Biuletyn WAT
Vol. LVIII, Nr 4, 2009

Zastosowanie adaptacyjnego wielokontekstowego kodera arytmetycznego do bezstratnej kompresji dźwięku

GRZEGORZ ULACHA

Zachodniopomorski Uniwersytet Technologiczny, Wydział Informatyki,
71-210 Szczecin, ul. Żołnierska 49

Streszczenie. W artykule przedstawiono szczegółową analizę zastosowania adaptacyjnego kodera arytmetycznego w bezstratnej kompresji dźwięku. Przedstawiono nowatorskie reguły podziału kontekstowego, dzięki którym dla każdej próbki istnieje możliwość właściwego doboru jednego z 20 rozkładów prawdopodobieństwa pozwalającego na efektywne kodowanie arytmetyczne. Scharakteryzowano sposób adaptacji długookresowej. Zwiększenie tempa adaptacji krótkookresowej osiągnięto przez wprowadzenie odpowiedniej kwantyzacji błędów predykcji. Zaproponowano także osobne kodowanie bitu znaku. Na końcu przedstawiono porównanie efektywności omawianej metody z innymi kodekami znanymi z literatury. Zaimplementowana metoda pozwala na kodowanie dźwięku w czasie rzeczywistym.
Słowa kluczowe: teoria informacji, bezstratna kompresja obrazu, kodowanie kontekstowe, skumulowany błąd predykcji, koder arytmetyczny
Symbole UKD: 681.3.05

1. Wprowadzenie

W przypadku stosowania nowoczesnych metod kompresji wykorzystuje się zwykle dwa etapy: modelowanie danych, a następnie kompresję jedną z wydajnych metod entropijnych, wśród których najefektywniejsze to kodowanie arytmetyczne i kodowanie Huffmana [16] wraz z jego modyfikacjami w postaci kodów Golomba i Rice'a [5].

Nieustanne dążenie do uzyskania coraz większej efektywności bezstratnej kompresji dźwięku prowadzi do opracowywania metod o wzrastającej złożoności implementacyjnej. Okresem największej aktywności projektantów nowych metod były lata 2002-2006, kiedy opracowywano propozycje standardu MPEG4 Lossless Audio Coding [6]. W sposób niezależny rozwija się gałąź niejako amatorskich rozwiązań, których algorytmy nie są w pełni prezentowane w publikacjach naukowych. Przykładowo OptimFrog [23] oraz Monkey's Audio [24] należą do czołówki najwydajniejszych programów do bezstratnej kompresji dźwięku.

2. Modelowanie danych

Podstawowym etapem kodowania bezstratnego jest modelowanie danych, czyli przekształcenie pierwotnej postaci sygnału do nowego ciągu danych, który charakteryzuje się większą podatnością na kodowanie entropijne. Wykorzystuje się tu fakt wzajemnych zależności między próbkami. Należy taką korelację jak najefektywniej usunąć, aby entropia H rzędu zerowego (źródła bez pamięci) była jak najniższa, dzięki czemu istnieje możliwość efektywnej kompresji tych danych przy użyciu prostych w implementacji kodów, takich jak Golomba i Rice'a. Wzór (1) przedstawia funkcję takiej entropii:

$$H = -\sum_{i=e_{\min}}^{e_{\max}} p_i \cdot \log_2 p_i, \tag{1}$$

gdzie p_i jest prawdopodobieństwem wystąpienia symbolu i należącym do zbioru symboli o wartościach z przedziału od $e_{\min}$ do $e_{\max}$. Entropia informuje nas o minimalnej średniej liczbie bitów potrzebnej do zakodowania próbki przy użyciu jednej z metod statycznego kodowania entropijnego. W praktyce wzór (1) jest jedynie pewnym przybliżeniem średniej bitowej. Użycie kontekstowego kodera pozwala uzyskać średnią bitową nawet niższą od entropii rzędu zerowego, co zostanie pokazane w punkcie 4.

