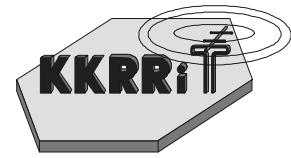


Tomasz Grajek*

Damian Karwowski[#]

Politechnika Poznańska, Instytut Elektroniki i Telekomunikacji
Zakład Telekomunikacji Multimedialnej i Radioelektroniki
ul. Piotrowo 3A, 60 – 965 Poznań

*tgrajak@multimedia.edu.pl, #dkarwow@multimedia.edu.pl



Kraków, 15-17 czerwca 2005

Złożoność obliczeniowa i efektywność kodowania entropijnego w standardzie H.264/AVC

Streszczenie: W niniejszym artykule przedstawiono algorytmy kodowania binarnego zdefiniowane w najnowszym standardzie kodowania sekwencji wizyjnych H.264/AVC. Przedstawiono ogólne założenia teoretyczne, dokonano wnikliwej analizy algorytmów kodowania entropijnego w kontekście ich złożoności obliczeniowej, a także efektywności kompresji. Przy użyciu sekwencji testowych przeprowadzono serię badań eksperymentalnych. Dokonano porównania złożoności obliczeniowej oraz efektywności algorytmicznie zoptymalizowanych wersji koderów i dekoderów binarnych dostępnych w H.264/AVC.

1. WSTĘP

W ciągu ostatnich kilkunastu lat, cyfrowe techniki kompresji obrazu i dźwięku całkowicie zmieniły sposób, w jaki tworzymy, przesyłamy i odbieramy informacje audiowizualne. Nie tylko przekształciły one istniejące aplikacje i usługi, ale także przyczyniły się do powstania całkiem nowych takich jak: wideo konferencje, transmisje satelitarne, telewizja wysokiej rozdzielczości, wideo na żądanie, itp.[8].

Obecnie najbardziej popularnym standardem kompresji sekwencji wizyjnych jest standard MPEG-2 (znany także jako ITU-T H.262) [1,2,4]. MPEG-2 jest szeroko stosowany w transmisji sygnałów telewizyjnych standardowej (SD) i wysokiej (HD) rozdzielczości przez systemy satelitarne, kablowe, jak i naziemne oraz jako standard przechowywania danych wizyjnych na płytach DVD. Jednakże rosnąca liczba usług, jak i zwiększająca się popularność telewizji wysokiej rozdzielczości stworzyła potrzebę większej efektywności kodowania. Co więcej, nowe systemy transmisyjne takie jak xDSL czy UMTS, które oferują łącza o przepustowości znacznie mniejszej niż typowe kanały rozsiewcze, powodują, że zwiększenie efektywności kodowania pozwoli na przesyłanie większej liczby kanałów telewizyjnych lub sygnałów o wyższej jakości przy zachowaniu istniejących przepustowości łącza [5,8].

Odzwierciedleniem aktualnego stanu technik kompresji wizyjnej jest nowy standard H.264/AVC, który na tle wcześniejszych standardów wyróżnia się znacznym wzrostem efektywności kompresji uzyskanej dzięki licznym usprawnieniom i modyfikacjom, w tym znacznie ulepszonemu kodowaniu entropijnemu [5,6].

W standardzie H.264/AVC zdefiniowano dwa różne algorytmy kodowania entropijnego:

- prostsze kodowanie o zmiennej długości słowa kodowego VLC (*Variable Length Coding*),

- bardziej efektywne i złożone adaptacyjne kodowanie arytmetyczne CABAC (*Context-based Adaptive Binary Arithmetic Coding*) [6,8].

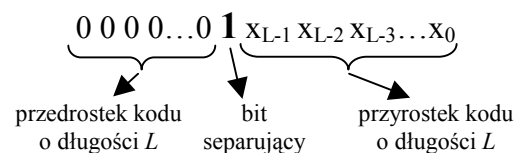
Ponieważ oczekuje się, że nowy standard H.264/AVC zastąpi starsze techniki kodowania i znajdzie zastosowanie w telewizji cyfrowej nowej generacji, ciekawym zagadnieniem jest zbadanie wpływu zastosowanych algorytmów kodowania binarnego na złożoność obliczeniową całego koderów i dekoderów oraz wymaganą prędkość transmisji. W tym celu autorzy przygotowali własne, zoptymalizowane algorytmicznie wersje koderów i dekoderów z uwzględnieniem cech programowych platformy Pentium IV, co pozwoliło na uzyskanie wiarygodnych wyników pomiarów.

