

# ENTROPICKÉ KÓDOVANIE V ŠTANDARDNÝCH VIDEOKODEKCH (ENTROPY CODING IN STANDARD VIDEOCODECS)

Ján MIHALÍK, Radoslav ŠTEFANIŠIN

Laboratórium číslicového spracovania obrazov a videokomunikácií, Katedra elektroniky a multimediálnych telekomunikácií, Fakulta elektrotechniky a informatiky, Technická univerzita v Košiciach, Park Komenského 13, 041 20 Košice, Slovenská republika, tel.: 055/602 2854, 055/602 4309, E-mail: Jan.Mihalik@tuke.sk, Radoslav.Stefanisin@tuke.sk

## SUMMARY

*The paper deals with algorithms of entropy coding in the standard video codecs H.261,263,264 and MPEG-1,2,4 considerably increasing their total data compression. These algorithms are variable length coding by using Huffman codes, arithmetic coding, universal variable length coding, context based adaptive binary arithmetic coding employing context modeling and binarization of nonbinary symbols. In generally they employ the statistical nonstationary properties of video signals, which depend on the content of visual scenes.*

*The tables of Huffman code used in all standard video codecs are fixed and optimized for the bounded range of bit rates. To increase the range in the video codec H.26L there is used a new algorithm of entropy coding based on universal variable length code. Disadvantage of both Huffman and universal variable length codes is that each symbol is encoded by integer number of bits even if with different length. Therefore the video codec H.263 allows using arithmetic coding that brings saving of bit rates about 3-4% in interframe and up to 10% in intraframe mode of the video codec. The biggest increasing of the performance of entropy coding can be achieved by context based adaptive binary arithmetic coding that removes of an intersymbol redundancy. By its applying in the video codec H.264 the saving of bit rates may be near 32% compared to the universal variable length code at the same quality of coded video signals.*

*All algorithms of entropy coding are applied on entropy coding not just of the main information about transform coefficients in intraframe and interframe modes of the standard video codecs but even of additional information.*

**Keywords:** standard video codecs, entropy coding, Huffman codes, arithmetic coding, universal variable length coding, context modeling, binarization

## 1. ÚVOD

Medzinárodná štandardizácia videokodekov na poli ITU-T a ISO/IEC umožnila nové aplikácie vo videokomunikáciách [1] ako sú: videotelefónia, videokonferencie, video na požiadanie, číslicové TV a HDTV vysielanie, multimediálne služby (video) atď. Zatiaľ čo štandardný videokodek H.261 [3] bol vyvinutý pre videotelefónne a videokonferenčné aplikácie v prostredí ISDN sietí, tak štandardný videokodek H.263 [5] je jeho vylepšením a je určený pre videokomunikácie najmä v prostredí konvenčných telefónnych a mobilných sietí, ale aj po Internete. Naopak štandardný videokodek MPEG-1 [7] pokrýva širokú oblasť multimediálnych aplikácií od interaktívnych systémov na CD-ROM až po videokomunikácie v telekomunikačných sieťach. Štandardný videokodek MPEG-2 [8] umožňuje rozvoj digitálnej TV a HDTV, ale aj multimediálnych systémov v prostredí ATM a širokopásmových ISDN sietí. V porovnaní s videokodekami MPEG-1 a MPEG-2 štandardný videokodek MPEG-4 [9] vo všeobecnosti kóduje jednotlivé videoobjekty namiesto celých vizuálnych scén, čím podporuje rozvoj aplikácií s obsahovou interakciou. Prítom efektívna reprezentácia videoobjektov, manipulácia s nimi, editovanie v ich bitových tokoch ako aj možnosť škálovania dávajú úplne nové druhy tejto interakcie. Najnovší štandardný videokodek H.264 [18] vychádza z dlhodobovo vyvíjaného videokodeka H.26L [12], ktorý výrazne zvyšuje kompresiu v porovnaní

s predchádzajúcimi štandardami. Týmto rozširuje rozsah možných aplikácií, resp. multimediálnych služieb, a to aj so zvýšenou kvalitou kódovaného videosignálu v prostredí telekomunikačných sietí s obmedzenou kapacitou.

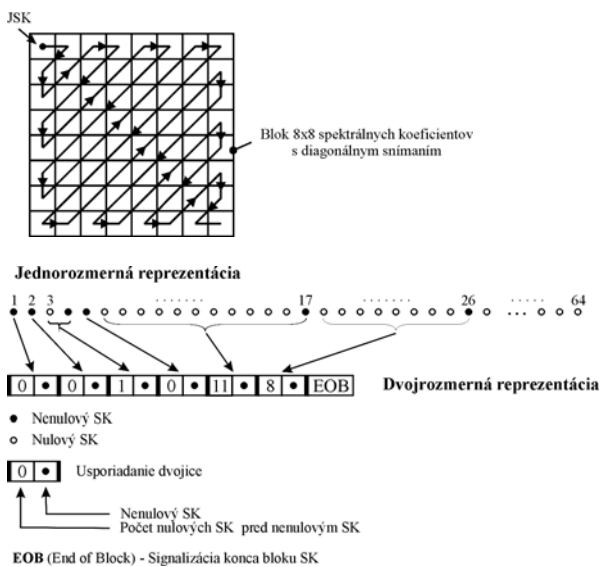
Všetky štandardné videokodeky dosahujú kompresiu údajov na základe redukcie vnútroštruktúry a medzisúštruktúry korelácie videosignálov. Ich jadro je tvorené medzisúštruktúry hybridným kódovacím systémom (MHKS) [19] s kompenzáciou pohybu, ktorý využíva blokové metódy na estimáciu a kompenzáciu pohybu pri medzisúštruktúry predikcii a dvojrozmernú diskretnú kosínusovú transformáciu na kódovanie predikčnej chyby (chybového videosignálu). Ďalšie zvýšenie kompresie sa v nich dosahuje pomocou entropického kódovania ako hlavnej informácie o spektrálnych koeficientoch v medzisúštruktúry alebo vnútroštruktúry móde, tak aj prídavných informácií, o vektoroch pohybu, type makrobloku, parametra výberu referenčnej snímky, atď.

