

# Algorytmy kodowania predykcyjnego

1. Zasada kodowania
2. Algorytm JPEG-LS
3. Algorytmy CALIC, LOCO-I
4. Algorytmy z wielokrotną rozdzielczością.  
Progresywna transmisja obrazów

## Kompresja obrazów - zestawienie

Obraz	Huffman	Huffman różnice	St. kompr. Huffman	St. kompr. Huffman różnice
1_ch	57 504	32 968	1.14 : 1	1.99 : 1
2_ch	61 430	38 541	1.07 : 1	1.70 : 1
Ziemia	40 543	33 880	1.62 : 1	1.93 : 1
Miasto	58 374	52 643	1.12 : 1	1.24 : 1

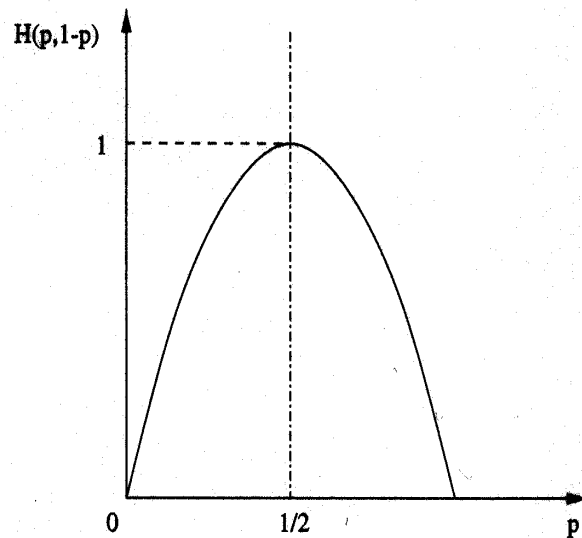
Algorytmy kodujące różnice wyraźnie lepsze.

Jaki jest tego powód ?

# Zasada kodowanie predykcyjnego

- ❑ Im bardziej zróżnicowane prawdopodobieństwa wystąpienia symboli tym większą kompresję można uzyskać
- ❑ Przykład 1: alfabet dwuznakowy, prawdopodobieństwa  $p$ ,  $q$ ;  $p + q = 1$ .

$$\text{Entropia } H = -p \log_2 p - (1-p) \log_2 (1-p)$$



Entropia osiąga maksimum gdy

$P = q = 0.5$ , minimum gdy  $p$  lub

$q \rightarrow 1$

## Zasada kodowanie predykcyjnego – c.d.

Przykład 2 – alfabet ośmioznakowy:

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	H
1/8	1/8	1/8	1/8	1/8	1/8	1/8	1/8	3
1/16	1/16	3/16	3/16	3/16	3/16	1/16	1/16	2.811
1/128	4/128	16/128	43/128	43/128	16/128	4/128	1/128	2.228
0.001	0.01	0.05	0.439	0.439	0.05	0.01	0.001	1.626

Im bardziej zróżnicowane prawdopodobieństwa, tym mniejsza entropia, czyli mniejsza średnia bitowa.

# Zasada kodowanie predykcyjnego – c.d.

## □ Jak osiągnąć taką sytuację?

- Użyć odwracalnego przekształcenia transformującego ciąg wejściowy na ciąg o zróżnicowanych prawdopodobieństwach
- Używać innego rozkładu prawdopodobieństwa dla każdego symbolu, by zwiększyć prawdopodobieństwo wystąpienia kodowanego symbolu (kontekst)
- Zadbać, by dekodery bez żadnej dodatkowej informacji „wiedział” jakiej transformacji lub jakiego rozkładu użyto → przekształcenie odwrotne lub rozkład będzie do odtworzenia bez żadnych dodatkowych informacji (np. na podstawie znajomości wcześniej zakodowanych elementów).

## □ Tak działające algorytmy = algorytmy kodowania predykcyjnego.

# Kodowanie predykcyjne – przykład

## ❑ Przykład – kodowanie różnic:

- Predykcja :  $\underline{x}_n = x_{n-1}$
- Transformacja:  $x_n \rightarrow e_n = x_n - \underline{x}_n$
- Rozkład różnic bardziej zróżnicowany – kompresja
  - Przykład: plik 2\_ch, fragment (8 x 16 pikseli)

Rozkład oryginalnych zmiennych:

66	67	68	69	70	71	72	73	74
3	13	11	29	25	27	8	8	6

Rozkład różnic

-3	-2	-1	0	1	2	3	4	42	43	44	45
2	5	24	44	27	12	4	1	1	2	2	3

Dekodowanie – odkodowanie  $e_n$  , następnie wyliczenie  $x_n$  – transformacja odwracalna.

