniej, b) mniejszej niż jasność średnia. Obliczenie nowej jasności średniej dla każdej z grup (*wartość górna* dla (a) i *wartość dolna* dla (b)). Przypisanie wszystkim pikslom danej grupy obliczonej jasności średniej (górnej lub dolnej), stąd blok zostaje zakodowany jako *mapa bitowa* określająca podział na grupy, plus dwie wartości jasności.

13	11	10	13
8	9	12	15
6	8	11	14
4	7	9	12

Średnia=10

$\longrightarrow$

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

Wartość górna=12
Wartość dolna=7

<u>Przykład:</u>

Obraz o 256 poziomach jasności, bloki o wymiarach 4x4 piksele.
Obszar pamięci zajmowany przez blok przed kompresją: KP = 4x4x8 = 128 bitów
Obszar pamięci zajmowany przez blok po kompresji: KW = 4x4x1 + 2x8=32 bity,
gdzie: 4x4x1 - obszar pamięci zajmowany przez mapę bitową,
2x8 - obszar pamięci zajmowany przez dwie wartości jasności

stąd: SK = KP/KW = 4

Niepożądany efekt: zwiększenie rozmiaru bloku powoduje coraz wyraźniejsze ukazywanie się siatki podziału na bloki.

Praca domowa

Przykład 1

Dany jest obraz w postaci tablicy [$f(m,n)$], M=16. Znaleźć tablicę wyrażeń różnicowych $[\varepsilon(m,n)]$. Sporządzić histogramy dla obu tablic oraz określić rozmiary kodu pierwotnego i wynikowego (rozważyć przypadek (1) i (2) dla kompresji bezstratnej).

$[f(m,n)]$

0	9	1	6	5
2	7	3	8	7
4	5	5	10	9
6	3	7	12	11
8	1	9	14	13

$[\varepsilon(m,n)]$

$$\varepsilon_{m,n} = f_{m,n} - f_{m,n-1} \qquad (1)$$

$$\varepsilon_{m,n} = f_{m,n} - f_{m\text{-}1,n} - (f_{m,n-1} - f_{m\text{-}1,n\text{-}1}) \qquad (2)$$

$$\begin{array}{l|ll} & & \xrightarrow{\quad m} \\ & f_{m-1,n-1} & f_{m,n-1} \\ & f_{m-1,n} & f_{m,n} \\ n\downarrow & & \end{array}$$

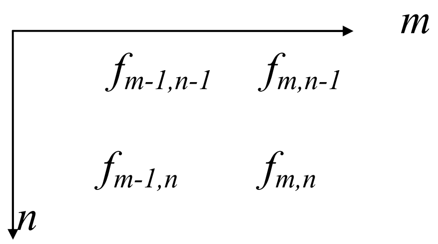
Przykład 2

Dany jest histogram obrazu o rozmiarach NxN:

0	0	0	0	5	0	0	5	5	5	0	0	0	0	0	5
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Znaleźć takie dwa obrazy [*f(m,n)*] odpowiadające podanemu histogramowi, aby po zastąpieniu piksli każdego z tych obrazów wyrażeniami różnicowymi $[\varepsilon(m,n)]$ odpowiedni kod Huffmana zajmował: a) jak najmniejszy (SK>1), b) jak największy (SK<1) obszar pamięci. Uwaga: współczynnik kompresji SK odnosi się do reprezentacji rastrowej obrazu. Wyliczyć wartości dla obu obrazów przy zastosowaniu najkrótszego kodu dla obrazu pierwotnego i różnicowego (16 wartości na 4 bitach, 32 na 5 bitach).
Rozważyć przypadki (1) i (2) sąsiedztwa.

$$\varepsilon_{m,n} = f_{m,n} - f_{m,n-1} \qquad (1)$$
$$\varepsilon_{m,n} = f_{m,n} - f_{m\text{-}1,n} - (f_{m,n-1} - f_{m\text{-}1,n\text{-}1}) \qquad (2)$$



R O Z W I Ą Z A N I E
Kod Huffmana jest tym efektywniejszy im mniej wartości różnic.
<u>Przypadek 1</u>

<u>Obraz I</u>	<u>Kod różnicowy obrazu I</u>
4 4 4 4 4	4 4 4 4 4
7 7 7 7 7	3 3 3 3 3
8 8 8 8 8	1 1 1 1 1
9 9 9 9 9	1 1 1 1 1
15 15 15 15 15	6 6 6 6 6

KP=5*5*4 bity=100 bitów.   Kod obrazu różnicowego:
5*5*5bitów = 125 bitów

Histogram obrazu różnicowego:

-15.. 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0.. 0	10	0	5	5	0	5	0	0	0	0	0	0	0	0	0

Odpowiednie kody (dla niezerowych 4 poziomów szarości podstawa kodu wynosi 3):

- - 0 - 10 110 - 111 - ..