## 2. ENTROPICKÉ KÓDOVANIE V H.261, 263

V procese entropického kódovania v štandardných videokodekoch sa používa mnohorozmerná reprezentácia postupnosti kvantovaných spektrálnych koeficientov (SK). V štandardnom videokodeku H.261 [3] je použitá dvojrozmerná reprezentácia (počet núl, hodnota), pričom koniec bloku sa signalizuje pomocou presne

definovaného kódového slova EOB (end of block). Pre ilustráciu je na obr. 1 znázornená dvojrozmerná reprezentácia postupnosti kvantovaných SK získanej ich diagonálnym snímaním v bloku 8x8.

Pri generovaní postupnosti kvantovaných SK [4] vo vnútroštruktúrnym móde H.261, na základe ktorej sú vytvárané tieto dvojice, sa neuvažuje jednosmerný SK (JSK). Tento sa vo vnútroštruktúrnym móde kóduje samostatne a na jeho kódovanie sa používa binárny rovnomerný kód s počtom bitov 8. Naopak jednotlivé dvojice sú kódované pomocou tabuľky Huffmanovho kódu [3], v ktorej sa nachádzajú len najčastejšie sa vyskytujúce dvojice a jej časť je uvedená v tab. 1.



**Obr. 1** Dvojrozmerná reprezentácia postupnosti kvantovaných SK získanej ich diagonálnym snímaním v bloku 8x8

**Fig. 1** Two-dimensional representation of the sequence of SC obtained by „zig-zag“ scanning of the 8x8 block

Počet núl	Hodnota	Kód
0	1	1s <sup>a)</sup> Pre prvý koeficient v bloku
0	1	11s Pre všetky koeficienty okrem prvého
0	2	0100 s
0	3	0010 1s
0	4	0000 110s
0	5	0010 0110 s
0	6	0010 0001 s
0	7	0000 0010 10s
0	8	0000 0001 1101 s
0	9	0000 0001 1000 s
0	10	0000 0001 0011 s
0	11	0000 0001 0000 s
0	12	0000 0000 1101 0s
0	13	0000 0000 1100 1s
0	14	0000 0000 1100 0s
0	15	0000 0000 1011 1s
1	1	011s
⋮	⋮	⋮

a) Nepoužíva sa vo vnútroštruktúrnym režime

**Tab. 1** Tabuľka Huffmanovho kódu použitá v H.261

**Tab. 1** Table of Huffman code used in H.261

Táto tabuľka platí pre všetky kvantované SK medziskrávkového módu a striedavé SK vnútroštruktúrného módu štandardného videokodeka H.261. Posledný bit kódových slov v tejto tabuľke označený „s“ predstavuje znamienko hodnoty, pričom pre kladné hodnoty je 0 a pre záporné 1.

Pre kódovanie dvojíc, ktoré sa nenachádzajú v tabuľke Huffmanovho kódu sa používa pevne 20 bitov, ktoré sú tvorené escape kódom (000001), rovnomerným kódom (6 bit) na binárne vyjadrenie počtu núl a dvojkovým doplnkom (8 bit), ktorý definuje hodnotu nenulového SK. Jednotlivé kódové slová neurčujú pozíciu daného kvantovaného SK v bloku a preto pre ich správne dekódovanie je potrebné poznať usporiadanie SK. Najčastejšie toto usporiadanie je dané diagonálnym snímaním SK v bloku, ktoré sa používa pri vytváraní dvojíc. Vo vnútroštruktúrnym móde H.261 usporiadanie začína JSK, pokračuje dvojicami predstavujúcimi počty núl pred striedavými nenulovými kvantovanými SK a končí EOB. V medziskrávkovom móde H.261 usporiadanie je tvorené obdobačnými dvojicami, ale pre všetky SK včítane JSK bloku a končí tiež EOB.

V štandardnom videokodeku H.263 [5], [10], podobne ako v H.261 [3], je použitá dvojrozmerná reprezentácia postupnosti kvantovaných SK. Na rozdiel od H.261 tento videokodek nepotrebuje špeciálne kódové slovo (EOB) na určenie konca bloku. Dvojica (počet núl, hodnota), používaná v H.261, je doplnená o jednu zložku na indikáciu posledného nenulového kvantovaného SK v bloku. Táto má hodnotu 1, ak SK je posledný v bloku, inak má hodnotu 0, keď za ním nasleduje ďalší nenulový kvantovaný SK. Po tomto doplnení dostaneme v H.263 trojice (indikácia, počet núl, hodnota). Pre kódovanie najčastejšie sa vyskytujúcich trojíc sa používa tabuľka Huffmanovho kódu, ktorej časť je uvedená v tab.2. Posledný bit kódových slov v tejto tabuľke, označený „s“, má rovnaký význam ako v tab.1. Prvý stĺpec tejto tabuľky obsahuje indexy jednotlivých riadkov tabuľky, ktoré zohrávajú dôležitú úlohu pri aritmetickom kódovaní v H.263.

Index	Indikácia	Počet núl	Hodnota	Kód
0	0	0	1	10s
1	0	0	2	1111 s
2	0	0	3	0101 01s
3	0	0	4	0010 111s
4	0	0	5	0001 1111 s
5	0	0	6	0001 0010 1s
6	0	0	7	0001 0010 0s
7	0	0	8	0000 1000 01s
8	0	0	9	0000 1000 00s
9	0	0	10	0000 0000 111s
10	0	0	11	0000 0000 110s
11	0	0	12	0000 0100 000s
...	...	...	...	...

**Tab. 2** Tabuľka Huffmanovho kódu použitá v H.263

**Tab. 2** Table of Huffman code used in H.263

Ostatné trojice sú kódované s nemennou dĺžkou kódového slova 22 bitov, ktoré pozostáva z escape kódu (0000011), indikácie posledného nenulového

kvantovaného SK v bloku (0 alebo 1), binárneho kódu (6 bit) na vyjadrenie počtu núl a dvojkového doplnku (8 bit) na vyjadrenie hodnoty nenulového kvantovaného SK. JSK vo vnútroštruktúrnym móde je kódovaný samostatne pomocou binárneho kódu s 8 bitmi ako v H.261. Kompletne tabuľky Huffmanovho kódu a rovnomerného kódu, zodpovedajúce jednotlivým dvojiciam alebo trojiciam, sú uvedené v odporúčaníach pre tieto štandardy [3], [5].

