

Run-Length kódovanie (Run Length Encoding – RLE)

- Existuje veľa verzií
- Použitie: kódovanie bitových rovín, čiernobiele obrázky, ...

Základná verzia:

- Na začiatku postupnosti sa predpokladajú nuly a kóduje sa ich počet, následne sa kóduje počet jednotiek, následne núl, atď.
- Počty sa kódujú fixným počtom bitov

Príklad:

Vstupná postupnosť 30 bitov: 0 0 0 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1

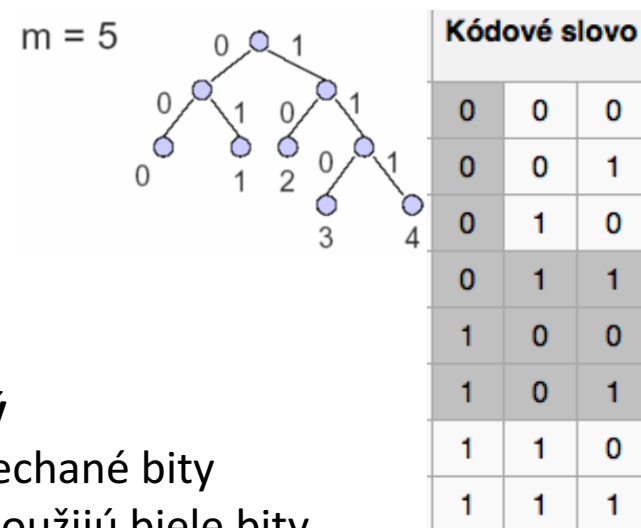
- Použijeme 3 bity na kódovanie počtov
 - Počty: 3, 5, 1, 2, 6, 5, 1, 7
 - bity: 011 101 001 010 110 101 001 111
 - Výsledný kód má 24 bitov.
- Použijeme 4 bity na kódovanie počtov
 - Počty sa nezmenia
 - bity: 0011 0101 0001 0010 0110 0101 0001 0111
 - Výsledný kód má 32 bitov
- Použijeme 2 bity na kódovanie počtov
 - Počty: 3, 3, 0, 2, 1, 2, 3, 0, 3, 3, 0, 2, 1, 3, 0, 3, 0, 1
 - bity: 11 11 00 10 01 10 11 00 11 11 00 10 01 11 00 11 00 01
 - Výsledný kód má 36 bitov

Run-Length kódovanie (Run Length Encoding – RLE) 2

- Ako zabezpečíme optimálnu kódovú dĺžku?
- Prečo nepoužiť variabilnú dĺžku?
 - Huffmanov kód
 - Exp. Golombov / Golombov kód
 - zvážime použitie najmä vtedy, ak je potencionálne nekonečná množina symbolov
 - Golombov → vysvetlený na ďalšej strane

Golombov (resp. Golomb-Rice) kód:

- $N = qm + r$, pričom $0 \leq r < m$
 - q je kvocient a r je zvyšok
- kód predstavuje q zakódované unárne (počtom jednotkových bitov) + „0“ + r zakódované ako skrátенý binárny kód, nazývaný aj Rice kód.
 - Skrátенý binárny kód sa používa pre rovnomerné rozdelenia pravdepodobnosti pre veľkosť abecedy m , pričom m nemusí byť mocninou o základe 2
 - ak $m = 2^r$, potom je výsledok identický s binárnym kódom
 - inak $m = 2^r + b$
 - prvým $2^r - b$ symbolom priradí kódové slová o dĺžke r
 - ostatným $2b$ symbolom priradí posledných $2b$ kódových slov o dĺžke $k + 1$
 - na obrázku sú príklady pre $m=5$
 - ako vyzerá strom – ukazuje, že výsledný kód je **prefixový**
 - ako vyzerá binárna reprezentácia - šedé hodnoty sú vynechané bity a kombinácie), na reprezentáciu 5 možných zvyškov sa použijú biele bity