2. KODOWANIE VLC

Kodowanie VLC jest prostszą metodą kodowania entropijnego w standardzie H.264/AVC i jest wykorzystywane w profilu podstawowym (*Baseline Profile*) [5]. Wykorzystuje dwie techniki kodowania binarnego:

- proste nieadaptacyjne kodowanie Exp-Golomba zerowego rzędu, stosowane dla zakodowania informacji sterujących i różnicowych wektorów ruchu;
- bardziej złożone kontekstowe kodowanie o zmiennej długości słowa kodowego CAVLC (*Context-Adaptive Variable Length Coding*) stosowane wyłącznie dla skwantowanych współczynników transformaty błędu predykcji.

Kody Exp-Golomba są kodami zmiennej długości. Cechują się prostotą składni i są optymalne dla elementów strumienia o wykładniczym rozkładzie prawdopodobieństwa [9]. Strukturę kodów Exp-Golomba przedstawiono na rys. 1.



Rys. 1. Struktura kodów Exp-Golomba

Kody Exp-Golomba składają się z:

- przedrostka kodu będącego ciągiem bitów zerowych,
- przyrostka kodu będącego ciągiem bitów informacyjnych,
- pojedynczego bitu o wartości '1' oddzielającego przedrostek kodu od przyrostka.

Istotną własnością kodów Exp-Golomba jest to, iż długości przedostka i przyrostka kodu są takie same. Kodowanie i dekodowanie Exp-Golomba cechuje niewielka złożoność obliczeniowa. Utworzenie kodu dla pewnej nieujemnej wartości x obejmuje następujące czynności:

- inkrementacja wartości x ;
- wyznaczenie pozycji najstarszej jedynek w reprezentacji binarnej nowo utworzonej wartości x ;
- długość L przedostka kodu jest równa wartości wyznaczonej wcześniej pozycji najstarszej jedynek, L młodszych bitów wartości x tworzy przyrostek kodu.

Zdekodowanie kodu Exp-Golomba sprowadza się do znalezienia pierwszej jedynek w słowie oraz odczytania przyrostka kodu. Na procesorach z rodziny Pentium czynność ta może zostać szybko wykonana jedną instrukcją asemblerową BSR (*Bit Scan Reverse*).

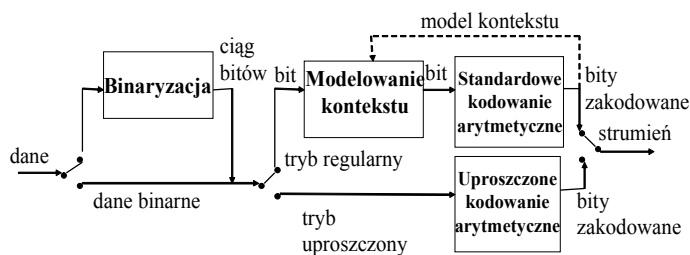
Dla skwantowanych współczynników transformaty błędu predykcji, stosuje się bardziej zaawansowane i złożone kodowanie CAVLC, ze względu na dużą ilość danych podlegających kodowaniu. Algorytm kodowania CAVLC jest kontekstowy i adaptacyjny. Kodowanie przebiega na dwóch poziomach adaptacji do bieżącej statystyki kodowanego sygnału. Dla każdego kodowanego elementu zdefiniowano kilka różnych tablic kodów zmiennej długości, biorąc pod uwagę odmienny charakter statystyczny poszczególnych danych. Stanowi to pierwszy etap adaptacji do statystyki kodowanych danych. Dodatkowo proces kodowania danego elementu jest uzależniony od jego wartości w sąsiednich blokach 4×4 , co stanowi drugi poziom adaptacji. W zależności od kontekstu kodowanego elementu, wybierana jest odpowiednia tablica kodów zmiennej długości, co pozwala na szybką i dokładną adaptację koderów do bieżącej statystyki sygnału. Zastosowanie koderów entropijnych działających w oparciu o kontekst kodowanych elementów znacząco zwiększa efektywność kodowania binarnego.