Videokodek H.263 umožňuje použiť okrem Huffmanovho kódu aj aritmetické kódovanie kvantovaných SK [2]. Experimenty ukazujú, že použitie aritmetického kódovania prináša úsporu bitovej rýchlosti 3-4% pre medzisímkový, a až 10% pre vnútroštruktúrný mód tohto videokodeka. Na základe tabuliek používaných pri kódovaní trojíc, boli vytvorené pravdepodobnostné modely zodpovedajúce JSK vnútroštruktúrného módu, indikácií posledného nenulového kvantovaného SK, počtu núl a hodnôt. Pri ich tvorbe sa používa indexovanie riadkov v týchto tabuľkách, pričom určitému indexu zodpovedá príslušná trojica a jej kódové slovo. V modeli však tomuto indexu prislúcha početnosť tejto trojice, ktorá vyplýva z pravdepodobnosti výskytu trojíc na základe ktorej boli pre ne navrhnuté aj kódovacie tabuľky. Indexovanie zohráva dôležitú úlohu pri samotnom kódovaní, pretože je to parameter na rozlíšenie vstupných údajov. Okrem toho vymedzuje pri aritmetickom kódovaní výber podintervalu zodpovedajúci vstupnému symbolu. Vychádzajúc z toho, že každá zložka trojrozmernej reprezentácie má svoj vlastný model, budú sa jednotlivé zložky kódovať samostatne. Použitie aritmetické kódovanie [6] je založené na prírastkovom prenose s rozširovaním aktuálneho podintervalu a využitím celočíselnej aritmetiky. Preto pre správne dekódovanie je potrebné na začiatku aritmetické kódovanie inicializovať. Táto inicializácia pozostáva z načítania určitého minimálneho počtu bitov, na základe ktorých sa určí začiatkový interval potrebný na správne dekódovanie. Pre H.263 je tento počet 17 bitov.

### 3. ENTROPICKÉ KÓDOVANIE V MPEG-1,2,4

Rovnaká reprezentácia kvantovaných SK dvojicami ako v H.261 je použitá v MPEG-1,2 [7], [8] a uvažuje sa pre všetky SK medzisímkového módu a striedavé SK vnútroštruktúrného módu týchto štandardných videokodekov. Opäť pre dvojice s najväčšou pravdepodobnosťou výskytu existuje tabuľka Huffmanovho kódu podobná tej v H.261. Ostatné dvojice sa kódujú obdobne ako v H.261 s tým rozdielom, že hodnota nenulového kvantovaného SK môže byť v MPEG-1 vyjadrená 8 bitmi ( $-127 \leq \text{hodnota} \leq 127$ ), alebo 16 bitmi, ak táto hodnota je z intervalu  $\langle -128, -255 \rangle$  alebo  $\langle 128, 255 \rangle$ . Na vyjadrenie hodnoty z intervalov  $\langle -128, -255 \rangle$  a  $\langle 128, 255 \rangle$  je potrebných 9 bitov, z toho 1 bit je znamienkový. Kódové slová pre

hodnoty z týchto intervalov sú však v MPEG-1 tvorené znamienkovým bitom, za ním nasleduje 7 nulových bitov a až potom dvojkový doplnok samotnej hodnoty s 8 bitmi. Hodnota nenulového kvantovaného SK je v MPEG-2 vyjadrená pomocou 12 bitov ( $-2047 \leq \text{hodnota} \leq 2047$ ). Štandardy MPEG-1,2 majú rovnaké kódové slovo EOB a escape kód ako H.261.

Jednosmerné SK susedných blokov vo vnútroštruktúrnym móde videokodekov si zachovávajú silnú koreláciu, preto je výhodnejšie kódovať diferenciu medzi dvoma JSK než priamo ich hodnoty. Tento princíp kódovania JSK vo vnútroštruktúrnym móde je použitý v MPEG-1 a 2. Diferencia JSK sa pred kódovaním reprezentuje dvoma zložkami a to indexom absolútneho rozsahu a hodnotou. Vzájomný vzťah medzi rozsahom absolútnych hodnôt diferencií JSK a jeho indexom, ktorému zodpovedá nerovnomerný kód, zobrazuje tab. 3. Táto tabuľka sa používa pre MPEG-1 [7], pričom odlišné kódové slová pre jasové a farebné zložky vychádzajú z ich rôznych štatistík.

Rozsah absolútnych hodnôt diferencií JSK	Index absolútneho rozsahu	Kód indexu absolútneho rozsahu pre jasové zložky	Kód indexu absolútneho rozsahu pre farebné zložky
0	0	100	00
1	1	00	01
2÷3	2	01	10
4÷7	3	101	110
8÷15	4	110	1110
16÷31	5	1110	11110
32÷63	6	11110	111110
64÷127	7	111110	1111110
128÷255	8	1111110	11111110

Tab. 3 Diferenčné kódovanie JSK v MPEG-1  
Tab. 3 Differential coding of DSC in MPEG-1

Rozsah diferencií JSK	Index absolútneho rozsahu	Dodatočné kódové slová
-255 ÷ -128	8	00000000 ÷ 01111111
-127 ÷ -64	7	0000000 ÷ 0111111
-63 ÷ -32	6	000000 ÷ 011111
-31 ÷ -16	5	00000 ÷ 01111
-15 ÷ -8	4	0000 ÷ 0111
-7 ÷ -4	3	000 ÷ 011
-3 ÷ -2	2	00 ÷ 01
-1	1	0
0	0	
1	1	1
2 ÷ 3	2	10 ÷ 11
4 ÷ 7	3	100 ÷ 111
8 ÷ 15	4	1000 ÷ 1111
16 ÷ 31	5	10000 ÷ 11111
32 ÷ 63	6	100000 ÷ 111111
64 ÷ 127	7	1000000 ÷ 1111111
128 ÷ 255	8	10000000 ÷ 11111111

Tab. 4 Dodatočné kódové slová pre kódovanie JSK v MPEG-1

Tab. 4 Additional codewords for coding of DSC in MPEG-1

Index absolútneho rozsahu určuje počet dodatočných bitov potrebných pre kódovanie hodnôt diferencií JSK.