Istnieją dwa podstawowe typy modelowania, pierwszy to wykorzystanie predykcji liniowej [4, 15] lub nieliniowej (np. z użyciem sieci neuronowych [11]). Drugim typem jest wykorzystanie transformacji DCT (MPEG-4 SLS [22]) lub falkowych. Najczęściej do modelowania służy typowy predyktor liniowy rzędu r, który jest wartością przewidywaną aktualnie kodowanej próbki na podstawie r poprzednich próbek sygnału. Ma on postać:

$$\hat{x}(n) = \sum_{i=1}^{r} b_i \cdot x(n-i), \tag{2}$$

gdzie elementy $x(n-i)$ są wartościami próbek poprzedzających aktualnie kodowaną $x(n)$, natomiast elementy b_i to współczynniki predykcji [16]. Użycie predyktora liniowego polega na kodowaniu tylko błędów predykcji, czyli różnic $e(n)$ między wartością rzeczywistą a przewidywaną (zaokrągloną do najbliższej liczby całkowitej), które najczęściej są niewielkimi wartościami oscylującymi w pobliżu zera i charakteryzują się rozkładem zbliżonym do rozkładu Laplace'a:

$$e(n) = x(n) - \hat{x}(n). \tag{3}$$

Kluczową rolę odgrywa tu sposób doboru współczynników b_i danego modelu. Mogą być one ustalone na stałe (predyktor stały), statyczne w obrębie jednej kodowanej ramki (zmieniają się po przejściu do kodowania kolejnej ramki), jak i w pełni adaptacyjne (ich zmiana może następować nawet po kodowaniu każdej kolejnej próbki). Najczęściej stosuje się modele statyczne, a współczynniki predykcji wyznacza się w koderze, minimalizując błąd średniokwadratowy lub długość zakodowanej ramki. Metoda ta jest określana mianem adaptacji w przód, gdyż koder musi mieć dostęp do całej ramki przed rozpoczęciem jej kodowania, a to oznacza, że wyznaczone współczynniki powinny być również przesłane dekoderowi. Z tego względu należy opracować efektywną metodę doboru wielkości ramki, rzędu predykcji, a także dokładności zapisu współczynników predykcji. W zaawansowanych metodach (np. MPEG-4 ALS) stosuje się technikę wielokrotnego kodowania z pomiarem długości ramki po zakodowaniu, co wiąże się ze znaczną złożonością obliczeniową całego procesu kodowania. Dekoder działa jednak stosunkowo szybko, bo pobiera jedynie informacje nagłówkowe skojarzone z daną ramką i rozpoczyna dekodowanie w oparciu o wzory (2) i (3).

Wadą kodowania z adaptacją wstecz jest mniejszy poziom wiedzy o najbliższych kodowanych próbkach. Aby uzyskać wysoką efektywność, dokonuje się więc adaptacji współczynników po kodowaniu każdej próbki, co wiąże się z dużą złożonością obliczeniową zarówno po stronie kodera, jak i dekodera (w dekoderze adaptacja jest identyczna). Zaletą jest brak potrzeby przesyłania współczynników predykcji, a to z kolei pozwala na wykorzystywanie modeli predykcyjnych wysokiego rzędu ($r > 100$). Jak do tej pory nie wykazano, która z metod adaptacji umożliwia uzyskanie wyższej wydajności kompresji w przypadku kodowania dźwięku. W przypadku obrazów nieznaczną tylko przewagę uzyskała metoda z adaptacją wstecz, co pokazano w pracy [18], gdzie porównano adaptację wprzód TMW$^{\mathrm{LEGO}}$ [8] z najlepszą metodą bezstratnej kompresji obrazów Multi-WLS [21].

W przypadku kodowania dźwięku wielokanałowego, oprócz czasowej zależności występującej między sąsiednimi próbkami, istnieje także zależność międzykanałowa (ang. *interchannel correlation*) [9].

W tej pracy główny nacisk został położony na rozwinięcie drugiego etapu, czyli omówionego w kolejnym punkcie kodera arytmetycznego. Jako etap modelowania zaproponowano natomiast stały predyktor liniowy, wspólny dla wszystkich utworów testowych. Wykorzystując trzy proste modele predykcyjne zastosowane w koderze SHORTEN [15]:

$$\hat{x}_1(n) = x(n-1), \tag{4}$$

$$\hat{x}_2(n) = 2x(n-1) - x(n-2), \tag{5}$$

$$\hat{x}_3(n) = 3x(n-1) - 3x(n-2) + x(n-3), \tag{6}$$

możemy wyznaczyć ich kombinację liniową:

$$\hat{x}(n) = \left\lfloor \frac{3\hat{x}_1 + 9\hat{x}_2 + 4\hat{x}_3 + 8}{16} \right\rfloor, \tag{7}$$

co po uproszczeniu daje:

$$\hat{x}(n) = \left\lfloor \frac{33x(n-1) - 21x(n-2) + 4x(n-3) + 8}{16} \right\rfloor. \tag{8}$$