Kodowanie danych przy użyciu techniki CAVLC w porównaniu z dekodowaniem jest zadaniem stosunkowo prostym i nie wymaga od systemu dużej mocy obliczeniowej. Może ono zostać szybko zrealizowane poprzez tablicę LUT (*Look-up Table*), która dla kodowanego elementu od razu wyznacza słowo kodowe o zmiennej długości. Czas kodowania jest tutaj uzależniony od czasu dostępu do pamięci. Natomiast proces dekodowania CAVLC jest bardziej złożony ze względu na dużą liczbę możliwych słów kodowych. Dekoder musi analizować strumień danych bit po bicie i każdorazowo sprawdzać, czy przeczytany ciąg bitów nie stanowi już kodu VLC dla danego elementu składni strumienia. Sprawia to, iż algorytm przeszukiwania tablic kodów zmiennej długości jest najbardziej krytyczną częścią dekodera CAVLC i ma ogromny wpływ na czas dekodowania.

W autorskiej implementacji szybkie dekodowanie kodów zmiennej długości zostało zrealizowane w oparciu o efektywne przeszukiwanie drzew binarnych utworzonych ze wszystkich możliwych kodów zmiennej długości [3]. Uzyskano w ten sposób znaczące zmniejszenie złożoności obliczeniowej dekodera CAVLC, co zredukowało czasy dekodowania sekwencji wizyjnych.

3. KODOWANIE CABAC

W profilu głównym (*Main Profile*) standardu H.264/AVC zdefiniowano algorytm kodowania binarnego działający w oparciu o kontekstowe adaptacyjne kodowanie arytmetyczne CABAC [6,8]. W porównaniu z algorytmem kodowania VLC algorytm CABAC odznacza się wyższą efektywnością kodowania, jednakże proces kodowania i dekodowania obciążony jest większą złożonością obliczeniową. Schemat blokowy kodowania CABAC przedstawiono na rys. 2.



Rys. 2. Schemat blokowy koderów CABAC

Kodowanie algorytmem CABAC obejmuje trzy etapy:

- proces binaryzacji;
- modelowanie statystyczne (modelowanie kontekstu);
- binarne kodowanie arytmetyczne.

Częścią algorytmu CABAC jest binarny koder arytmetyczny. Przed właściwym procesem kodowania arytmetycznego wszystkie symbole niebinarne poddawane są tzw. procesowi binaryzacji. Proces binaryzacji polega na przyporządkowaniu niebinarnym symbolom pewnego ciągu symboli binarnych. Ze względu na różną statystykę kodowanych danych, w koderze CABAC zdefiniowano cztery typy binaryzacji. Pozwalają one na przyporządkowanie niebinarnym symbolom możliwie najkrótszych słów kodowych. Ma to znaczący wpływ na czas potrzebny na zakodowanie bądź zdekodowanie symboli danych. Po binaryzacji otrzymane symbole binarne dodatkowo podlegają kodowaniu arytmetycznemu, co znacznie zmniejsza redundancję międzysymbolową i zwiększa efektywność kodowania.

Binarny koder arytmetyczny koduje symbole binarne z uwzględnieniem prawdopodobieństw warunkowych ich wystąpienia. Aby zwiększyć efektywność kodowania, dla wszystkich kodowanych elementów składni strumienia zdefiniowano łącznie 399 kontekstów (modeli prawdopodobieństw). Konteksty są inicjowane na początku każdej warstwy, a proces inicjalizacji jest uzależniony od aktualnej statystyki kodowanego sygnału.

W procesie kodowania danego symbolu wykorzystuje się wiedzę o jego wartościach w sąsiednich blokach 4×4 , która służy wyznaczeniu prawdopodobieństwa warunkowego dla bieżącego symbolu. W zależności od statystyki ostatnio zakodowanych symboli danych ze zbioru dostępnych modeli prawdopodobieństw wybierany jest ten, który zapewnia optymalne kodowanie. Po każdym etapie zakodowania symbolu binarnego następuje uaktualnienie modelu prawdopodobieństwa. Pozwala to na skuteczne usunięcie redundancji między sąsiednimi bitami w binarnym ciągu i szybką reakcję koderów na zmianę statystyki kodowanego sygnału.