# Algorytm JPEG-LS

- ❑ Dla każdego piksela – liniowa predykcja w oparciu o wartości 3 sąsiednich pikseli
- ❑ 8 możliwych schematów predykcji
- ❑ Błędy predykcji kodujemy algorytmem entropijnym

	C	B		
	A	X		

Nr	Formuła
P0	bez predykcji
P1	A
P2	B
P3	C
P4	$A + B - C$
P5	$A + (B - C)/2$
P6	$B + (A - C)/2$
P7	$(A + B)/2$

## Uwaga:

(0,0) – zawsze P0

(0,i) – zawsze P1

(i,0) – zawsze P2

## Algorytm JPEG-LS – c.d.

Wyniki:

Obraz	P0	P1	P2	P3	P4	P5	P6	P7
1_ch	53 431	37 220	31 559	38 261	31 055	<b>29 742</b>	33 063	32 179
2_ch	58 306	41 298	37 126	43 445	<b>32 429</b>	33 463	35 965	36 428
Ziemia	38 295	32 295	<b>32 137</b>	34 089	33 570	33 057	33 072	32 672
Miasto	56 061	<b>48 818</b>	51 283	53 909	53 771	53 520	52 542	52 189

- Na ogół – dla różnych obrazów optymalne różne metody
- Kodowanie predykcyjne wyraźnie lepsze od metod słownikowych
- Jeżeli nie kodujemy w czasie rzeczywistym – próbujemy wszystkie schematy, wybieramy optymalny (informacja w nagłówku)
- Dla grafiki komputerowej metody słownikowe mogą być lepsze



# Algorytm CALIC

- ❑ CALIC = Context Adaptive Lossless Image Compression
- ❑ Zaproponowany w 1995 jako propozycja nowego standardu ISO/JPEG
- ❑ Używa kontekstu do określenia predykcji jak też rozkładu prawdopodobieństwa w czasie kodowania
- ❑ Najlepsza z istniejących metod bezstratnej kompresji obrazów naturalnych
- ❑ Przy kodowaniu każdego punktu korzysta z informacji o otoczeniu:

		NN	NNE
	NW	N	NE
WW	W	X	

X – kodowany piksel

Pozostałe piksele znane koderowi i dekodekowi w momencie kodowania X

# Algorytm CALIC - schemat

## Główne etapy algorytmu:

- ❑ Predykcja wstępna (wykrycie krawędzi – gradienty)
- ❑ Korekta predykcji wstępnej zależna od kontekstu
- ❑ Obliczenie błędu predykcji; odwzorowanie na zakres równy naturalnej zmienności piksela
- ❑ Kodowanie błędu predykcji w sposób zależny od kontekstu

## Predykcja wstępna

$$d_h = |W - WW| + |N - NW| + |NE - N|$$

$$d_v = |W - NW| + |N - NN| + |NE - NNE|$$

$d_v \gg d_h \rightarrow$  krawędź pozioma, najlepsza aproksymacja X to W

$d_h \gg d_v \rightarrow$  krawędź pionowa, najlepsza aproksymacja X to N

Nie ma wyraźnych różnic  $\rightarrow$  aproksymujemy średnią ważoną otoczenia

## Algorytm CALIC – predykcja

IF ( $dv - dh > 80$ )	{ ostra krawędź pozioma }
$PX = W$	
ELSE IF ( $dh - dv > 80$ )	{ ostra krawędź pozioma }
$PX = N$	
ELSE {	
$PX = (W+N)/2 + (NE - NW)/2$	
IF ( $dv - dh > 32$ )	{ krawędź pozioma }
$PX = (PX + W)/2$	
ELSE IF ( $dv - dh > 8$ )	{ słaba krawędź pozioma }
$PX = (3 PX + W)/4$	
ELSE IF ( $dh - dv > 32$ )	{ krawędź pionowa }
$PX = (PX + N)/2$	
ELSE IF ( $dh - dv > 8$ )	{ słaba krawędź pionowa }
$PX = (3 PX + N)/4$	
}	

# Algorytm CALIC – korekta predykcji

## Korekta zależna od kontekstu

- Predykcja gradientowa nie usuwa całej redundancji
- Musimy uwzględnić dodatkowe informacje o wzajemnych relacjach między X a otoczeniem – kontekst
- Definicja kontekstu:  
 $Y = \{N, W, NW, NE, NN, WW, 2N - NN, 2W - WW\}$   
Kwantyzacja:  $Y_i < PX \rightarrow Y_i = 0$  w przeciwnym razie  $Y_i = 1$   
To daje 144 możliwe wektory binarne  
Dodatkowo:  $\delta = dh + dv + 2|N - PN|$ , kwantyzacja na przedziały  
To daje w sumie  $4 * 144 = 576$  możliwych kontekstów
- Podczas kodowania zapamiętujemy błąd kodowania dla każdego piksela, wyznaczamy średni błąd w danym kontekście – to pozwala dokonać korekty predykcji zależnej od kontekstu:  
 $PX = PX + \langle e(Y, \delta) \rangle$