Dodatkowo obliczana jest wartość składowej stałej błędu predykcji, która skojarzona jest z odpowiednio wyznaczonym kontekstem. Składową tę usuwa się z sygnału błędów predykcji. Adaptacyjny algorytm wyznaczania kontekstów został zaczerpnięty z technik usuwania składowej stałej z obrazu zaimplementowanych w metodach CALIC [20] oraz JPEG-LS [19]. W proponowanym rozwiązaniu wykorzystuje się 1024 konteksty, a każdy z nich traktowany jest jako zbiór indywidualnych cech sąsiedztwa. Pozwala to na dalsze obniżenie błędu średniokwadratowego oraz średniej bitowej zakodowanego sygnału.

3. Adaptacyjny koder arytmetyczny

Większość metod kodowania błędów predykcji sygnału audio oparta jest o modyfikacje kodów Golomba [5] i Rice'a [13]. Rozbudowaną wersję takiego podejścia zaproponowano w pracy [12]. Aby uzyskać wyższą efektywność, należy zaimplementować adaptacyjną wersję kodera arytmetycznego [10, 16], który dzięki technice przełączania kontekstów pozwala precyzyjniej dopasować aktualnie kodowane próbki do właściwego rozkładu prawdopodobieństwa [4].

Poniżej przedstawiono zasadę działania proponowanego rozwiązania.

3.1. Długookresowa forma adaptacji rozkładu prawdopodobieństwa

Ze względu na fakt kodowania wartości bezwzględnych błędów predykcji (bit znaku koduje się osobno), mamy do czynienia z rozkładem zbliżonym do jednostronnego rozkładu Laplace'a. Początkowy rozkład prawdopodobieństwa, a właściwie wektor liczebności $\mathbf{N}_e$ wystąpień wartości i (poszczególnych wartości bezwzględnych błędów predykcji), możemy potraktować jako jednostajny. Adaptacja długookresowa przejawia się tym, że po wczytaniu i zakodowaniu każdej kolejnej wartości $|e(n)|$ należy uaktualnić wektor liczebności, zwiększając wartość pod indeksem $|e(n)|$ o jeden: $\mathbf{N}_e(|e(n)|) = \mathbf{N}_e(|e(n)|) + 1$. Dodatkowo możemy wprowadzić efekt zapominania, sprzyjający zmniejszaniu tych liczebności, których wartości wśród ostatnio kodowanych danych się nie pojawiły lub pojawiały się rzadziej niż we wcześniejszym etapie kodowania [3]. W tym celu kontrolujemy łączną liczbę zakodowanych do tej pory wartości, a dokładniej wartość licznika, który jest zwiększany o jeden po każdym zakodowaniu liczby $|e(n)|$. Jeśli licznik osiągnie z góry założoną wartość 2^s, wówczas wszystkie elementy wektora liczebności zmniejszane są o połowę:

$$\mathbf{N}_e(i) = \left\lfloor \frac{\mathbf{N}_e(i)}{2} \right\rfloor + 1, \text{ dla każdego } i \text{ od } 0 \text{ do } e_{\max}. \tag{9}$$

Po przeskalowaniu ponownie obliczana jest wartość licznika jako suma wszystkich elementów $\mathbf{N}_e(i)$ wektora liczebności. W proponowanym koderze użyto $s = 12$ dla kontekstów wykorzystywanych przy kodowaniu wartości skwantyzowanych błędów predykcji, $s = 17$ dla kontekstów wykorzystywanych przy kodowaniu błędów kwantyzacji (patrz punkt 3.3), a $s = 6$ dla kontekstów bitu znaku (patrz punkt 3.4).