Etap modelowania kontekstu i binarnego kodowania arytmetycznego jest najbardziej czasochłonną częścią całego algorytmu kodowania i dekodowania CABAC. Możliwości optymalizacyjne tych części algorytmu są już jednak bardzo ograniczone, ponieważ na etapie definiowania techniki CABAC zostały one znacznie uproszczone. Aby zmniejszyć moc obliczeniową potrzebną na zakodowanie oraz zdekodowanie symbolu danych, w algorytmie CABAC zdefiniowano zbiór 64 prawdopodobieństw. Koder i dekodery arytmetyczne na podstawie aktualnej statystyki sygnału wybiera właściwe prawdopodobieństwo z tego zbioru, a dla poszczególnych symboli danych nie ma konieczności każdorazowego wyznaczenia prawdopodobieństwa warunkowego, co znacząco zmniejsza czas kodowania i dekodowania symboli. Dodatkowa redukcja czasu kodowania i dekodowania uzyskana została poprzez wprowadzenie dwóch trybów kodowania arytmetycznego: trybu regularnego i trybu uproszczonego, co przedstawiono na rys. 2. Kodowanie uproszczone jest stosowane tylko i wyłącznie dla symboli danych o równomiernym rozkładzie prawdopodobieństwa, a uproszczenie polega na pominięciu złożonego etapu modelowania statystycznego dla kodowanych symboli.

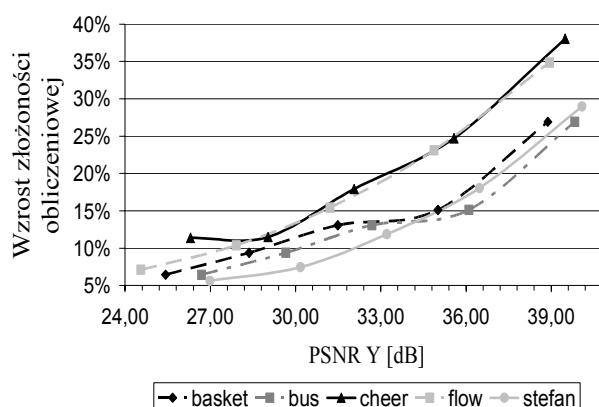
Znaczącemu uproszczeniu uległ również sam rdzeń koder i dekodery arytmetycznego. Praca binarnego koder arytmetycznego polega na każdorazowym dzieleniu aktualnego przedziału liczbowego $[L, L+R)$ o zakresie R na dwa podprzedziały odpowiadające symbolom binarnym 0 i 1. Długości podprzedziałów zależą od wartości prawdopodobieństw symboli binarnych 0 i 1. Dzielenie przedziału liczbowego $[L, L+R)$ na podprzedziały wymaga operacji mnożenia, która jest bardzo kosztowna zarówno w implementacji sprzętowej, jak i programowej. Na potrzeby algorytmu CABAC stworzony został alternatywny, wolny od operacji mnożenia binarny koder arytmetyczny tzw. modulo koder (M koder) [7,8]. Operacja mnożenia została zastąpiona predefiniowaną tablicą o rozmiarze 64×4 , zawierającą iloczyny zakresu przedziału i prawdopodobieństwa symbolu.

Algorytm dekodowania CABAC jest analogiczny do procesu kodowania. Rozpoczyna się on działaniem adaptacyjnego binarnego dekodera arytmetycznego, a kończy procesem debinaryzacji (binaryzacji odwrotnej). W przeciwieństwie do technik kodowania wykorzystujących algorytm Huffmana, proces kodowania arytmetycznego jest równie mocno złożony jak proces dekodowania i wymaga od systemu dużej mocy obliczeniowej.