V tab. 4 sú uvedené dodatočné kódové slová pre jasové a farebné zložky používané v MPEG-1, ktoré vyjadrujú hodnoty diferencií ich JSK. Potom výsledné kódové slovo pre JSK v MPEG-1 a 2 je tvorené kódovým slovom indexu absolútneho rozsahu a dodatočným kódovým slovom, ktoré definuje hodnotu diferenciie JSK. Napr. diferencií 10 JSK jasovej zložky zodpovedá kódové slovo 1101010 (110-kódové slovo indexu absolútneho rozsahu, 1010-dodatočné kódové slovo hodnoty). MPEG-2 [8] používa podobné kódovacie tabuľky pre kódovanie JSK vo vnútrostránkovom móde, pričom index možného absolútneho rozsahu diferencií JSK môže byť až 11.

Vo videokodeku MPEG-4 [9], [11] je použitá rovnaká reprezentácia kvantovaných SK pomocou trojíc ako v H.263 a pre najčastejšie vyskytujúce sa trojice v MPEG-4 existujú jeho vlastné tabuľky Huffmanovho kódu, ktoré sú rôzne pre medzisímkový a vnútrostránkový mód, pričom navyše sa zohľadňuje či ide o jasové alebo farebné zložky. Ostatným trojiciam v MPEG-4 zodpovedajú 22 bitové kódové slová, ktoré sa vytvárajú analogicky ako v H.263 so zohľadnením vlastných tabuliek rovnomerných kódov. JSK vo vnútrostránkovom móde sa kóduje diferenčne, podobne ako v MPEG-1 a 2, pričom index absolútneho rozsahu v MPEG-4 môže byť až 12. V prípade, že hodnota JSK je nula, tak sa tento diferenčne nekóduje, ale jednoducho sa zvýši počet núl pred nasledovným nenulovým kvantovaným SK a ten sa kóduje ako SK s vyššími frekvenciami, t.j. pomocou trojíc.

#### 4. ENTROPICKÉ KÓDOVANIE POĽA VEKTOROV POHYBU V H.261, 263 A MPEG-1, 2, 4

Celková bitová rýchlosť videokodeka obsahuje aj prídavné bity, použité na kódovanie poľa vektorov pohybu (VP). Z tohto dôvodu je efektívne kódovanie vektorov pohybu dôležitou súčasťou štandardných videokodekov [1]. Podobne ako jednosmerné SK vo vnútrostránkovom móde aj vektory pohybu susedných blokov si zachovávajú koreláciu. Na základe čoho sa vo všetkých štandardných videokodekoch používa diferenčné kódovanie poľa vektorov pohybu [10,11] samostatne po ich horizontálnych a vertikálnych zložkách. V H.261 sa najprv VP aktuálneho bloku predikuje pomocou VP predchádzajúceho bloku a potom diferenciie jeho horizontálnej aj vertikálnej zložky sú kódované použitím tabuľky Huffmanovho kódu. Kódové slovo vektora pohybu je tvorené kódovým slovom jeho horizontálnej zložky, za ktorým nasleduje kódové slovo jeho vertikálnej zložky. Časť tabuľky Huffmanovho kódu pre VP je uvedená v tab. 5, ktorá zohľadňuje, že v H.261 môžu zložky VP nadobúdať len celočíselné hodnoty z rozsahu  $\langle -15, 15 \rangle$ .

Treba uviesť dva dôležité poznatky vyplývajúce z tejto tabuľky. Prvý znamená, že na kódovanie malých diferencií sú použité krátke kódové slová, pretože tieto majú väčšiu pravdepodobnosť výskytu. Druhý poukazuje na to, že jedno kódové slovo je priradené až dvom možným hodnotám diferencií zložiek VP. Vo videesignále môže dôjsť k situácii, že susedné bloky majú úplne rozdielne vektory pohybu. V tomto prípade môžu byť diferenciie zložiek VP až  $\pm 30$ . Huffmanov kód zodpovedajúci veľkým diferenciám by nadobúdal veľký počet bitov, čo vyplýva z malej pravdepodobnosti týchto diferencií. Efektívnejšie riešenie kódovania diferencií je použitie jedného kódového slova pre kódovanie dvoch možných diferencií, ak je známe, že zložky výsledného vektora pohybu sú z určitého rozsahu ich hodnôt. Výsledný VP v štandardných videokodekoch sa získava súčtom VP získaného z prediktora a vysielanej diferenciie VP po jej dekódovaní. V H.261 sa musí nachádzať výsledný VP svojimi zložkami vždy v rozsahu hodnôt  $\pm 15$ . Tento poznatok na základe VP z prediktora určuje správne hodnoty zložiek prijatej diferenciie VP.

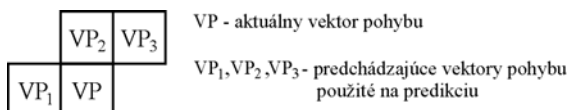
Diferencia zložiek VP		Kód
...	...	...
-7	25	0000 0111
-6	26	0000 1001
-5	27	0000 1011
-4	28	0000 111
-3	29	0001 1
-2	30	0011
-1		011
0		1
1		010
2	-30	0010
3	-29	0001 0
4	-28	0000 110
5	-27	0000 1010
6	-26	0000 1000
7	-25	0000 0110
...	...	...