3.2. Podział kontekstowy w koderze arytmetycznym

Opisany w punkcie 3.1 koder adaptacyjny potrafi dostosować się do rozkładu w sensie długookresowym, lecz można wykorzystać jeszcze istnienie krótkookresowych zależności między kolejno kodowanymi danymi, analizując najbliższe sąsiedztwo złożone z próbek $x(n-i)$ oraz sygnału błędów predykcji $e(n-i)$. Na podstawie tych cech można dość dokładnie określić właściwy typ rozkładu aktualnie kodowanej wartości $|e(n)|$. Wychodząc z takiego założenia, możemy zaprojektować kontekstowy koder arytmetyczny mający nie jeden, lecz t rozkładów prawdopodobieństw skojarzonych z poszczególnymi numerami konteksów od 0 do $t - 1$. Teoretycznie wraz ze wzrostem liczby kontekstów można oczekiwać zwiększenia efektywności kompresji, jednakże pojawia się problem rozrzedzenia kontekstów, czyli zbyt wolnej adaptacji ich rozkładu [10]. Adaptacyjny charakter budowy rozkładów prawdopodobieństwa wymaga szybkiego ustalenia się przybliżonej docelowej postaci każdego z t rozkładów, dlatego należy ustalić pewien kompromis między liczbą kontekstów a szybkością ich adaptacji. W kodowaniu obrazów wykorzystuje się 8 [20], 16, a nawet 20 kontekstów [2]. W proponowanym tu rozwiązaniu wykorzystano 20 kontekstów.

Istotnym czynnikiem wpływającym na efektywność kompresji jest odpowiedni dobór reguły decyzyjnej wyznaczającej numer kontekstu. Wartość ω_1 jest obliczana na podstawie błędów $e(n - i)$ oraz próbek $x(n - i)$ znajdujących się w najbliższym otoczeniu:

$$\begin{aligned}\omega_1 = \max\{&5|e(n-1)|,\ 5|e(n-2)|,\ 3|e(n-3)|,\ 3\ |e(n-4)|,\\ &1{,}5|x(n-1)-x(n-2)|,\ 2|x(n-2)-x(n-3)|\}.\end{aligned} \tag{10}$$

Nieco inne rozwiązania zastosowano w pracach [1, 7]. W proponowanym tu rozwiązaniu, podobnie jak w pracy [7], kolejny parametr ω_2 wyznaczany jest jako średnia wagowa modułów błędów $|e(n - i)|$ znajdujących w najbliższym otoczeniu:

$$\omega_2 = \frac{1}{\delta}\sum_{i=1}^{m}\frac{|e(n-i)|}{\sqrt{i}}, \tag{11}$$

gdzie warość $m = 26$, a δ jest wyznaczane ze wzoru:

$$\delta = \sum_{i=1}^{m}\frac{1}{\sqrt{i}}. \tag{12}$$

Na podstawie parametrów ω_1 oraz ω_2 możemy wyznaczyć końcową wartość ω:

$$\omega = \frac{44}{25+\vartheta}\max\{0{,}037\omega_1, \omega_2\}, \tag{13}$$

przy czym wartość ϑ jest pierwiastkiem z odchylenia standardowego całego sygnału zaokrąglonym do najbliższej 8-bitowej liczby całkowitej:

$$\vartheta = \left\lfloor \sqrt[4]{\frac{1}{N}\sum_{i=1}^{N}x^2(i) - \left(\frac{1}{N}\sum_{i=1}^{N}x(i)\right)^2} + \frac{1}{2}\right\rfloor, \tag{14}$$

gdzie N oznacza liczbę próbek sygnału. Wartość ω poddajemy kwantyzacji z użyciem $t - 1$ progów $\mathbf{T}_h(j)$, aby uzyskać numer kontekstu arytmetycznego wskazującego na aktualny rozkład prawdopodobieństwa. Przykładowo dla $t = 20$ progi można ustalić następująco: $\mathbf{T}_h = \{3, 6, 11, 17, 23, 30, 38, 46, 55, 67, 82, 98, 115, 135, 155, 175, 195, 225, 255\}$.

