STAVOVÉ ARITMETICKÉ KÓDOVANIE BINÁRNYCH OBRAZOV

(CONTEXT-BASED ARITHMETIC ENCODING OF BINARY IMAGES)

Iveta GLADIŠOVÁ, Ján MIHALÍK
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.: +421 55 602 2940, +421 55 602 2854, E-mail: iveta.gladisova@tuke.sk, jan.mihalik@tuke.sk

SUMMARY

This paper deals with an approach to the compression of binary images. Specifically, the approaach utilises a context-based arithmetic encoding method. The coding and modeling aspects are treated separately. Model statistics are studied from stationary and stationary adaptive assumptions This efficient binary context-based arithmetic code is relatively simple to implement because it avoids the multiplication operation. In the last part of the paper there are analyzes and investigation effeciency of the methods of context-based arithmetic encoding for given images

Keywords: ***binary*** *arithmetic code, context-based arithmetic encoding, conditional probabilities, the skew number Q, the carry-over problem, templare ,stationary model, adapting to stationary statistics, simulations*

1. ÚVOD

Základnou problematikou v obrazových komunikáciách [1] je prenos a záznam obrazov, ktorú možno riešiť efektívnym entropickým kódovaním. Ak vezmeme do úvahy, že prenosové a záznamové systémy sú navrhnuté takmer vždy pre prácu len s dvomi symbolmi „0“ a „1“, tak pomocou nich sa musí zakódovať celá vstupná postupnosť. Symboly „0“ a „1“ tvoria kódovú abecedu a symboly zo vstupnej postupnosti tvoria zdrojovú abecedu. Každý symbol zo zdrojovej abecedy musí byť kódovaný pomocou symbolov z kódovej abecedy. Ak má byť kódovanie entropické, tak potom symboly zo zdrojovej abecedy budú mať rôznu dĺžku kódu, a preto kód musí byť prefixový. To znamená, že nijaký kód symbolu zo zdrojovej abecedy nesmie byť predponou iného kódu symbolu. V opačnom prípade by bol kód nedekódovateľný. Medzi takéto kódy patria Shannonov kód a Huffmanov kód [2]. Ich nevýhodou je, že každý symbol zo zdrojovej abecedy musí byť kódovaný celočíselným počtom symbolov z kódovej abecedy, t.j. celočíselným počtom bitov. Ak pre kódovanie symbolu postačuje napr. 2.1 bitov, tak daný symbol musí byť kódovaný minimálne pomocou 3 bitov, čo vedie k predĺženiu výsledného kódu celej postupnosti.

Tento problém rieši aritmetický kód [4], [6], ktorý je založený na inom princípe, a tak dokáže zakódovať symbol aj pomocou neceločíselného počtu symbolov (bitov) z kódovej abecedy. Týmto sa aritmetický kód blíži k ideálnemu entropickému kódu. Jeho realizácia je však na rozdiel od Huffmanovho alebo Shannonovho kódu zložitejšia. Článok sa zaoberá rôznymi metódami stavového aritmetického kódovania binárnych obrazoch, ich implementáciou, ako aj dosiahnutými výsledkami.

2. ARITMETICKÉ KÓDOVANIE

Nedostatkom Huffmanovho kódu predchádza aritmetický kód (AK). Aritmetický kód obchádza myšlienku nahradenia symbolu špecifickým kódom a namiesto toho nahrádza tok vstupných symbolov jednoduchým desatinným číslom z intervalu <0,1). V aritmetickom kódovacom systéme musí kodér spolupracovať s modelom, ktorý poskytuje informáciu o pravdepodobnosti. Kodér potom využíva tieto pravdepodobnosti na zakódovanie aktuálneho symbolu. Na zabezpečenie dekódovateľnosti môže model použiť len informácie známe kodéru aj dekodéru. Model sa môže zmeniť ihneď po zakódovaní vstupnej postupnosti [4], [5] .

Základný algoritmus pre AK začne s aktuálnym polootvoreným intervalom pravdepodobnosti počiatočne nastaveným na <0,1). Pre každý symbol vstupnej postupnosti urobí nasledujúce dva kroky :

1) Aktuálny interval rozdelí na subintervaly pre každý možný symbol abecedy zdroja. Veľkosť subintervalu každého symbolu je úmerná pravdepodobnosti jeho výskytu vo vstupnej postupnosti.

2) Vyberie ten subinterval korešpondujúci so symbolom, ktorý sa aktuálne nachádza v postupnosti a vytvorí nový aktuálny interval.

Nakoniec pridelí dostatok bitov na rozlíšenie konečného aktuálneho intervalu od všetkých iných možných konečných intervalov. Dĺžka konečného intervalu je rovná súčinu pravdepodobností všetkých doterajších symbolov. Na indikovanie konca postupnosti sa používa špeciálny znak EOF (end of file) alebo sa prenáša dĺžka postupnosti.

Základná implementácia AK má dve nevýhody: zmenšovanie aktuálneho intervalu vyžaduje použitie veľmi presnej matematiky a bity výsledného kódového slova dostaneme až po prečítaní všetkých symbolov vstupnej postupnosti. Najlepšie riešenie týchto dvoch problémov je začiatočné výstupné bity vysielať hneď, ako sú známe, a potom zdvojnásobiť dĺžku aktuálneho intervalu, takže ostane iba neznáma časť konečného intervalu. Z hľadiska účinnosti sa v praxi potvrdili teoretické predpoklady, že AK kóduje symboly účinnejšie ako Huffmanov kód, pričom najväčší rozdiel je pri kódovaní symbolov, ktorých pravdepodobnosti sú veľmi rozdielne.