# Algorytm CALIC – kodowanie błędu

## Wyznaczenie błędu predykcji i kodowanie

- Wyznaczymy błąd predykcji:  $e = PX - X$
- $X, PX$  – liczby z zakresu  $0 \dots M - 1 \rightarrow e$  może być pomiędzy  $-(M - 1)$  a  $M - 1$ .
- Przed zakodowaniem algorytm dokonuje transformacji  $e$  do przedziału  $0 \dots M - 1$
- W trakcie kodowania bierzemy pod uwagę kontekst wyznaczony tym razem przez  $\delta$  – kwantyzowane w 8 przedziałach (osiem kontekstów kodowania)
- Od kontekstu kodowania zależy wybór alfabetu (alfabety o małej długości, kodowanie rekurencyjne).

## Podsumowanie:

CALIC to dobry, ale bardzo złożony algorytm – potrzeba znalezienia podobnie skutecznego, ale prostszego schematu.

# Algorytm LOCO-I

- ❑ LOCO-I = Low Complexity Lossless Compression for Images
- ❑ Zaproponowany w 1996 przez grupę z HP
- ❑ Jest podstawą dla aktualnej wersji JPEG-LS
- ❑ Schemat podobny do CALIC:
  - Wstępna predykcja
  - Wyznaczenie kontekstu, korekta predykcji
  - Obliczenie błędu predykcji, odwzorowanie na naturalny zakres
  - Kodowanie (dynamiczne kody Golomba)
- ❑ Informacja o otoczeniu:

NW	N	NE
W	X	

X – kodowany piksel

## Algorytm LOCO-I – c.d.

### Wstępna predykcja

Zadana przez algorytm:

IF  $NW \geq \max(W, N)$

$PX = \max(W, N)$

ELSE

{

    IF  $NW \leq \min(W, N)$

$PX = \min(W, N)$

    ELSE

$PX = N + W - NW$

}

Wstępna predykcja jest poprawiana w zależności od kontekstu opartego o:

$$D_1 = NE - N$$

$$D_2 = N - NW$$

$$D_3 = NW - W$$

# Algorytm LOCO-I – c.d.

## Określenie kontekstu

- Wybór trzech dodatnich współczynników  $T1 < T2 < T3$  używanych do kwantyzacji składowych  $D_i$

- Wyznaczenie składowych wektora kontekstu  $Q$

$D_i \leq -T3$	$\Rightarrow$	$Q_i = -4$
$-T3 < D_i \leq -T2$	$\Rightarrow$	$Q_i = -3$
$-T2 < D_i \leq -T1$	$\Rightarrow$	$Q_i = -2$
$-T1 < D_i < 0$	$\Rightarrow$	$Q_i = -1$
$D_i = 0$	$\Rightarrow$	$Q_i = 0$
$0 < D_i \leq T1$	$\Rightarrow$	$Q_i = 1$
$T1 < D_i \leq T2$	$\Rightarrow$	$Q_i = 2$
$T2 < D_i \leq T3$	$\Rightarrow$	$Q_i = 3$
$T3 < D_i$	$\Rightarrow$	$Q_i = 4$

- Utożsamienie  $Q$  i  $-Q$  gdy  $Q_1 < 0$  (wtedy  $SIGN = -1$ ). To daje  $(9*9*9 + 1) / 2 = 365$  różnych kontekstów



# Algorytm LOCO-I – c.d.

## Korekta predykcji zależne od kontekstu

$$PX = PX + \text{SIGN} * \langle e(Q) \rangle$$

## Wyznaczenie błędu predykcji, transformacja i kodowanie

- $e = X - PX$
- Transformacja odchylenia
$$\begin{aligned} e < -M/2 &\Rightarrow e = e + M \\ e > M/2 &\Rightarrow e = e - M \end{aligned}$$
- Kodowanie odchylenia – dynamiczne kodowanie Golomba (optymalne dla kodowania dużych liczb o rozkładzie geometrycznym).

## Metody predykcyjne - podsumowanie

Obraz	Stary JPEG		Nowy JPEG		CALIC	
1_ch	31 055	2.11:1	27 339	2.40:1	26 433	2.48:1
2_ch	32 429	2.02:1	30 344	2.16:1	29 213	2.24:1
Ziemia	32 137	2.04:1	26 088	2.51:1	25 280	2.59:1
Miasto	48 818	1.34:1	50 765	1.29:1	48 249	1.36:1

- Metody predykcyjne – najlepsze metody bezstratnej kompresji obrazów
- LOCO-I – wyraźnie lepszy od „starego” JPEG, nieznacznie słabszy od CALIC – nowy standard JPEG-LS
- Maksymalne stopnie kompresji < 3:1 – konieczność poprawy – kompresja stratna