Tab. 5 Diferenčné kódovanie VP  
Tab. 5 Differential coding of MV

Uvažujme o príklade, v ktorom VP z prediktora má zložku (vertikálnu alebo horizontálnu) s hodnotou 12 a prijaté kódové slovo dekódovanej zložky je 00000110. Tomuto kódovému slovu zodpovedajú hodnoty 7 alebo  $-25$ , ako to vyplýva z tab. 5. Výsledný VP bude mať túto zložku získanú súčtom buď  $12+7=19$ , ktorý je mimo prípustného rozsahu  $\langle -15, 15 \rangle$  alebo  $12-25=-13$ , ktorý sa už nachádza v tomto rozsahu, takže v tomto príklade kódové slovo 00000110 predstavuje hodnotu  $-25$ . Princíp kódovania poľa vektorov pohybu použitý vo videokodeku H.263 je rovnaký ako v H.261. Tiež sa používa diferenčné kódovanie VP na základe tabuľky Huffmanovho kódu. Rozdiel je v tom, že v H.263 zložky VP môžu nadobúdať aj neceločíselné hodnoty, pretože v ňom je prípustná estimácia pohybu s pol-prvkovou presnosťou [1]. V základnej modifikácii je rozsah možných hodnôt zložiek vektorov pohybu  $\langle -16, 15.5 \rangle$ . Využitie informácie, že výsledný vektor pohybu nadobúda len

hodnotu z tohto rozsahu, umožňuje opäť vytvoriť tabuľku Huffmanovho kódu, kde jedno kódové slovo zodpovedá dvom možným hodnotám diferencie zložiek VP. Iný rozdiel je v predikcii aktuálneho VP, kedy v H.263 pre jeho predikciu sa používa viacej VP ako to vidno z obr. 2 namiesto jedného predchádzajúceho VP<sub>1</sub> v H.261.

V H.263 je možné aj aritmetické kódovanie diferencii zložiek VP a to rovnakým spôsobom ako kvantovaných SK. Na základe tabuľky Huffmanovho kódu pre kódovanie diferencii zložiek VP je vytvorený pravdepodobnostný model použitý pri aritmetickom kódovaní [5].



**Obr. 2** Predikcia VP v štandarde H.263.  
**Fig. 2** Prediction of MV in the standard H.263

Videokodek MPEG-1 [7] umožňuje definovanie rôznych rozsahov zložiek VP ako aj presností (prvková alebo pol-prvková) estimácie pohybu. Na definovanie rozsahu zložiek VP je použitý parameter index rozsahu VP, pričom informácia o ňom ako aj o presnosti estimácie pohybu je prenášaná v hlavičke snímky [10]. Tab. 6 ukazuje vzájomný vzťah medzi indexom a možnými rozsahmi zložiek VP pre prvkovú a pol-prvkovú presnosť estimácie pohybu.

Index rozsahu	Rozsah zložiek VP	
	Prvková presnosť	Pol-prvková presnosť
1	-16÷15	-8÷7.5
2	-32÷31	-16÷15.5
3	-64÷63	-32÷31.5
4	-128÷127	-64÷63.5
5	-256÷255	-128÷127.5
6	-512÷511	-256÷255.5
7	-1024÷1023	-512÷511.5

**Tab. 6** Rozsah zložiek VP  
**Tab. 6** The component range of MV

Index rozsahu	Modul
1	32
2	64
3	128
4	256
5	512
6	1024
7	2048

**Tab. 7** Moduly pre VP  
**Tab. 7** Modules for MV

Ak diferencia niektorej zložky VP je mimo rozsahu, ktorý vyplýva z indexu rozsahu, táto diferencia je upravená pričítaním alebo odčítaním určitej hodnoty (modulu), tak aby sa nachádzala v žiadanom rozsahu. Hodnoty modulu sú závislé od

indexu rozsahu a pre prvkovú presnosť estimácie pohybu sú uvedené v tab. 7.

Pre kódovanie diferencii zložiek VP sa používa tabuľka Huffmanovho kódu [7], ktorá obsahuje kódové slová pre obmedzený rozsah hodnôt <-16,16>, ale tieto diferencie môžu nadobúdať aj iné hodnoty. Preto všetky diferencie zložiek VP sú upravené podľa nasledujúceho zlomku

$$\frac{[\text{diferencia zložky} \pm (\text{index rozsahu} - 1)]}{\text{index rozsahu}} \quad (1)$$

kde zo znamienok  $\pm$  kladné znamienko zodpovedá kladným hodnotám a záporné znamienko záporným hodnotám diferencii zložiek VP. Výsledkom tejto úpravy je hodnota nachádzajúca sa v tabuľke Huffmanovho kódu, pričom podiel čitateľa s menovateľom v zlomku (1) nemusí byť stále celé číslo. Potom celá časť tohto čísla sa použije pre nájdenie kódového slova v tejto tabuľke a absolútna hodnota zvyšku (včítane nulového) podielu sa vyjadří binárnym kódom s pevnou dĺžkou kódového slova. Výsledné kódové slovo je tvorené kódovým slovom z tabuľky Huffmanovho kódu a binárnym kódovým slovom, ktoré vyjadruje zvyšok.

Pre objasnenie predpokladajme prvkovú presnosť estimácie pohybu a rozsah zložiek VP <-32,31>, ktorému zodpovedá index 2 ako to vidno z tab. 6. Hodnota diferencie zložky VP nech je -44. Táto hodnota je mimo tohto rozsahu možných hodnôt <-32,31> a preto sa použije na jej korekciu modul s hodnotou 64 (tab. 7). Pričítaním modulu 64 k diferencii zložky VP -44 dostaneme jej upravenú hodnotu diferencie 20. Hodnota čitateľa v zlomku (1) je 21, ktorá je delená indexom rozsahu 2, výsledkom čoho je hodnota 10,5. Hodnote 10 zodpovedá v tabuľke Huffmanovho kódu kódové slovo 0000010010. K tomuto kódovému slovu pridáme kódové slovo 1 vyjadrujúce zvyšok podielu 0,5.index rozsahu=1. Potom výsledné kódové slovo bude **00000100101**. Podobný spôsob kódovania vektorov pohybu ako v MPEG-1 sa používa aj v MPEG-2 [8] a MPEG-4 [9].