2.1. Základné pojmy a spôsob AK

Nech **s** je vstupná binárna postupnosť, ktorá sa má kódovať. Nech **s**y je binárny reťazec, ktorý vznikne pridaním binárneho symbolu y na koniec postupnosti **s**. Nech k je menej pravdepodobný symbol, taký že jeho pravdepodobnosť $p(k) \leq 0{,}5$ a nech m je komplementom k, teda viac pravdepodobný symbol.

Ďalej nech C(**s**) je binárna premenná postupnosti **s** vyjadrujúca dolnú hranicu subintervalu pravdepodobnosti, pričom jej začiatočná hodnota pred zakódovaním prvého symbolu (keď *s* je prázdna postupnosť) je rovná 0. Nech C(**s**m) je takáto binárna premenná postupnosti **s** pre viac pravdepodobný symbol a C(**s**k) pre menej pravdepodobný symbol.

A(**s**) je binárna premenná postupnosti **s** vyjadrujúca veľkosť subintervalu pravdepodobnosti, pričom jej začiatočná hodnota pred zakódovaním prvého symbolu (keď **s** je prázdna postupnosť) je rovná 0.1111...1. A(**s**m) je takáto binárna premenná veľkosti intervalu postupnosti **s** pre viac pravdepodobný symbol a A(**s**k) pre menej pravdepodobný symbol.

Nech q je počet bitov aritmetickej jednotky, t.j. počet binárnych číslic pri aritmetických operáciách určených pre kódovanie a dekódovanie. Nech Q je celé číslo z intervalu {1,2,…,q-1} a vyjadruje posun. Hodnota Q je určená aproximáciou pravdepodobnosti menej pravdepodobného symbolu k.

Podstata binárneho aritmetického kódovania spočíva v kódovaní binárnej postupnosti *s* Eliasovym kódom [3]. Eliasov kód pracuje na princípe postupného delenia začiatočného intervalu <0,1) v závislosti od pravdepodobnosti výskytu aktuálneho symbolu vstupnej binárnej postupnosti **s**. Aritmetický kód binárnej postupnosti **s** sa generuje rekurzívne pre každý symbol tejto postupnosti použitím nasledujúcich rovníc, pričom sa použil spôsob AK bez použitia operácie násobenia [4]

$$A(\mathbf{s}k) = A(\mathbf{s}) \,.\, 2^{-Q} \tag{1}$$

$$A(\mathbf{s}m) = < A(\mathbf{s}) - A(\mathbf{s}k) > \tag{2}$$

$$C(\mathbf{s}m) = C(s) \tag{3}$$

$$C(\mathbf{s}k) = C(s)+A(sm) \tag{4}$$

V rov. (1) sa síce vyskytuje operácia násobenia, avšak výraz $A(\mathbf{s}).2^{-Q}$ znamená posun A(**s**) o Q bitov doprava. Operácia orezania v rov.(2) zabezpečuje, že nedôjde k pretečeniu q bitovej aritmetickej jednotky pre operácie s relevantnou časťou premennej A(**s**)., t.j. jej hodnota bude orezaná na q významových bitov. Pod relevantnou časťou premennej A(**s**) rozumieme tú časť jej bitov, ktorá sa zúčastňuje na aritmetických operáciách. Podstatou tejto metódy je aproximácia pravdepodobnosti menej pravdepodobnejšieho symbolu na hodnotu 2^{-Q}. Týmto spôsobom orezaná premenná bude obsahovať q-1 bitov za najviac významovým bitom a označujeme ju v zátvorkách < >. Napr nech B = 0.0011 , D = 0.11 a q = 2, potom < B.D > =0.0010, keďže B.D = 0.001001 sa orezáva na 2 významové bity.

Postupným kódovaním symbolov z binárnej postupnosti **s** sa získava premenná C(**s**). Po zakódovaní posledného symbolu tak máme výsledné binárne premenné C(**s**) a A(**s**). Výsledný aritmetický kód musí byť binárna hodnota z polootvoreného intervalu $<C(\mathbf{s}),\ C(\mathbf{s}) + A(\mathbf{s})\ )$. Hodnotu z daného intervalu zvolíme tak, aby obsahovala čo najmenší počet platných binárnych číslic. Týmto spôsobom však nie je možné zaznamenať informáciu o tom, ktorý symbol v poradí je posledným symbolom postupnosti **s**. Aby dekodér nepokračoval v procese dekódovania aj po dekódovaní posledného symbolu postupnosti **s**, musíme prenášať aj informáciu o dĺžke postupnosti.