3.3. Kwantyzacja błędów predykcji

Najczęściej w technikach bezstratnego kodowania dźwięku stosuje się kod Rice'a, będący podzbiorem kodów Golomba. Czasami wykorzystuje się różne modyfikacje kodu Rice'a, w tym kod Eliasa γ' [2, 17], który jest czasem określany jako eksponencjalny kod Golomba (ang. *Exponential Golomb code*) [14]. Aby uogólnić sposób podejścia do takiego kodu, wystarczy posłużyć się zasadą kwantyzacji skalarnej. W proponowanym rozwiązaniu, aby zwiększyć szybkość adaptacji rozkładów prawdopodobieństwa skojarzonych z poszczególnymi kontekstami, stosuje się kwantyzację wartości $|e(n)|$. Polega to na zrzutowaniu przedziału liczb $|e(n)|$ od 0 do 65535 na mniejszy zakres liczb k, np. od 0 do 24. Idea taka jest stosowana w wielu metodach kodowania, np. w standardzie JPEG [16]. Technika kwantyzacji z bezstratnym kodowaniem ma za zadanie podzielić wartość $|e(n)|$ na dwie liczby, które są kodowane z użyciem odrębnych rozkładów (czasem przyjmuje się, iż błąd kwantyzacji zapisywany jest jako ciąg bitów bez kompresji). Poniżej przedstawiono etapy kwantyzacji i kodowania.

Na początku wyznacza się wartość k z zależności $\mathbf{T}(k) \le |e(n)| < \mathbf{T}(k+1)$, mając dany zbiór kolejnych progów kwantyzacji $\mathbf{T}$ = {0, 1, 2, 4, 8, 12, 16, 32, 48, 64, 96, 128, 192, 256, 384, 512, 768, 1024, 1536, 2048, 3072, 4096, 8192, 16384, 32768}. Skwantyzowana wartość oznaczona jako k jest wysyłana do kodera arytmetycznego i kodowana po skojarzeniu jej z odpowiednim rozkładem prawdopodobieństwa przypisanym do kontekstu wyznaczonego według zasady opisanej w punkcie 3.2. Błąd kwantyzacji $e_q = |e(n)| - \mathbf{T}(k)$ jest jest traktowany jako $\mathbf{q}(k)$ bitowa liczba, gdzie $\mathbf{q}(k)$ odczytujemy jako k-tą wartość z wektora liczb $\mathbf{q}$ = {0, 0, 1, 2, 2, 2, 4, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 12, 13, 14, 15}. Jeśli $\mathbf{q}(k) > 0$, wartość e_q może być wysłana na wyjście w postaci $\mathbf{q}(k)$ bitów lub kodowana za pomocą jednego z 15 koderów arytmetycznych (o numerze $\mathbf{q}(k)$), przy czym 12 pierwszych jest adaptacyjnych, a trzy ostatnie mają na stałe ustawiony rozkład prawdopodobieństwa.

Dekodowanie polega na odczytaniu wartości k będącej indeksem do wektora wartości progowych $\mathbf{T}$ oraz do wektora liczby bitów kwantyzacji $\mathbf{q}$. Jeśli $\mathbf{q}(k) > 0$, to odczytywana jest z dekodera (o numerze $\mathbf{q}(k)$) wartość błędu kwantyzacji e_q, w przeciwnym przypadku $e_q = 0$. Następnie wyznaczana jest wartość $|e(n)|$ z zależności: $|e(n)| = \mathbf{T}(k) + e_q$. Przykładowo niech $|e(n)| = 29$. Wówczas $\mathbf{T}(6) = 16 \le 29 < \mathbf{T}(7) = 32$, czyli $k = 6$, $\mathbf{T}(6) = 16$, $e_q = 29 - 16 = 13$, $\mathbf{q}(6) = 4$, zatem liczbę 13 kodujemy, używając czwartego kodera arytmetycznego błędów kwantyzacji (kodującego czterobitowe liczby od 0 do 15). W dekoderze $|e(n)| = \mathbf{T}(6) + e_q = 16 + 13 = 29$.

3.4. Kodowanie bitu znaku

Ze względu na symetryczność rozkładu prawdopodobieństwa błędów predykcji wygodniej kodować jest ich wartości bezwzględne $|e(n)|$, co pozwala na szybszą adaptację rozkładów w poszczególnych kontekstach kodera arytmetycznego. Osobno kodowany jest bit znaku, przy czym informację o nim trzeba zapisywać jedynie dla niezerowych wartości $e(n)$. W niektórych algorytmach nie przewiduje się specjalnego sposobu kodowania bitu znaku, zapisując do pliku wynikowego wartość tego bitu bez zmian. W tej pracy zaproponowano użycie adaptacyjnego kodera arytmetycznego z podziałem na 128 kontekstów. Rozkład tego dwusymbolowego źródła jest początkowo ustawiany jako jednostajny w każdym z kontekstów (liczebność wystąpień 0 i 1 inicjalizowana jest wartością 1).