## 5. ENTROPICKÉ KÓDOVANIE V H.26L A H.264

Tabuľky Huffmanovho kódu používané v štandardoch H.261,263 a MPEG-1,2,4 sú nemenné a optimalizované pre obmedzený rozsah bitových rýchlostí. Okrem toho nespĺňajú požiadavky rozmanitosti aplikácií, t.j. rozsah ich použitia je tiež obmedzený. Napr. tabuľky Huffmanovho kódu pre kódovanie SK sú v MPEG-2 navrhnuté pre sieťové aplikácie s veľkými bitovými rýchlosťami, ale nespĺňajú požiadavky pre nízko-bitové aplikácie, napr. videotelefóniu. Z toho vyplýva potreba existencie takých nerovnomerných kódov, ktoré by pokrývali široký rozsah aplikácií so zreteľom na rôznu kvalitu kódovania. V H.26L [12] je použitý nový algoritmus entropického kódovania založený na univerzálnom nerovnomernom kóde, ktorý

poskytuje mnohé výhody v porovnaní s tabuľkami Huffmanovho kódu.

### 5.1. Univerzálne nerovnomerné kódovanie

V H.26L je použité univerzálne nerovnomerné kódovanie (UNK) [12], [13] pre všetky druhy kódovaných symbolov (SK, zložky vektorov pohybu, atď.) a je popísané svojou tabuľkou univerzálneho nerovnomerného kódu. Kódové slová UNK sú vytvárané prekladaním (interleaving) prefixových a doplnkových kódových slov, ako je to znázornené v zjednodušenom tvare na obr. 3, kde  $x_n$  sú hodnoty 0 alebo 1.

$$\begin{array}{c} 1 \\ 0 x_0 1 \\ 0 x_1 0 x_0 1 \\ 0 x_2 0 x_1 0 x_0 1 \\ 0 x_3 0 x_2 0 x_1 0 x_0 1 \\ \dots \end{array}$$

**Obr. 3** Univerzálne nerovnomerné kódovanie v H.26L

**Fig. 3** Universal variable length coding in H.26L

Prefixové kódové slová (nezvýraznené časti), ako je vidno z obr. 3, sú tvorené 1, pred ktorú sa vkladajú 0 napr. 1,01,001,0001, atď. Ak  $L=2N-1$  je celková dĺžka kódového slova v bitoch, kde  $N$  je dĺžka prefixového kódového slova, potom  $N-1$  je počet bitov doplnkového kódového slova (DKS), t.j.  $(x_{N-2}, \dots, x_1, x_0)$ . Kódové slová sú číslované vzostupne počnúc 0, pričom ich čísla vypočítame z nasledovnej rovnice

$$\text{číslo kódového slova} = 2^{N-1} + \text{DKS}_{10} - 1 \quad (2)$$

kde  $\text{DKS}_{10}$  predstavuje dekadickú hodnotu DKS a pre  $N=1$   $\text{DKS}_{10}=0$ .

Číslo kódového slova	Kódové slovo
0	1
1	001
2	011
3	00001
4	00011
5	01001
6	01011
7	0000001
8	0000011
...	.....

**Tab. 8** Tabuľka UNK použitá v H.26L

**Tab. 8** Table of UVLC used in H.26L

Tab. 8 zobrazuje časť tabuľky UNK používanej v H.26L, kde silnejšie vyznačené bity kódových slov v tejto tabuľke predstavujú DKS. Pre overenie platnosti rov. (2) predpokladajme napr. číslo 5 kódového slova, ktorému zodpovedá dĺžka prefixového kódového slova  $N=3$  a  $\text{DKS}=10$ , čo predstavuje dekadické číslo 2. Ak sú známe hodnoty  $N$  a  $\text{DKS}$ , potom je jednoduché vytvoriť pravidelnú štruktúru tabuľky UNK. Podobne dekodér môže jednoducho čítať kódové slovo bit po bite až po

poslednú 1, ktorá signalizuje koniec kódového slova. Medzi symbolom, ktorý sa má kódovať a číslom kódového slova, ktorému zodpovedá určité kódové slovo existuje definované pravidlo prevodu z kódovaného symbolu na číslo kódového slova [12]. Postup kódovania určitého čísla kódového slova, t.j. jemu zodpovedajúceho symbolu bude [13]:

1. Výpočet dĺžky prefixového kódového slova

$$N = \lfloor \log_2(\text{číslo kódového slova} + 1) \rfloor + 1 \quad (3)$$

kde  $\lfloor \log_2(\text{číslo kódového slova} + 1) \rfloor$  predstavuje celú časť výrazu vo vnútri vonkajších zátvoriek.

2. Výpočet dekadickej hodnoty

$$\text{DKS}_{10} = \text{číslo kódového slova} + 1 - 2^{N-1} \quad (4)$$

3. Vytvorenie univerzálneho kódového slova preložením prefixového kódového slova dĺžky  $N$  a DKS (binárne vyjadrenie dekadickej hodnoty zistenej v bode 2).

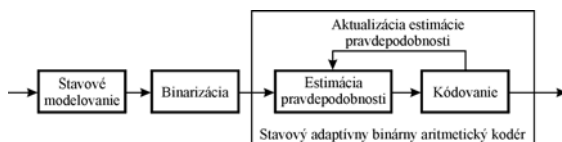
Pre lepšie objasnenie tohto procesu predpokladajme číslo 8 kódového slova. Z rov. (3) dostaneme  $N=4$ , čomu zodpovedá prefixové kódové slovo 0001. Z rov. (4) dostaneme  $\text{DKS}_{10}=1$ , čomu zodpovedá DKS 001 s dĺžkou  $N-1=3$  bity. Preložením týchto dvoch kódových slov podľa obr. 3 dostaneme výsledné kódové slovo **000011**. Prefixové kódové slovo výsledného kódového slova môže definovať určitú skupinu symbolov a doplnkové kódové slovo už konkrétny symbol danej skupiny. Pri dekódovaní musí dekodér načítavať výsledné kódové slovo bit po bite až po poslednú 1, ktorá signalizuje koniec tohto kódového slova. Potom na základe znalosti výsledného kódového slova sa určí hodnota  $N$  a  $\text{DKS}_{10}$  a z rov. (3) vyplynie číslo tohto kódového slova a tým aj dekódovaný symbol.