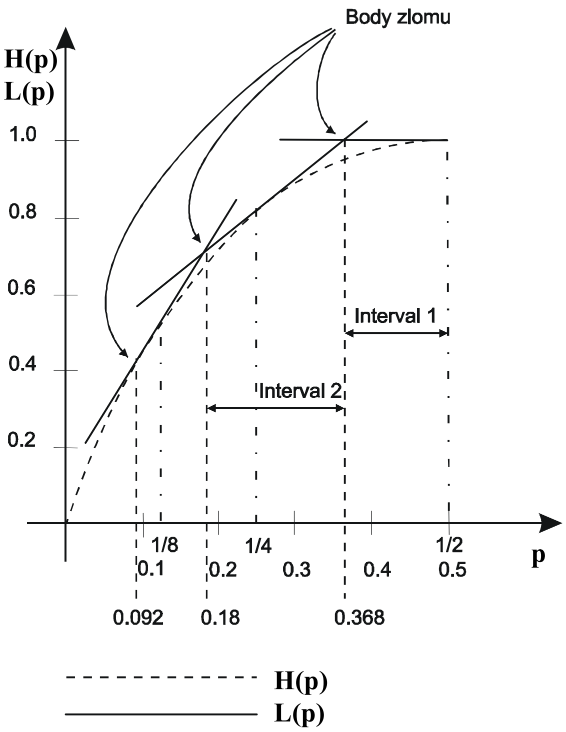
Tento spôsob kódovania síce obišiel operáciu násobenia a pre rekurzívny výpočet premennej A postačuje q-bitová aritmetická jednotka, avšak pre premennú C takúto aritmetickú jednotku použiť nemôžme. Ak by sme použili pre relevantnú časť premennej C 4-bitovú aritmetickú jednotku (rovnakú ako pre A), tak v niektorých prípadoch dochádza k operácii, ktorá presahuje rozsah tejto aritmetickej jednotky a došlo by k chybe. Problém, ktorý týmto vzniká je známy ako pretečenie [5]. Vhodnou úpravou algoritmu kódovania je možné tento problém odstrániť a tak môžeme použiť 4-bitovú aritmetickú jednotku aj pre premennú C.

Aby sme odstránili pretečenie, tak premenná C bude obsahovať ešte jednu relevantnú časť, ktorá bude zapísaná v ďalšom registri. V našom prípade je táto druhá časť zapísaná v 3-bitovom registri a nachádza sa bezprostredne pred dvojkovou čiarkou. Zvyšné bity premennej C sa počas kódovania meniť nebudú a môžeme ich vysielať rovno do dekodéra, alebo ukladať do toku dát. Ak sa však druhá časť premennej C naplní len bitmi 1, po zakódovaní ďalšieho symbolu by mohlo dôjsť ku pretečeniu. Potom tú časť premennej, ktorá sa nachádza pred dvojkovou čiarkou posunieme raz doľava. Na základe týchto princípov sa zakóduje celá binárna postupnosť **s**.

2.2. Aproximácia pravdepodobnosti výskytu symbolov

V predchádzajúcej časti bolo uvedené, že pravdepodobnosť výskytu menej pravdepodobného symbolu aproximujeme hodnotou 2^{-Q}, čím sa odstraňujú operácie násobenia a spôsob jej určenia vyplýva z teórie entrópie [2].

Na obr. 1 je zobrazená funkcia binárnej entropie nultého rádu H(p) v závislosti od pravdepodobnosti p [4]. Nájdeme takú lineárnu lomenú funkciu, ktorá sa najviac blíži k entropii H(p) a aby každý lineárny úsek ležal na intervale pravdepodobnosti, v ktorom sa nachádza jeden bod 2^{-Q}. Takou funkciou je funkcia L(p), ktorá je vyznačená na obr. 1 a pre každý bod 2^{-Q} potom dostávame interval, na ktorom lineárny úsek L(p) aproximuje entrópiu H(p). Takto dané intervaly slúžia na aproximáciu pravdepodobnosti, pričom pravdepodobnosť určitého intervalu aproximujeme hodnotou 2^{-Q} prislúchajúcu tomuto intervalu.



Obr. 1 Aproximácia L(p) a entropia H(p) v závislosti od pravdepodobnosti p
Fig. 1 Aproximation L(p) and entropy H(p) in dependancy on probability p

Daná aproximácia síce znižuje efektívnosť kódovania, ale na druhej strane umožňuje požiť metódy kódovania a dekódovania bez operácie násobenia a to je pre technickú realizáciu výhodné.

interval	Q	interval	Q
< 0.5 ; 0.368)	1	< 0.0027 ; 0.0013)	9
< 0.368 ; 0.18)	2	< 0.0013 ; 0.0006)	10
< 0.18 ; 0.092)	3	< 0.0006 ; 0.00028)	11
< 0.092 ; 0.045)	4	< 0.00028 ; 0.00012)	12
< 0.045 ; 0.022)	5	< 0.00012 ; 0.00005)	13
< 0.022 ; 0.011)	6	< 0.00005 ; 0.00002)	14
< 0.011 ; 0.0055)	7	< 0.00002 ; 0)	15
< 0.0055; 0.0027)	8	-	-

Tab. 1 Intervaly pravdepodobnosti a im prislúchajúce hodnoty Q
Tab. 1 The intervals of probabilities p and their values of Q

Hodnoty pre 15 intervalov pravdepodobnosti a ich prislúchajúce hodnoty Q sú v tab. 1.

2.3. Dekódovanie

Každá metóda kódovania má svoju zodpovedajúcu metódu dekódovania a každá z nich má svoj špecifický postup. Aritmetické dekódovanie je založené na nasledovných rovniciach tvaru

$$A(sk) = A(s) \,.\, 2\text{-}Q \qquad (5)$$

$$A(sm) = \langle A(s) - A(sk) \rangle \qquad (6)$$

$$C(s) < A(sm) \;\Rightarrow\; y = m;\; C(sy) = C(s) \qquad (7)$$

$$C(s) \geq A(sm) \;\Rightarrow\; y = k;\; C(sy) = C(s) - A(sm) \qquad (8)$$

Rov. (7) určuje, že ak hodnota C(**s**) je menšia ako A(**s**m), tak sa bude dekódovať viac pravdepodobný symbol a hodnota C(**s**y), ktorá je potrebná pre dekódovanie ďalšieho symbolu je rovná C(**s**). Ak platí podmienka v rov. (8), tak sa bude dekódovať menej pravdepodobný symbol a C(**s**y) určíme odčítaním A(**s**m) od C(**s**). Pre premennú A(**s**) platia rovnaké pravidlá ako pri kódovaní, t.j. A(**s**) = A(**s**m), ak bol dekódovaný viac pravdepodobný symbol a naopak [8].