Zakodowania legendy:

4*5 bitów (na histogram) + 9 bitów (na kody) = 20 + 9 = 29

Zakodowania kodu różnicowego:

10*1 bit = 10 kod 1
5*2 bity = 10 kod 3
5*3 bity = 15 kod 4
5*3 bity = 15 kod 6

Suma w bitach: 50 bitów

KW_I = obraz + legenda = 50 + 29 = 79 bitów

$SK_I = KP / KW_I$ = 100 / 79 = 1,26582~1,266

Obraz II

15	7	9	8	4
4	8	9	7	15
7	15	4	9	8
9	4	15	8	7
8	9	7	15	4

KP=5*5*4bity = 100 bitów

Kod różnicowy obrazu II

15	7	9	8	4
-11	1	0	-1	11
3	7	-5	2	-5
2	-11	11	-1	-1
-1	5	-8	7	-3

Kod obrazu różnicowego:
5*5*5bitów = 125 bitów

Histogram obrazu różnicowego:

-11 ...	-8	-5	-3	-1	0	1	2	3	4	5	7	8	9	11	15
2	1	2	1	4	1	1	2	1	1	1	3	1	1	2	1

Odpowiednie kody (dla niezerowych 16 poziomów szarości podstawa kodu wynosi 15):

Zakodowania kodu różnicowego:

-11	2	110	2*3 = 6
-8	1	1111110	1*7 =14
-5	2	1110	2*4 = 8
-3	1	11111110	1*8 = 8
-1	4	0	4*1 = 4
0	1	111111110	1* 9= 9
1	1	1111111110	1*10=10
2	2	11110	2* 5=10
3	1	11111111110	1*11=11
4	1	111111111110	1*12=12
5	1	1111111111110	1*13=13
7	3	10	3* 2= 6
8	1	11111111111110	1*14=14
9	1	111111111111110	1*15=15
11	2	111110	2* 6=12
15	1	111111111111111	1*15=15

Suma w bitach: 167 bity

Zakodowania legendy: 16*5 bitów (kod histogramu) + 135 bitów = 80+135 = 215 bitów
Uwaga: Sama legenda zawiera więcej bajtów niż kod obrazu pierwotnego.

KW_{II} = obraz + legenda = 167 + 215 = 382 bitów
SK_{II} = KP / KW_{I} = 100 / 382 = 0,26178~0,262

ODPOWIEDŹ:
SK_{I} ~1,266
SK_{II}~0,262

Przykład 3

Na podstawie histogramu przedstawionego za pomocą tablicy LUT

2	9	7	14	0	0	0	0	0	0	0	0	0	3	18	11
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

utworzyć dwa obrazy [p1(i,j)] i [p2(i,j)] o rozmiarach NxN takie, aby po ich poddaniu kompresji wykorzystującej kodowanie blokowe (bloki o rozmiarach 4x4) zostały spełnione następujące warunki:

a) obraz pierwotny [p1(i,j)] i obraz wynikowy (odtworzony) [q1(i,j)] różnią się między sobą (kompresja stratna),
b) obraz pierwotny [p2(i,j)] i obraz wynikowy (odtworzony) [q2(i,j)] nie różnią się między sobą (kompresja bezstratna).

Dla każdego z przypadków obliczyć:

- średnią arytmetyczną poziomów szarości piksli każdego z bloków,
- średnie arytmetyczne dla każdej z dwóch grup piksli w bloku (wartość dolna i wartość górna),
- kody o minimalnej długości pierwotny (KP) i wynikowy (KW) oraz stopień kompresji (SK),
- różnicę pomiędzy obrazem pierwotnym i wynikowym.

<u>Uwaga: </u>Żaden z bloków 4x4 obrazu pierwotnego nie może składać się z jednakowych piksli

Literatura podstawowa:

M. Doros, Przetwarzanie obrazów, Skrypt WSISIZ, Warszawa 2005.

(w szczególności rozwiązać Zadania 1, 2, 3, 7, 9, 11 (str.133-139))

Literatura uzupełniająca:

W.Skarbek, Metody Reprezentacji Obrazów Cyfrowych, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993.