Wartość kontekstu wyznaczana jest jako liczba składająca się z 7 bitów, cztery pierwsze są bitami znaków błędów predykcji: $\mathrm{sgn}(e(n-i))$, gdzie $i = \{1, 2, 3, 4\}$. Piąty bit kontekstu jest bitem znaku poprzedniej próbki: $\mathrm{sgn}(x(n-1))$. Bity 6 i 7 ustawiane są na 1, gdy spełniona jest nierówność $|e(n-i)| > \gamma$, przy czym dla bitu 6 $i = 1$, $\gamma = 300$, a dla bitu 7 $i = 2$, $\gamma = 400$.

4. Analiza efektywności

W tabeli 1 zaprezentowano wyniki średnich bitowych rozumianych jako przeciętna liczba bitów potrzebna do zakodowania jednej próbki dźwięku. Do celów testowych wykorzystano zestaw 16 kilkudziesięciosekundowych fragmentów dźwiękowych różnych gatunków (2 kanały, próbkowanie 44,1 kHz, próbki 16-bitowe) [25]. Kolumna Pomiar 1 zawiera średnie bitowe po kodowaniu arytmetycznym bez użycia modelowania i podziału kontekstowego. W kolumnie Pomiar 2 umieszczono średnie bitowe po kodowaniu arytmetycznym bez podziału kontekstowego, ale po zastosowaniu modelowania opisanego wzorem (8). W kolejnych dwóch kolumnach znajdują się wyniki po uaktywnieniu przełączania kontekstowego dla wartości $|e(n)|$ (Pomiar 3) oraz ostateczna postać z aktywnym także przełączaniem kontekstów kodera bitu znaku błędów predycji (Pomiar 4). W ostatniej kolumnie przedstawiono wartość entropii sygnału po modelowaniu. Analizując uśrednione wyniki (ostatni wiersz tabeli 1), można wysnuć wniosek, że bardzo istotnym etapem jest modelowanie danych, jednakże wykorzystanie przełączania kontekstowego w koderze arytmetycznym również znacząco obniża średnią bitową, co można zauważyć w odniesieniu do wartości entropii (średnia jest niższa od entropii o 6,6%).

TABELA 1

Wyniki średnich bitowych oraz entropii dla zestawu 16 plików testowych

Utwory	Pomiar 1	Pomiar 2	Pomiar 3	Pomiar 4	Entropia
ATrain	11,55304	8,62992	8,33097	8,26073	8,79122
BeautySlept	11,83761	10,80190	10,58493	10,45690	10,97278

cd. tabeli 1

chanchan	13,90031	10,61353	10,34745	10,28227	10,78340
death2	14,23712	7,48334	7,12370	7,07529	7,90609
experiencia	14,60721	12,31254	12,06639	11,99347	12,44488
female_speech	11,56793	7,70016	7,36688	7,29220	8,37170
FloorEssence	14,29714	11,30584	11,03681	10,90544	11,66568
ItCouldBeSweet	14,44217	11,49935	11,28387	11,18609	11,71721
Layla	13,38915	10,80733	10,56382	10,48871	11,12147
LifeShatters	14,73687	11,74883	11,68596	11,57747	11,82800
macabre	13,54923	10,28724	10,23401	10,15679	10,56189
male_speech	11,78120	7,62313	7,30811	7,25169	8,05166
SinceAlways	14,44945	12,16756	11,93918	11,83172	12,48493
thear1	14,93391	12,15004	12,09523	12,00401	12,38926
TomsDiner	11,85473	9,66682	9,41472	9,33476	10,09635
velvet	13,06027	12,08197	11,36035	11,16748	12,16598
Średnia	13,38733	10,42997	10,17140	10,07906	10,79053

W tabeli 2 zaprezentowano porównanie efektywności wybranych kodeków sygnału audio na podstawie [25]. Testy przeprowadzono sumarycznie dla tego samego zestawu nagrań, które wykorzystano w poprzednim pomiarze. W przypadku każdego programu dobrane zostały parametry dające najkrótsze pliki wynikowe. Mimo bardzo prostego modelu predykcyjnego proponowane tu rozwiązanie AWKA (adaptacyjny wielokontekstowy koder arytmetyczny) okazało się wydajniejsze od popularnego archiwizera RAR 3.70, jak i zaprojektowanych specjalnie do kodowania sygnału audio programów SHORTEN i WaveZIP. Jednak w porównaniu z najwydajniejszymi obecnie programami Monkey's Audio 4.01 i OptimFROG 4.600ex (o efektywności zbliżonej do MPEG4 Lossless Audio) rezultat okazał się gorszy o około 18%.