Na začiatku dekódovacieho procesu nastavíme hodnotu A(**s**) = 0.11...1 a C(**s**) na hodnotu, ktorá určuje výsledný aritmetický kód reťazca **s**. Rekurziou rov. (5) až (8) postupne dekódujeme pôvodný reťazec **s**. Dekódovanie je ukončené po dekódovaní posledného symbolu reťazca **s**. Dĺžka reťazca **s** sa neprenáša v kóde, preto je potrebné ju prenášať ako prídavnú informáciu.

3. STAVOVÉ ARITMETICKÉ KÓDOVANIE A MODELY PRE BINÁRNE OBRAZY

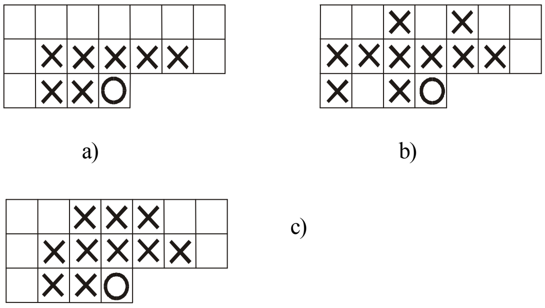
Stavové aritmetické kódovanie (SAK) patrí medzi entropické kódovanie a využíva medzisymbolové závislosti, ktoré sú založené na výbere stavov. Stav je určitá, vopred zvolená, skupina symbolov prislúchajúca aktuálnemu symbolu, ktorý sa má kódovať.

Pre aritmetický kód bez použitia operácie násobenia je potrebné poznať hodnotu Q a informáciu o tom, ktorý symbol je pravdepodobnejší. Tieto informácie je potrebné poznať pre každý symbol postupnosti a musia byť zhodné na strane kodéra aj dekodéra, pričom sú obsiahnuté v modeli. Vo všeobecnosti model určuje vzťah aktuálne kódovaného symbolu a symbolov, ktoré boli kódované pred ním. Tento model má úlohu estimátora, ktorý odhaduje pravdepodobnosť výskytu aktuálneho symbolu na základe predchádzajúcich symbolov.

3.1. Model SAK pre binárne obrazy

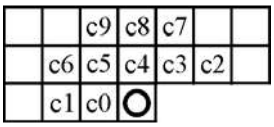
Pre binárne obrazy je model charakterizovaný šablónou, ktorá určuje rozloženie jednotlivých bodov obrazu voči aktuálnemu bodu. Tvar šablóny môže byť ľubovoľný, ale musí byť v súlade s postupom kódovania jednotlivých obrazových prvkov (op) binárneho obrazu. To znamená, že ak tento obraz kódujeme po riadkoch z ľavého horného bodu smerom k pravému dolnému bodu, tak všetky body šablóny musia ležať pred alebo nad aktuálnym bodom.

Na obr. 2 sú zobrazené tvary niektorých známych šablón [4], [5]. Aktuálny bod je označený krúžkom a krížikmi sú vyznačené ostatné body šablóny. Šablóny na obr. 2a a 2b boli navrhnuté Langdonom a Rissanenom a šablóna na obr. 2c sa používa pre JBIG algoritmus [3], [7].



Obr. 2 Tvary šablón a) 7-bodová, b) 10-bodová navrhnuté Langdonom a Rissanenom, c) 10-bodová pre JBIG algoritmus
Fig. 2 Shapes of templates a) 7-pel ,b) 10-pel designed by Langdon and Rissanen, c) 10-pel for JBIG algorithm

Jednotlivé body šablóny po jej preložení nad binárny obraz v mieste aktuálneho bodu určia stav v tomto bode. Na základe hodnôt op v bodoch obrazu určených šablónou vypočítame hodnotu tohto stavu. Na obr. 2c označme predchádzajúce body šablóny hodnotami op c0 až c9, ako je to vidno na obr. 3.



Obr. 3 Hodnoty op binárneho obrazu v predchádzajúcich bodoch šablóny
Fig. 3 Values of pel of binary image at previous points of template

Analyzujme binárny obraz pomocou šablóny na obr. 3, potom po preložení tejto šablóny nad obraz vypočítame stav aktuálneho bodu pomocou hodnôt označených c0 až c9. Napr. hodnota c5 = 0, ak zodpovedá čiernemu a c5 = 1 pre biely bod binárneho obrazu. Pre body šablóny mimo obrazu predpokladáme nulové hodnoty. Stav S(y) aktuálneho bodu vypočítame pomocou rov. (9), pričom y predstavuje aktuálny bod a n=*10* je počet bodov šablóny [4]

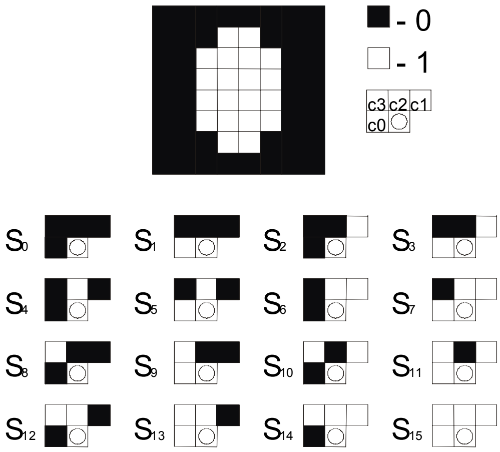
$$S(y) = \sum_{i=0}^{n-1} ci.2^i = c0 + 2.c1 + 4.c2 + 8.c3 + 16.c4 +$$
$$+ 32.c5 + 64.c6 + 128.c7 + 256.c8 + 512.c9 \qquad (9)$$