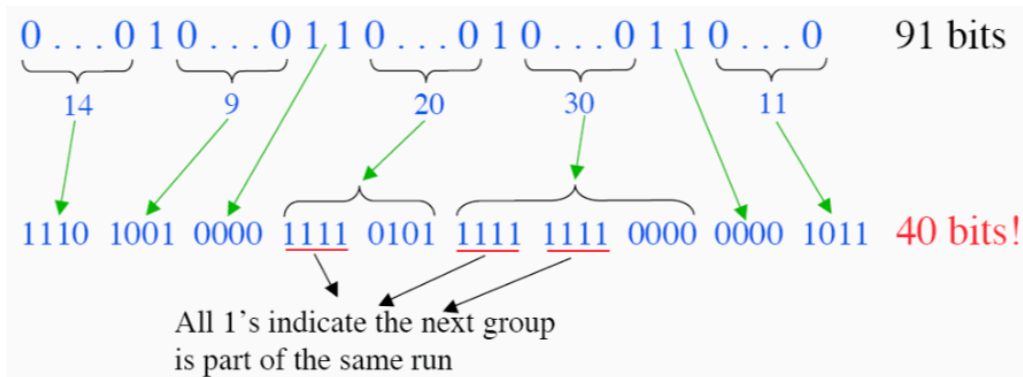


Príklad: Golombov kód pre $m=5$ a náhodne zvolené čísla N

N	q	r	Rice(r)	Golombov kód
2	0	2	10	010
6	1	1	01	1001
9	1	4	111	10111
10	2	0	00	11000
27	5	2	10	11111010

RLE Verzia 2

- predpokladáme, že jednotky sa vyskytujú zriedkavo a osamotene
- kóduje iba behy núl, behy jednotiek nekóduje
- ak sa náhodou vyskytuje viac jednotiek za sebou, oddeľujú sa oznámením, že medzi nimi je 0 núl
- Príklad (fixná dĺžka behov, kóduje sa pomocou r=4 bitov):



Výsledok: 40 bitov

- Pre túto verziu sa dá jednoducho vypočítať aj optimálne m, ak má byť použitý Golombov kód (viď ďalšia strana)

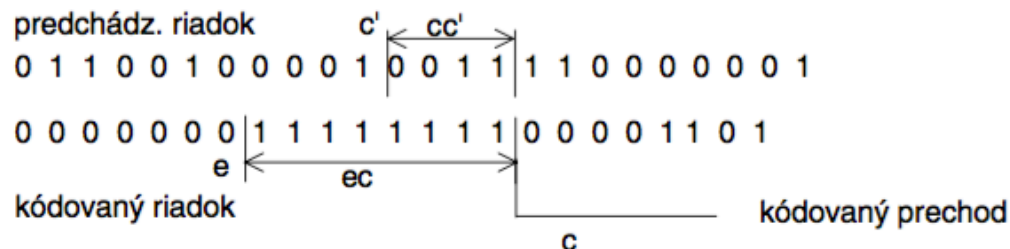
Aké m v golombovom kóde je optimálne pre RLE verzia RLE 2

- Nech pravdepodobnosť symbolu 0 je p a pravdepodobnosť symbolu 1 je $1 - p$
- Potom optimálne $m = \text{ceil}(-1/\log_2 p)$
- Napr. ak $p = \frac{127}{128}$, potom $m = \text{ceil}\left(-\frac{1}{\log_2\left(\frac{127}{128}\right)}\right) = \text{ceil}(88.38) = 89$

2D verzia RLE – kódovanie bitových rovín

- Používa sa najmä RAC (relative address coding) kódovanie
- Kódujeme napr. po riadkoch
- Môžeme si vybrať čo kódujeme:
 - Vzdialenosť od posledného prechodu v tom istom riadku
 - Vzdialenosť od rovnakého prechodu v predchádzajúcom riadku smerom vľavo
 - vzdialenosť od rovnakého prechodu v predchádzajúcom riadku smerom vpravo
- Potom potrebujeme
 - Vybrať tú vzdialenosť, ktoré je najkratšia (kódovateľná najmenším počtom bitov)
 - vybrať prefix tejto vzdialenosti, aby bolo jasné o ktorú s uvedených vzdialeností kódujeme

Príklad [Gonzales, Woods, str. 453]:

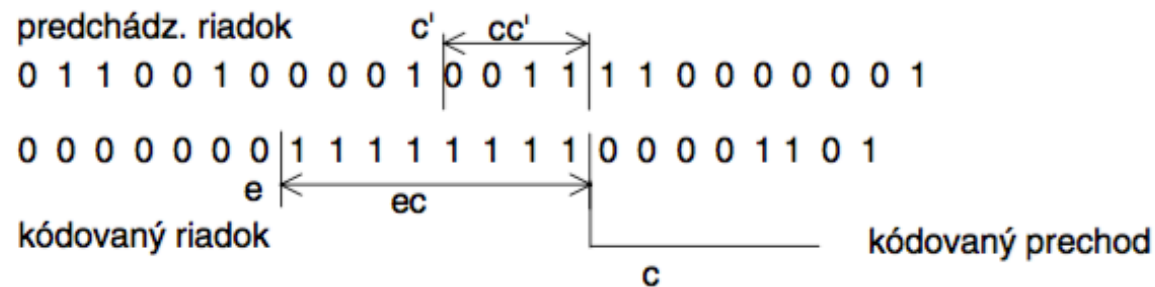


možná vzdialenosť	vzdialenosť	kód
cc'	0	0
ec alebo cc' (vľavo)	1	100
cc' (vpravo)	1	101
ec	$d (d > 1)$	$111h(d)$
cc' (c' vľavo)	$d (d > 1)$	$1100h(d)$
cc' (c' vpravo)	$d (d > 1)$	$1101h(d)$