# Techniki z wieloma poziomami rozdzielczości

- Metody, w których tworzone i kodowane są kolejno reprezentacje obrazów o różnych poziomach rozdzielczości
- Najbardziej znana metoda: HINT (Hierarchical Interpolation)

$\Delta$	•	X	•	$\Delta$	•	X	•	$\Delta$
•	*	•	*	•	*	•	*	•
X	•	◦	•	X	•	◦	•	X
•	*	•	*	•	*	•	*	•
$\Delta$	•	X	•	$\Delta$	•	X	•	$\Delta$
•	*	•	*	•	*	•	*	•
X	•	◦	•	X	•	◦	•	X
•	*	•	*	•	*	•	*	•
$\Delta$	•	X	•	$\Delta$	•	X	•	$\Delta$

Kolejność kodowania:

- punkty  $\Delta$
- punkty  $\circ$
- punkty X
- punkty \*
- punkty •

# Progresywna transmisja obrazów

- Polega na przesyłaniu przez sieć obrazów z coraz lepszymi rozdzielczościami
- Stosowana przy konieczności przeglądania dużych obrazów przez sieć o niskiej przepustowości
- Pozwala na szybsze zorientowanie się w zawartości obrazu

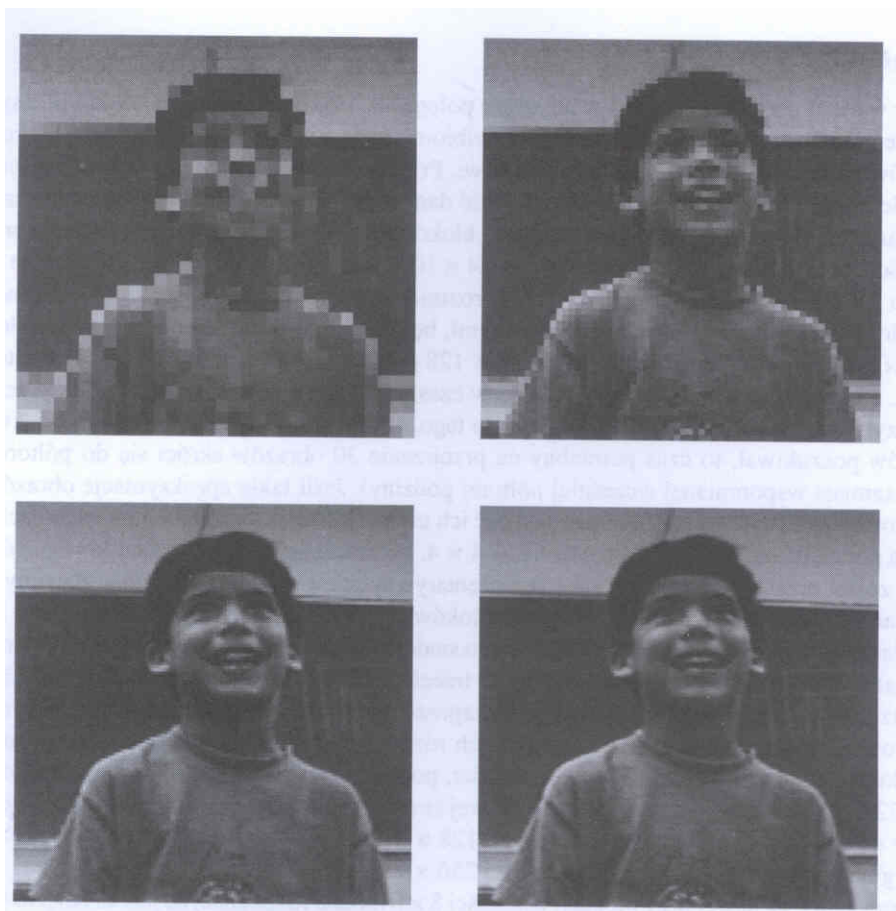
## Przykład

Użytkownik ma znaleźć 1 spośród 30 obrazów o rozdzielczości 1024 x 1024 poprzez sieć o przepustowości 56 kbps. Przejrzenie całej bazy –  $8 \times 1024 \times 1024 / 56 \text{ 000 sek} > 1 \text{ h}$ .

1 przybliżenie – bloki 8 x 8 reprezentowane jedną wartością – jeden obraz transmitowany przez  $< 2.5 \text{ sek}$ , pełna baza  $< 1.5 \text{ min}$

2 przybliżenie – płoki 4 x 4 reprezentowane jedną wartością – by poprawić wyświetlenie jednego obrazu trzeba dodatkowo 7 sek, dla całej bazy  $\sim 4 \text{ min}$ .

## Progresywna transmisja obrazów - przykład



# Progresywna transmisja obrazów - przykład