Pre určenie modelu binárneho obrazu je potrebné zistiť stav v každom bode tohto obrazu. Po preskúmaní celého obrazu určíme, ktorý stav sa koľko krát vyskytol v danom obraze a pre akú hodnotu aktuálneho op, pričom táto je 1 pre biely bod a 0 pre čierny bod. Potom zistíme, ktorá z týchto hodnôt sa častejšie vyskytuje v danom stave. Nakoniec vypočítame podmienené pravdepodobnosti uvedených hodnôt, resp. čierneho alebo bieleho aktuálneho bodu v jednotlivých stavoch

$$p(c = v \mid S_k) = p(S_{v,k}) = \frac{n_{v,k}}{n_k} \qquad (10)$$

kde S_k je stav určený rov.(9); hodnota op v = 0 alebo 1; $S_{v,k}$ je stav S_k s hodnotou op rovnou v ; n_k je počet stavov S_k ; $n_{v,k}$ je počet stavov $S_{v,k}$; v prípade, že n_k je nulové definujme, $p(S_{v,k}) = 0$.

Výsledné pravdepodobnosti tvoria model obrazu. Pre kódovanie bez použitia operácií násobenia vypočítané podmienené pravdepodobnosti $p(S_{v,k})$ aproximujeme hodnotou 2^{-Q}. Pre binárne obrazy budeme používať 16-bitovú aritmetickú jednotku, t.j. q = 16. Na zaznamenanie hodnoty Q potom postačia 4 bity. Pre určitý stav potrebujeme mať informáciu o tom, ktorá hodnota op sa viacej vyskytuje t.j. m a hodnotu Q. Potom pre jeden stav potrebujeme 5 bitov. Pre danú 10-bodovú šablónu, ktorá obsahuje 2^{10} = 1024 stavov potrebujeme 5120 bitov. Týmito bitmi je model binárneho obrazu úplne určený. Vytvorenie modelu binárneho obrazu si ukážeme pomocou nasledujúceho jednoduchého príkladu na obr. 4. Binárny obraz má rozmer 8x8 op a šablóna je 4-bodová kvôli jednoduchosti.



Obr. 4 Určenie modelu binárneho obrazu
Fig. 4 Determination of binary image model

Výsledky analýzy jednotlivých stavov sú v tab.2 (ČB je čierny bod a BB je biely bod). Ak nejaký stav sa nevyskytol ani raz, potom uvažujeme, že viac vyskytujúca sa hodnota m = 0, ktorá zodpovedá čiernému bodu a Q = 1.

stav	výs-kyt	výskyt		podmien. pravdep.		m	Q
		ČB	BB	ČB	BB		
0	29	28	1	0,965	0,034	0	5
1	3	2	1	0,667	0,333	0	2
2	6	5	1	0,833	0,167	0	3
3	0	0	0	0	0	0	1
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1
6	5	2	3	0,4	0,6	1	1
7	1	0	1	0	1	1	15
8	2	2	0	1	0	0	15
9	4	3	1	0,75	0,25	0	2
10	0	0	0	0	0	0	1
11	0	0	0	0	0	0	1
12	1	1	0	1	0	0	15
13	5	1	4	0,2	0,8	1	2
14	1	0	1	0	1	1	15
15	7	0	7	0	1	1	15

Tab. 2 Analýza stavov v binárnom obraze na obr. 4
Tab. 2 State analyzis of binary image at Fig. 4

Výsledný model binárneho obrazu na obr. 4 potom tvorí 16 usporiadaných dvojíc [m, Q] t.j. { [0,5] , [0,2] , [0,3] , [0,1] , [0,1] , [0,1] , [1,1] , [1,15] , [0,15] , [0,2] , [0,1] , [0,1] , [0,15] , [1,2] , [1,15] , [1,15] }.

3.2. Nemenný model binárnych obrazov

O nemennom modeli hovoríme vtedy, ak model obrazu je vytvorený ešte pred jeho samotným kódovaním. To znamená, že skôr ako začneme obraz kódovať, musíme prejsť celým obrazom a vytvoriť tak model spôsobom uvedeným v predchádzajúcej časti. Počas kódovania binárneho obrazu pre každý bod musíme pomocou šablóny určiť jeho stav. Na základe tohto stavu vyberieme už z estimovaného modelu danú usporiadanú dvojicu. Hodnoty m a Q z usporiadanej dvojice sa potom priamo zúčastňujú v procese kódovania. Keďže model sa musí nachádzať aj na strane dekodéra, tak sa musí spolu so samotným kódom prenášať aj model. Pre 10-bodovú šablónu tak musíme okrem samotného kódu preniesť ešte 5120 bitov.

Existuje spôsob ako obísť prenášanie modelu a to tak, že na kódovanie a dekódovanie použijeme model z nejakej banky modelov. Táto banka modelov, potom musí byť prístupná pre kodér aj dekodér. Modely v tejto banke je potom rozumné vytvoriť z obrazov s často sa vyskytujúcou štruktúrou. Ešte lepšie je tieto modely vytvoriť spriemernením štatistík obrazov s často sa vyskytujúcou štruktúrou.