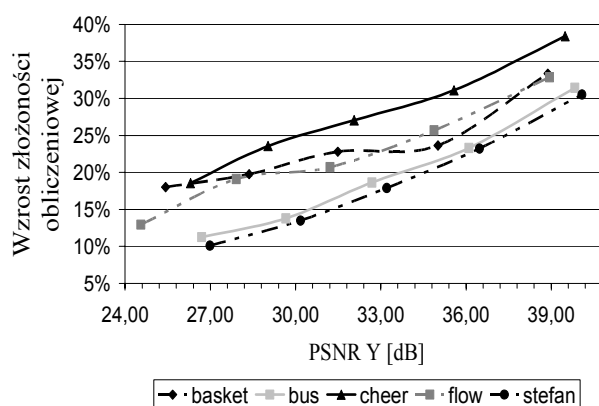
4. WYNIKI EKSPERYMENTALNE

W celu porównania złożoności obliczeniowej oraz efektywności algorytmów kodowania VLC i CABAC autorzy przeprowadzili serię badań eksperymentalnych z wykorzystaniem własnej, oryginalnej implementacji koder i dekodera. Kodek został zoptymalizowany algorytmicznie na platformę Pentium IV. Każdy pomiar obejmował zakodowanie lub zdekodowanie 200 ramek każdej sekwencji o rozdzielczości 4cif (704x576) i częstotliwości 25 Hz z wykorzystaniem predykcji wewnątrz- i między-obrazowej. Za wyznacznik złożoności obliczeniowej przyjęto czas kodowania i dekodowania poszczególnych sekwencji przy utrzymaniu takiej samej jakości mierzonej miarą PSNR dla składowej luminancji. Zależność ta dla koder

i dekodera została przedstawiona odpowiednio na wykresie 1 i 2.

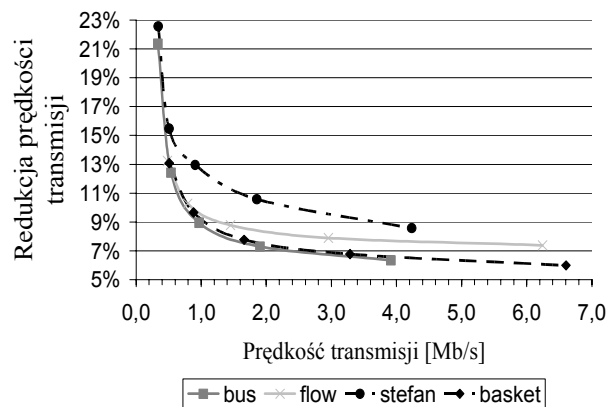


Wykres 1. Wzrost złożoności obliczeniowej koder z algorytmem CABAC w stosunku do VLC przy zachowaniu takiej samej jakości dla 5 sekwencji testowych.



Wykres 2. Wzrost złożoności obliczeniowej dekodera z algorytmem CABAC w stosunku do VLC przy zachowaniu takiej samej jakości dla 5 sekwencji testowych.

Z przedstawionych wykresów wynika, że zastosowanie algorytmu CABAC w porównaniu z VLC wymaga większych nakładów obliczeniowych. Wzrost złożoności obliczeniowej jest uzależniony od miary PSNR. Dla koder jest to $5 \div 40\%$, natomiast dla dekodera $9 \div 40\%$.



Wykres 3. Redukcja prędkości transmisji wynikająca z zastosowania kodowania entropijnego CABAC.

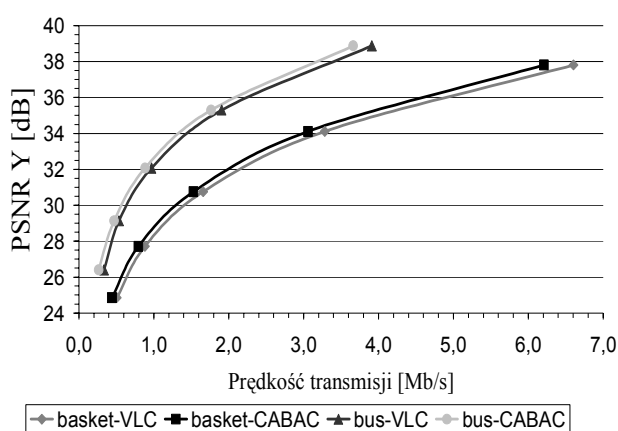
Większa złożoność algorytmu CABAC pozwala na uzyskanie mniejszej prędkości transmisji przy zachowa-

niu takiej samej jakości mierzonej miarą PSNR. Zysk kodowania zależy od charakteru kodowanego sygnału i może przekraczać nawet 20% dla prędkości poniżej 1 Mb/s, natomiast dla większych prędkości transmisji wynosi około 7%. Na wykresie 3 przedstawiono redukcję prędkości transmisji wynikającą z zastosowania bardziej efektywnego kodowania CABAC.