UNK zavedené v H.26L podstatne zjednodušuje kódovací a dekódovací proces. Okrem toho na strane kodéra a dekodéra nie je nevyhnutná tabuľka kódových slov. Napriek mnohým výhodám, ktoré prináša UNK, táto metóda entropického kódovania nie je optimálna z hľadiska účinnosti kódovania [13]. Pri návrhu jeho tabuľky kódových slov sa predpokladá len základné nemenné rozdelenie pravdepodobnosti kódovaných symbolov a nevyužíva sa korelácia medzi jednotlivými symbolmi. Okrem toho na kódovanie jedného symbolu je potrebný minimálne 1 bit. Predpokladajme univerzálne nerovnomerné kódovanie kvantovaných SK, t.j. dvojíc (počet núl, hodnota). Koncu bloku (EOB) zodpovedá v H.26L číslo 0, ktorému prislúcha kódové slovo s jedným bitom 1. Výsledky štatistickej analýzy v medzisnímkom, a hlavne vnútrošnímkom, móde [4] ukazujú, že z hľadiska kódovania je efektívnejšie priradenie tohto kódového slova inému symbolu a nie EOB. Nedostatky UNK, ktoré vyplývajú z uvažovania

doteraz nemenného rozdelenia pravdepodobnosti je možné zlepšiť v H.26L modifikáciou kódových slov, napr. adaptívnym prepínaním medzi dvoma tabuľkami UNK [14]. Ďalšou možnosťou je priama adaptácia priradovania medzi symbolmi určenými na kódovanie a univerzálnymi kódovými slovami [15].

## 5.2. Stavové adaptívne binárne aritmetické kódovanie

Oveľa väčšiu účinnosť entropického kódovania v dlhodobom vyvíjanom videokodeku H.26L [12] a následne v najnovšom štandarde H.264 [18] je možné dosiahnuť pomocou stavového adaptívneho binárneho aritmetického kódovania (SABAK) [16]. Základná bloková schéma entropického kódovania založená na SABAK je znázornená na obr. 4.



**Obr. 4** Bloková schéma entropického kódovania založená na stavovom adaptívnom binárnom aritmetickom kódovaní

**Fig. 4** Block diagram of entropy coding based on the context adaptive binary arithmetic coding

Prvým krokom tohto kódovania je vybratie vhodného modelu pre aktuálny kódovaný symbol na základe predchádzajúcich pozorovaní, ktorý bude štatistickým modelom zdroja pri jeho kódovaní. Proces vytvárania modelu sa často nazýva stavové modelovanie a závisí od druhu kódovaného symbolu. Ak aktuálny kódovaný symbol nie je binárnej hodnoty, potom je potrebné ho vyjadriť pomocou binárnej postupnosti. Tento proces je druhým krokom uvažovaného kódovania a nazýva sa binarizácia, ktorá je realizovaná na základe definovaného binárneho prúdu. Nakoniec každý bit tejto binárnej postupnosti je kódovaný stavovým adaptívnym binárnym aritmetickým kódérom, ktorý používa estimáciu jeho pravdepodobnosti, a tým implicitne štatistického modelu, ktorý vyplýva buď zo stavového modelovania alebo z procesu binarizácie. Po zakódovaní každého bitu bude prislúchajúci model aktualizovaný samotným kódovaným binárnym symbolom a tým sa stále prispôbuje k aktuálnym štatistikám.

Úlohou stavového modelovania je vybrať vhodný model pre aktuálny kódovaný symbol na základe predchádzajúcich už kódovaných symbolov. V H.26L alebo H.264 je navrhnuté stavové modelovanie [17] pre viacej druhov kódovanej informácie, konkrétne pre typ makrobloku (MB), vektory pohybu, parameter výberu referenčnej snímky, dvojice (počet núl, hodnota) SK atď.

Všetky nebinárne symboly sú zmenené na binárnu postupnosť (vyjadrenie). Binarizácia všetkých symbolov okrem symbolu definujúceho typ

makrobloku je realizovaná pomocou unitárneho kódu [16]. Pre binárne vyjadrenie typu MB, ktorého hodnoty sú z intervalu 0÷9, je použitá alternatívna binarizácia. Každý bit týchto kódových slov je potom kódovaný pomocou stavového adaptívneho binárneho aritmetického kódovania. Na základe rozlíšenia jednotlivých bitov v binárnej postupnosti pomocou ich pozície (číslo bitu) sú im pridelené rôzne modely (stavy).

Algoritmom entropického kódovania založenom na SABAK v porovnaní s UNK je možné dosiahnuť úsporu bitovej rýchlosti až do 32%. Pre videosekvencie QCIF sa dosahuje úspora bitovej rýchlosti v rozsahu 4,5÷15% a pre videosekvencie CIF je tento rozsah 5÷32%.

## 6. ZÁVER

Vo všeobecnosti entropické kódovanie v štandardných videokodekoch využíva štatistickú nestacionaritu videosignálov, ktorá závisí od obsahu vizuálnej scény.

Z algoritmov entropického kódovania v štandardných videokodekoch sa najčastejšie používa nerovnomerné kódovanie založené na nemenných tabuľkách Huffmanovho kódu. Pri entropickom kódovaní kvantovaných spektrálnych koeficientov sa tieto predtým efektívne reprezentujú pomocou dvojíc alebo trojíc a naopak vektory pohybu pomocou diferencií, ktoré sú výsledkom ich predikcie. Tabuľky Huffmanovho kódu používané v štandardoch H.261,263 a MPEG-1,2,4 sú nemenné a optimalizované pre obmedzený rozsah bitových rýchlostí. Okrem toho nespĺňajú požiadavky rozmanitosti aplikácií, t.j. rozsah ich použitia je tiež obmedzený. Z toho vyplýva potreba existencie takých nerovnomerných kódov, ktoré by pokrývali široký rozsah aplikácií so zreteľom na rôznu kvalitu kódovania. V H.26L je použitý nový algoritmus entropického kódovania založený na univerzálnom nerovnomernom kóde, ktorý poskytuje mnohé výhody v porovnaní s tabuľkami Huffmanovho kódu.