Aby bol kódovaný binárny obraz dekódovaný na pôvodný, tak musíme zabezpečiť aby na strane kodéra aj dekodéra bol rovnaký model aj tvar šablóny, rovnaký počet bitov aritmetickej jednotky a zodpovedajúca štruktúra premenných, metóda dekódovania musí zodpovedať metóde kódovania.

3.3. Adaptívny model binárnych obrazov

Zatiaľčo nemenný model musel byť vytvorený už pred samotným kódovaním, adaptívny model sa vytvára počas procesu kódovania. Majme tabuľku, v ktorej je uvedený počet výskytov všetkých stavov pre obidve hodnoty op. Pred začatím samotného procesu kódovania obsah tabuľky vynulujeme, t.j. počet výskytov všetkých stavov s hodnotami op 0 alebo 1 bude rovný nule. Pri kódovaní každého op binárneho obrazu budeme postupovať takto:

1. Zistíme stav aktuálneho bodu S(y).
2. Pre daný stav určíme z tabuľky podmienené pravdepodobnosti čierneho a bieleho bodu.
3. Určíme hodnoty m, Q a zakódujeme aktuálny op binárneho obrazu .
4. Zvýšime počet výskytov aktuálneho stavu op o jedna a pokračujeme s ďalším bodom obrazu a krokom 1.

Týmto spôsobom zakódujeme celý binárny obraz. Postupným kódovaním jednotlivých op sa tak pravdepodobnosti v tabuľke blížia k pravdepodobnostiam nemenného modelu. Po zakódovaní posledného op tohto obrazu sú pravdepodobnosti adaptívneho modelu zhodné s pravdepodobnosťami nemenného modelu. Pri dekódovaní obrazu opäť vynulujeme tabuľku a postupujeme tými istými krokmi ako pri kódovaní s tým, že v 3. kroku aktuálny op dekódujeme. Táto metóda má výhodu oproti metóde s nemenným modelom v tom, že nie je potrebné prenášať spolu so samotným kódom aj model alebo používať banku modelov. Na druhej strane však samotný kód je dlhší v porovnaní s kódom získaným pomocou nemenného modelu z dôvodu postupnej adaptácie štatistík.

4. SIMULÁCIA SAK A DOSIAHNUTÉ VÝSLEDKY

Navrhnuté metódy stavového aritmetického kódovania binárnych obrazov boli simulované na osobnom počítači rady PC s operačným systémom Windows 95/98/ME pomocou programových prostriedkov Builder C++ 4.0. Vzhľadom na najjednoduchšiu implementáciu navrhnutých metód bolo použité kódovanie a dekódovanie bez použitia operácie násobenia a blokovaním pretečenia. Ako štandardná šablóna modelu SAK bola použitá 10-bodová šablóna pre JBIG algoritmus uvedená na obr. 2c.

Binárne obrazy, ktoré boli použité pri simulácii SAK sme vybrali tak, aby sme z nich mohli vytvárať skupiny s určitými vlastnosťami. Napr. môžu to byť skupiny binárnych obrazov malej (256x256), strednej (640x480) a veľkej (1024x768) veľkosti. Ďalej skupiny vytvorené podľa zložitosti (detailnosti) týchto obrazov, ktoré tvorili jednoduché (málo detailov), stredne zložité (viac detailov) a zložité (veľa detailov) obrazy. Pri výbere binárnych obrazov si však musíme uvedomiť, že detailnosť obrazu určujeme subjektívne, zatiaľ čo rozmery obrazu sú dané objektívne. Na obr. 5 sú uvedené niektoré skúmané binárne obrazy.

Skúmali sme účinnosť kódovania jednotlivých metód SAK pre vybrané obrazy v závislosti od ich uvedených vlastnosti. Vytvorili sme spoločné modely pre tieto skupiny a pomocou nich sme kódovali jednotlivé binárne obrazy.

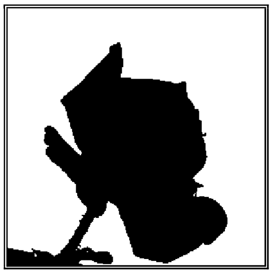
Pre metódu SAK s nemenným modelom sme vytvorili tieto modely :

- jednoduchý - model jednoduchých obrazov, t.j. ruža256, vtáky640 a kôň1024,
- stredný - model stredne zložitých obrazov, t.j. papagáj256, had640 a tučniaky1024,
- zložitý - model zložitých obrazov, t.j. slnečnica256, kvety640 a kačice1024,
- spoločný - model všetkých obrazov,
- vlastný – model jedného obrazu.

Najprv sme skúmali efektívnosť metód SAK s vlastným nemenným (NM) alebo adaptívnym (AM) modelom a klasického aritmetického kódovania (KAK). Efektívnosť kódovania je daná pomerom dĺžky kódu a celkovým počtom bitov binárneho obrazu uvedený v percentách. Hodnoty tohto pomeru jednotlivých metód sú v tab. 3. S nárastom hodnoty pomeru klesá efektívnosť kódovania.