Výsledký kód predstavuje

- prefix možností pre rôzne kombinácie vľavo/vpravo, aktuálny/minulý riadok, $d=0$, $d=1$, $d>1$
- kódovanie vzdialenosti d pomocou vhodného kódu s variabilnou dĺžkou – $h(d)$

Príklad – postup:

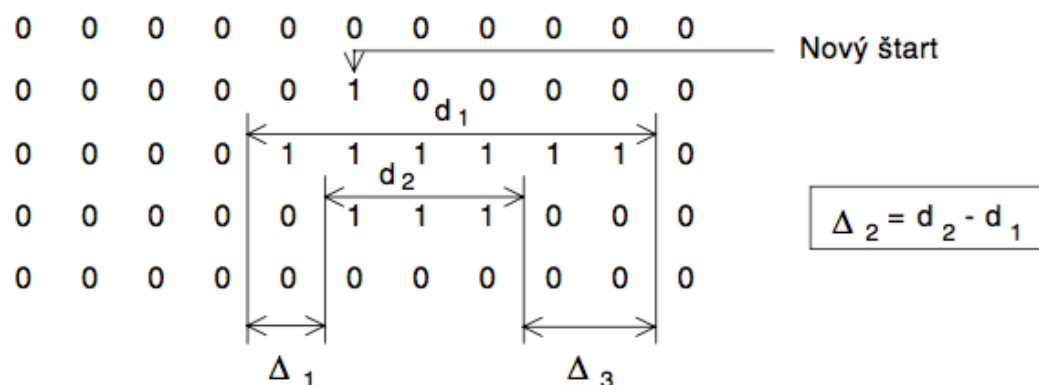


možná vzdialenosť	vzdialenosť	kód
cc'	0	0
ec alebo cc' (vľavo)	1	100
cc' (vpravo)	1	101
ec	d (d>1)	111h(d)
cc' (c' vľavo)	d (d>1)	1100h(d)
cc' (c' vpravo)	d (d>1)	1101h(d)

Výsledný kód:

- horný riadok: ec d=1, ec d=3, ec d=5, ec d=1, ec d=2, ec d=4, ec d=6, ec d=1
- dolný riadok:
 - rozhodovanie medzi ec d=7 a ccl d=6, ak vyhra ec, mame refernciu do ďalšieho cc ...
 - ec d=8 (rozhodovanie medzi tým a ccl d=4)

Kódovanie bitových rovín - kódovanie kontúr objektov: PDQ, DDC



Po štarte kódujeme každý riadok od „štartu“ objektu (jeho „najsevernejší bod“).

1) Najprv kódujeme prvý riadok: kóduje sa pozícia štartu (x,y) a dĺžka prvého riadku d_1

2) Potom kódujeme riadky pod

a. Metóda PDQ (predikčná diferenčná kvantizácia) kóduje pre každý riadok Δ_1, Δ_2 , vo vzorcoch d_2 predstavuje dĺžku aktuálneho a d_1 dĺžku predošlého riadku

b. Metóda DDQ (dvojnásobné delta kódovanie) kóduje pre každý riadok Δ_1, Δ_3

3) Na kódovanie uvedených hodnôt sa použije niektorý z kódov s variabilnou dĺžkou slova kódujúci aj znamienko.

Príklad na obrázku:

PDQ	DDQ
1) Kóduj (x,y)=(6,2), d=1	4) Kóduj (x,y)=(6,2), d=1
2) Kóduj $\Delta_1 = -1, \Delta_2 = 5$	5) Kóduj $\Delta_1 = -1, \Delta_3 = 4$
3) Kóduj $\Delta_1 = 1, \Delta_2 = -3$	6) Kóduj $\Delta_1 = 1, \Delta_3 = -2$