Nevýhodou Huffmanovho alebo univerzálného nerovnomerného kódu je, že každý symbol je zakódovaný s celistvým počtom bitov, aj keď tento je rôzny pre jednotlivé symboly. Tento nedostatok odstraňuje aritmetické kódovanie symbolov, ktoré dosahuje vyššiu účinnosť a táto sa viacej približuje k teoretickej hranici danej entropiou ich zdroja. Videokodek H.263 umožňuje použiť okrem Huffmanovho kódu aj aritmetické kódovanie kvantovaných SK a diferencií zložiek VP. Experimenty ukazujú, že použitie aritmetického kódovania prináša úsporu bitovej rýchlosti 3-4% pre medzismímkový a až 10% pre vnútrosmímkový mód tohto videokodeka.

Oveľa väčšiu účinnosť entropického kódovania v H.26L a následne v H.264 je možné dosiahnuť pomocou stavového adaptívneho binárneho aritmetického kódovania, ktoré umožňuje potlačiť aj medzismímkovú redundanciu. Stavovým modelovaním sa v ňom generujú vhodné modely stavov

v závislosti od meniacich sa štatistických charakteristík kódovaných symbolov. Potom estimované podmienené pravdepodobnosti týchto symbolov zodpovedajúce jednotlivým stavovým modelom vedú k adaptácii binárneho aritmetického kódovania. Neoddeliteľnou súčasťou tohto algoritmu entropického kódovania je binarizácia, pomocou ktorej všetky nebinárne symboly sú najprv zmenené na binárne postupnosti (vyjadrenia). V tomto prípade dosiahnutá úspora bitovej rýchlosti môže byť až 32 % v porovnaní s univerzálnym nerovnomerným kódovaním pri zachovaní kvality kódovaného videosignálu.

**Táto práca bola podporená s GAV MŠ a SAV SR v projekte č. 1/0384/03.**

## LITERATÚRA

- [1] Mihalík, J.: Kódovanie obrazu vo videokomunikáciách. Mercury-Smékal, Košice, 2001. ISBN 80-89061-47-8.
- [2] Witten, I.H. - Radford, M.N. - Cleary, J.G.: Arithmetic Coding for Data Compression. Communication of the ACM, Vol.30, No.6, 1987, pp.520-540.
- [3] ITU-T Recommendation H.261. Video Coding for Audiovisual Services at p×64 kbit/s. Geneva, 1990.
- [4] Mihalík, J. - Štefanišin, R.: Algoritmy kvantovania štandardných videokodekov. Acta Electrotechnica et Informatica, Vol.4, No.1, 2004, p.47-54, ISSN 1335-8243.
- [5] ITU-T Recommendation H.263. Video Coding for Narrow Telecommunication Channels at <64 kbit/s. Geneva, 1996.
- [6] Gladišová, I. - Mihalík, J.: Stavové aritmetické kódovanie binárnych obrazov. Acta Electrotechnica et Informatica, Vol.3, No.3, 2003, p.36-44, ISSN 1335-8243.
- [7] ISO/IEC 11172-2, MPEG-1. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s (Part Video). Geneva, 1993.
- [8] ISO/IEC 13818-2, MPEG-2. Generic Coding of Moving Pictures and Associated Audio Information (Part Video). Geneva, 1995.
- [9] ISO/IEC JTC1/SC29/WG11, MPEG98/N1902, MPEG-4 Video Verification Model Version 10.0. San Jose, 1998.
- [10] Mihalík, J.: Štandardný videokodek H.263 a jeho porovnanie s H.261. Slaboproudý obzor, roč. 60, č.1, 2003, s.1-7, ISSN0037-668X.
- [11] Mihalík, J.: Štandardný videokodek MPEG-4. Slaboproudý obzor, roč. 60, č.2, 2003, s.7-11, ISSN0037-668X.
- [12] Bjontegaard, G.: H.26L Test Model Long Term Number 4 (TML-4). ITU-T SG16/Q.6 Q15-J-72, August, 1999.
- [13] Zhou, M.: Modified Universal Variable Length Coding. ITU-T Document VCEG-N036, Santa Barbara, USA, 2001.
- [14] Bjontegaard, G.: Use of adaptive switching between two VLCs for intra luma coefficients. ITU-T SG16/Q.6 Q15-K-30, August, 2000.
- [15] Jeon, B.: Entropy Coding for H.26L. ITU-T SG16/Q.6 Q15-J-57, May, 2000.
- [16] Marpe, D. - Blättermann, G. - Heising, G.-Wiegand, T.: Video Compression Using Context-Based Adaptive Arithmetic Coding. ICIP'01, Thessaloniki, Greece, 2001.
- [17] Marpe, D. - Blättermann, G. - Wiegand, T.: Adaptive Codes for H.26L. ITU-T SG16/Q.6 VCEG-L-13, January, 2001.
- [18] Wiegand, T. - Sullivan, G.J. - Bjontegaard, G.-Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology, July, 2003, pp.1-19.
- [19] Mihalík, J.: Adaptívne hybridné kódovanie obrazu. Elektrotech. časopis, 44, č.3, 1993, s.85-89.

## BIOGRAPHY

**Ján Mihalík** graduated from Technical University in Bratislava in 1976. Since 1979 he joined Faculty of Electrical Engineering and Informatics of Technical University of Košice, where received his PhD degree in Radioelectronics in 1985. Currently, he is Full Professor of Electronics and Telecommunications and the head of the Laboratory of Digital Image Processing and Videocommunications at the Department of Electronics and Multimedia Telecommunications. His research interests include information theory, image and video coding, digital image and video processing and multimedia videocommunications.

**Radoslav Štefanišin** was born on 1978 in Svidník, Slovak Republic. He received the Ing. (M.Sc.) degree from Technical University of Košice in 2001. At present he is PhD student at the Department of Electronics and Multimedia Telecommunications of Technical University, Košice. His research interest includes video coding with very low bit rate.