Z uvedených hodnôt v tab. 3. vyplývajú nasledujúce skutočnosti:

- Efektívnosť pre stacionárny a adaptívny model klesá so zložitosťou a narastá s rozmerom obrazu.
- Efektívnosť SAK s AM je vyššia ako SAK s NM pri jednoduchých obrazoch, s narastajúcou zložitosťou sa však efektívnejšou stáva SAK s NM. Je to spôsobené tým, že pri zložitejších obrazoch dochádza k „pomalšej adaptácií“ štatistík u SAK s AM.
- Pri jednoduchej zložitosti klesá rozdiel medzi pomerom efektívnosti SAK s AM a NM s narastajúcimi rozmermi obrazu. Príčinou toho je, že v dĺžke kódu pri NM je zahrnutá aj dĺžka modelu. S narastajúcimi rozmermi sa však dĺžka modelu stáva zanedbateľnou voči dĺžke vlastného SAK kódu. Efektívnosť KAK je tiež závislá len od pomeru počtu čiernych a bielych bodov binárneho obrazu. Najlepšia efektívnosť bola dosiahnutá pre obraz “kôň1024“, kde počet bielych bodov prevláda nad počtom čiernych bodov.



ruža

papagáj

slnečnice

Obr. 5 Binárne obrazy rozmeru 256x256 op
Fig. 5 Binary images of the size 256x256 pel

Z uvedených skutočností potom vyplýva, že pre kódovanie malých a jednoduchých binárnych obrazov je výhodné používať adaptívny model a pre väčšie a zložitejšie obrazy používať nemenný model. Klasický aritmetický kód má význam použiť len vtedy, ak počet čierných bodov je ďaleko väčší ako počet bielych bodov alebo naopak.

Ďalej sme skúmali efektívnosť metód SAK s NM v závislosti od použitého modelu. Každý binárny obraz sme kódovali pomocou nemenných modelov: jednoduchý, stredný, zložitý a spoločný. Pre porovnanie uvedieme aj hodnotu pomeru efektívnosti pre vlastný NM určeného len z čistej dĺžky SAK kódu bez dĺžky kódu samotného modelu. Je to minimálna dĺžka kódu aká sa dá pomocou ktorejkoľvek metódy SAK vôbec dosiahnuť. Hodnoty pomeru efektívnosti SAK pre jednotlivé modely sú v tab. 4.

Z hodnôt uvedených v tejto tabuľke vyplývajú tieto poznatky:

- Najlepšej efektívnnosti kódovania binárneho obrazu sa dosiahlo vtedy, ak pre daný obraz bol použitý model, ktorý bol vytvorený z obrazov s podobnou zložitosťou ako kódovaný obraz

Obraz	**ruža256**	**papagáj256**	**slnečnica256**	**vtáky640**	**had640**
SAK s NM	10.420%	19.653%	55.502%	5.188%	19.299%
SAK s AM	3.363%	18.196%	64.090%	4.201%	21.538%
KAK	95,162%	96,675%	96,963%	73,34%	90,931%
obraz	**kvety640**	**kôň1024**	**tučniaky1024**	**kačice1024**	**-**
SAK s NM	25.994%	2.528%	13.892%	18.316%	-
SAK s AM	29.459%	2.213%	14.587%	18.627%	-
KAK	98,047%	69,09%	99,996%	99,831%	-

Tab. 3 Hodnoty pomeru efektívnosti jednotlivých metód aritmetického kódovania pre vybrané obrazy
Tab. 3 Values of effective ratio of particular arithmetic methods for choice images

a dosiahnutý pomer efektívnosti je blízky pomeru efektívnosti, ktorý bol dosiahnutý pre vlastný model.

- S nárastom rozdielu zložitosti binárneho obrazu a modelu narastá aj hodnota tohto pomeru.
- Druhá najlepšia efektívnosť bola dosiahnutá pri použití modelu spoločný, ktorý bol vytvorený zo všetkých obrazov.

Z uvedeného vyplýva, že pre efektívne SAK je vhodné vytvoriť banku modelov s rôzne zložitými modelmi a pri kódovaní binárneho obrazu potom použiť model s podobnou zložitosťou. Tu však vzniká problém v tom, že zložitosť nie je vyjadrená kvantitatívne, ale len kvalitatívne. Naviac zložitosť určujeme len subjektívne, aj napriek tomu, že by bolo možné ju určiť objektívne pomocou výskytu jednotlivých stavov mnoho prvkovej šablóny, ale bola by to veľmi zložitá úloha. Výhodné je vytvoriť model z veľkého počtu obrazov rôznej zložitosti a pomocou neho potom kódovať binárne obrazy. Takýto postup je síce menej efektívny, ale nevyžaduje odhad zložitosti binárneho obrazu.

Sledovali sme tiež vplyv šablóny na efektívnosť kódovania. Kódovali sme binárne obrazy postupne pomocou šablón s počtom bodov 0 až 15. Tvar šablón sme zvolili taký, aby jednotlivé body boli v okolí aktuálneho bodu. Zistili sme nasledujúce poznatky:

- Metóda SAK s NM pri použití prázdnej šablóny (len s aktuálnym bodom) je totožná s metódou klasického aritmetického kódovania t.j. bezstavové SAK. Ak šablóna nemá žiadny prvok, tak sa vlastne jedná o bezstavové aritmetické kódovanie.
- Pre NM postupne s nárastom počtu bodov šablóny sa efektívnosť SAK najprv zlepšuje a od určitého počtu týchto bodov sa začne zhoršovať. Príčinou je, že s nárastom bodov šablóny klesá síce dĺžka samotného SAK kódu, ale exponenciálne narastá dĺžka kódu modelu, ktorá je v celkovej dĺžke SAK tiež zahrnutá. Od určitého počtu bodov potom prírastok dĺžky kódu modelu je väčší ako úbytok dĺžky SAK kódu. Ku zmene dochádza pri počte okolo 8 bodov.
- Pre AM sa tiež s nárastom počtu bodov efektívnosť SAK najprv zvyšuje a od určitého počtu sa začne zmenšovať. V tomto prípade je príčinou to, že s nárastom počtu bodov šablóny efektícnosť stúpa v dôsledku nárastu počtu stavov, ale ak počet bodov je už veľký, tak dochádza k pomalšej adaptácii štatistík. Ku zmene dochádza tiež v okolí 8 bodov.
- Ak efektívnosť SAK s AM a NM, tak pri malom počte bodov šablóny je nemenný model o niečo lepší, ale od určitého počtu bodov sa stáva efektívnejším adaptívny model a s ďalším nárastom počtu bodov sa rozdiel medzi efektívnosťami veľmi zväčšuje. Príčinou je to, že s nárastom počtu bodov šablóny dĺžka kódu modelu narastá oveľa rýchlejšie, ako nárast dĺžky kódu SAK v dôsledku pomalej adaptácie štatistík pre adaptívny model.