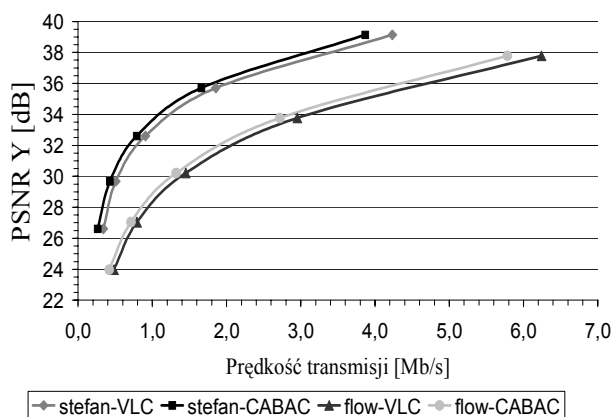
Korzyści płynące z zastosowania algorytmu CABAC można interpretować na dwa sposoby:

- przy zachowaniu takiej samej jakości uzyskuje się zmniejszenie prędkości transmisji,
- przy zachowaniu takiej samej prędkości transmisji uzyskuje się sekwencję wizyjną o lepszej jakości.

Porównanie zależności jakości od prędkości transmisji dla omówionych technik kodowania binarnego przedstawiono na wykresach 4 i 5.



Wykres 4. Zależność PSNR Y od przepływności dla sekwencji basket.4cif i bus.4cif dla algorytmów CABAC i VLC



Wykres 5. Zależność PSNR Y od przepływności dla sekwencji stefan.4cif i flow.4cif dla algorytmów CABAC i VLC

Przy takiej samej prędkości transmisji zastosowanie algorytmu CABAC poprawia wielkość miary PSNR średnio o 0,3 dB.

5. WNIOSKI

Złożoność obliczeniowa własnej implementacji kodera i dekodera w przypadku zastosowania algorytmu CABAC jest większa niż w przypadku użycia algorytmu VLC i znacząco wpływa zarówno na czas kodowania, jak i dekodowania sekwencji wizyjnych. Wzrost złożoności obliczeniowej ściśle zależy od jakości zdekodowanej sekwencji i rośnie wraz ze wzrostem wartości miary PSNR (dla PSNR powyżej 37 dB przekracza 30% czasu kodowania i dekodowania). Należy zaznaczyć, że wzrost wielkości PSNR wiąże się ze wzrostem prędkości transmisji zakodowanej sekwencji.

Zastosowanie algorytmu CABAC pozwala na uzyskanie wyższej efektywności kompresji przy zachowaniu takiej samej jakości. W porównaniu z alternatywnym algorytmem VLC redukcja strumienia binarnego dla testowanych sekwencji wynosi 4 ÷ 23%.

Biorąc pod uwagę ciągły wzrost mocy obliczeniowej procesorów oraz dużą efektywność kompresji oferowaną przez nowy standard H.264/AVC z algorytmem kodowania binarnego CABAC przewiduje się szerokie zastosowanie tego standardu w systemach radiokomunikacji ruchomej nowej generacji oraz telewizji cyfrowej.

LITERATURA

- [1] M. Domański, *Zaawansowane techniki kompresji obrazów i sekwencji wizyjnych*, Poznań 2000.
- [2] W. Skarbek, *Metody reprezentacji obrazów cyfrowych*, Warszawa 1993.
- [3] D. Karwowski, *An efficient Architecture of H.264/AVC CAVLC Decoder*, 11th IWSSIP Poznań 2004, s. 151 - 154.
- [4] ISO/IEC IS 13818-2 / ITU-T Rec. H.262, *Generic coding of moving pictures and associated audio, part 2: video*, 1993.
- [5] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression. Video Coding for Next-generation Multimedia*, Wiley 2003.
- [6] ISO/IEC JTC 1/SC29/WG11/N5555, Text of ISO/IEC FDIS 14496-10 : *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*, 2003.
- [7] D. Marpe, T. Wiegand, *A highly efficient multiplication – free binary arithmetic coder and its application in video coding*, ICIP 2003.
- [8] *Special issue on H.264/AVC video coding standard*, IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, July 2003.
- [9] R. Gallager, D. Van Voorhis, *Optimal source codes for geometrically distributed integer alphabets*, IEEE Trans. Inform. Theory, Vol. 21, s. 228–230, Mar. IT-1975.