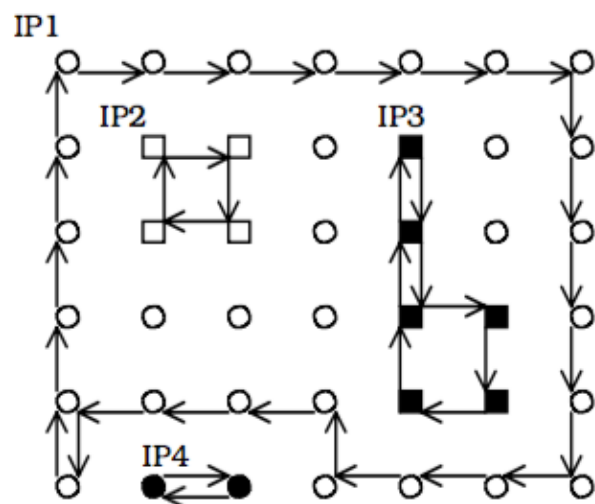
Viacúrovňové kódovanie

Podobný spôsob ako je PDQ a DDC kódovanie, je možný realizovať pre kódovanie obrazu, ktorý obsahuje viac kvantizačných úrovní [29]. Princiipiálne ide o opis uzavretých geometrických útvarov s rovnakou kvantizačnou úrovňou.

Pre realizáciu takéhoto kódovania je potrebné kódovať:

1. úroveň jas (prípadne poradie farby),
2. štartovací bod príslušného objektu,
3. tvar objektu - uzavretú cestu od štartovacieho bodu opisujúcu obrys objektu a vracajúcu sa späť k štartu.

Postup je už len logickým riešením daného problému. Je potrebné obrysy označovať tak, aby v každom uzavretom reťazci existovala buď len príslušná kvantizačná úroveň alebo ďalší uzavretý objekt, s tou istou podmienkou. Ukážeme si to na príklade so štyrmi úrovňami:



Na záver potrebujeme zakódovať podľa tabuľky znaky napr. v poradí :
poradie obrysu (IPx), úroveň, riadková súradnica IPx, stĺpcová súradnica IPx, smer prvého posunu,
smer druhého posunu, ...

Kód je samozrejme najúčinnnejší pri malom počte kvantizačných úrovní a veľkých jed-
nourovňových objektoch.

Viacúrovňové kódovanie – tabuľky

poradie obrysu	kódové slovo	riadky a stĺpce	kódové slovo
1	00	1	000
2	01	2	001
3	10	:	
4	11	8	111
úroveň	kódové slovo	smer popisu	kódové slovo
○	00	↑	00
□	01	→	01
■	10	↓	10
●	11	←	11

Kódovanie tvarov v MPEG4 – reťazový kód

Pri tejto metóde sú hranice objektu reprezentované uzavretou linkou. Kódovanie spočíva v zaznamenávaní smeru pohybu tejto linky. Kódovanie je ukončené, keď je dosiahnutý východzí bod.

Zakódované dáta určujú štartovací bod s prvým smerovým kódom a nasledujúce diferencne zakódované smery. Ak VOP obsahuje viacero uzavretých liniek, tak reťazové kódy nasledujú za informáciou o počte oblastí.

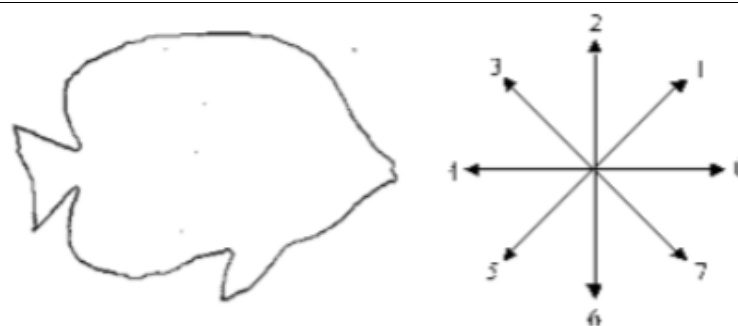
Keďže reťazový kód je cyklický, smer môžeme diferencne kódovať do hodnôt od -3 do 4:

$$\begin{aligned} &cn - cn-1 + 8, && \text{ak } cn - cn-1 > -3 \\ d = &cn - cn-1 - 8, && \text{ak } cn - cn-1 < -4 \\ &cn - cn-1, && \text{inak} \end{aligned} \quad (4.1)$$

kde d je diferencný reťazový kód, cn súčasný a $cn-1$ predchádzajúci smer. Na kódovanie d sa používa Huffmanov kód.

Huffmanov kód pre diferencný reťazový kód

d	kód
0	1
1	00
-1	011
2	0100
-2	01011
3	010100
-3	0101011
4	0101010



Obr. 6.7 Kódovaný objekt a znázornenie ôsmich kódovacích smerov [64]

Na strane prijímača je cn dekódovaná podľa vzorca

$$cn = (cn-1 + d + 8) \bmod 8$$