Ďalej môžeme povedať, že ak na kódovanie nepoužívame banku modelov, tak je najvhodnejšia šablóna s počtom okolo 8 bodov. Šablóny s väčším počtom bodov sú potom vhodné, ak používame banku modelov.

5. ZÁVER

Článok bol zameraný na stavové aritmetické kódovanie. Najprv bol popisovaný základný algoritmus aritmetického kódovania bez použitia operácie násobenia, spôsob aproximácie podmienených pravdepodobnosti výskytu symbolov a proces dekódovania. Boli skúmané, popisované a vyhodnocované jednotlivé metódy stavového aritmetického kódovania binárnych obrazov. Bolo popisované aj vytváranie modelov pre binárne obrazy. V súvislosti s týmto bola aj popisovaná šablóna, ktorá určuje stav jednotlivých bodov binárneho obrazu. V poslednej časti boli skúmané a navzájom porovnávané jednotlivé metódy SAK na základe účinnosti kódovania daných binárnych obrazov. Bola sledovaná závislosť efektívnosti metódy SAK s nemenným modelom v závislosti od zložitosti modelu. A taktiež aj závislosť efektívnosti jednotlivých metód SAK od zvoleného počtu bodov šablóny.

Z hodnôt pomeru efektívnosti vyplynulo, že stavové aritmetické kódovanie poskytuje dobré výsledky v kódovaní binárnych obrazov. Tieto efektívnosti sú podstatne vyššie ako efektívnosť klasického aritmetického kódovania a tiež Hufmanovho kódu. Nevýhodou SAK je, že má zložitejšiu implementáciu kodéra a dekodéra.

model \ obraz	**ruža256**	**papagáj256**	**slnečnica256**	**vtáky640**	**had640**
jednoduchý	2.870%	17.215%	93.465%	3.727%	27.713%
stredný	3.539%	13.045%	57.205%	4.127%	18.363%
zložitý	4.225%	13.593%	52.914%	4.822%	18.745%
spoločný	3.476%	13.194%	54.823%	4.229%	18.690%
vlastný	2.608%	11.841%	47.690%	3.521%	17.633%
model \ obraz	**kvety640**	**kôň1024**	**tučniaky1024**	**kačice1024**	**-**
jednoduchý	47.140%	1.910%	17.101%	22.466%	-
stredný	29.213%	2.396%	13.382%	18.532%	-
zložitý	25.322%	2.982%	13.670%	18.150%	-
spoločný	25.879%	2.382%	13.434%	18.184%	-
vlastný	24.327%	1.877%	13.241%	17.665%	-

Tab. 4 Hodnoty pomeru efektívnosti jednotlivých metód SAK pre rôzne modely a vybrané obrazy
Tab. 4 Values of effective ratio of particular method of arithmetic coding for different models and images

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

LITERATÚRA

[1] Mihalík, J.: Kódovanie obrazu vo videokomunikáciach. Mercury, ISBN 80-89061-47-8, Košice, 2001.
[2] Mihalík, J.: Číslicové spracovanie signálov. Alfa Bratislava, 1987.
[3] Howard, G. P. - Vitter, S. J.: Analysis of Arithmetic Coding for Data Compression. Information and Management, Vol. 28, No. 6, 1992, pp. 749-763.
[4] Langdon, G. G. - Rissanen, J.: Compression of Black - White Images with Arithmetic Coding. IEEE Trans. on Communications, Vol. COM-29, No. 6, 1981, pp. 858-867.
[5] Gladišová, I. - Mihalík, J.: Stavové aritmetické kódovanie binárneho tvaru videoobjektu. Zborník vedeckej konferencie s medzinárodnou účasťou "Nové smery v spracovaní signálov VI", Tatranské Zruby, 2002, s.219-222.
[6] Gladišová, I. – Mihalík, J.: Mriežkové vektorové kvantovanie s entropickým kódovaním. Zborník ved. konf. s medzinárodnou účasťou „Nové smery v spracovaní signálov V“, L. Mikuláš, 2000, s. 355 – 360.
[7] Brady, N. – Bossen, F.: Shape Compresion of Moving Objects Using Context-based Arithmetic Encoding. Signal Proc : Image Communication, vol. 15, 2000, pp.. 601-617.
[8] 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.

BIOGRAPHY

Iveta Gladišová graduated from Technical University of Košice in 1984. She is employed at same university of Košice, where received her PhD Degree in Radioelectronics in 1997. She is Assistant Professor at the Department of Electronics and Multimedia Telecomunications , Faculty of Electrical Engineering and Informatics, Technical University, Košice. Her research interests include entropy coding, vector quantization and digital signal processing.

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 Videocommunication at the Department of Electronics and Multimedia Telecommunications. His research interests include information theory, image coding, digital image processing and videocommunication.