Tabela 2

Porównanie efektywności wybranych kodeków sygnału audio

Nazwa programu	Wersja	Parametry	Rozmiar [B]
OptimFROG	4.600ex	-maximumcompression -experimental -seek min	27.051.627
Monkey's Audio	4.01	-c5000	27.980.632
FLAC	1.1.4	-8 -no-padding -b 4096 -l 32 -r 16 -lax	29.707.629

cd. tabeli 2

AWKA	1.0	–	32.097.628
Shorten	2.3b	-b 128	33.262.548
WaveZIP	2.01	–	33.999.250
RAR (LZ77)	3.70 beta 5	-m5 -mdg -mct- -mc4a+	34.468.346

5. Podsumowanie

Biorąc pod uwagę stały wzrost mocy obliczeniowej i pojemności pamięci w systemach komputerowych, należy dążyć do dalszego udoskonalania algorytmów. Zaproponowany w tym artykule kontekstowy koder arytmetyczny jeszcze kilka lat wcześniej nie był brany pod uwagę przy projektowaniu standardu MPEG4, gdzie starano się wykorzystać proste rozwiązania oparte o kody Golomba i Rice'a. W obecnej formie implementacja proponowanego kodera w języku C bez optymalizacji pozwala na kodowanie i dekodowanie w czasie rzeczywistym utworów stereofonicznych nawet o częstotliwości próbkowania 96 kHz przy zajętości ok. 90% mocy obliczeniowej procesora Pentium4 2,8 GHz. Uwzględniając dalszy wzrost mocy obliczeniowej, możliwość zrównoleglenia oraz optymalizację kodu, można stwierdzić, że propozycja używania adaptacyjnego kodowania arytmetycznego z przełączaniem kontekstów ma realne szanse zastąpić dotychczas stosowane rozwiązania.

Dalszy etap badań ukierunkowany zostanie na rozwój metod modelowania, w tym usunięcie zależności międzykanałowych, co pozwoli na znaczący wzrost efektywności kompresji.

Artykuł wpłynął do redakcji 15.05.2009 r. Zweryfikowaną wersję po recenzji otrzymano w maju 2009 r.

LITERATURA

[1] B. Aiazzi, L. Alparone, S. Baronti, *Context modeling for near-lossless image coding*, IEEE Signal Processing Letters, vol. 9, no. 3, March 2002, 77-80.

[2] G. Deng, H. Ye, *Lossless image compression using adaptive predictor combination, symbol mapping and context filtering*, Proceedings of IEEE 1999 International Conference on Image Processing, Kobe, Japan, vol. 4, Oct. 1999, 63-67.

[3] R. G. Gallager, *Variations on a theme by Huffman*, IEEE Transactions on Information Theory, vol. 24, no. 6, November 1978, 668-674.

[4] C. D. Giurcaneau, I. Tabus, J. Astola, *Adaptive context based sequential prediction for lossless audio compression*, Proceedings of IX European Signal Processing Conference EUSIPCO 1998, Rhodes, Greece, vol. 4, Sept. 1998, 2349-2352.

[5] S. W. Golomb, *Run-length encoding*, IEEE Transactions on Information Theory, vol. 12, July 1966, 399-401.

[6] H. Huang, P. Fränti, D. Huang, S. Rahardja, *Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding*, IEEE Trans. on Audio, Speech and Language Processing, vol. 16, no. 3, March 2008, 554-562.

[7] I. Matsuda, N. Ozaki, Y. Umezu, S. Itoh, *Lossless coding using Variable Blok-Size adaptive prediction optimized for each image*, Proceedings of 13th European Signal Processing Conference EUSIPCO-05 CD, September 2005.

[8] B. Meyer, P. Tischer, *TMW^{Lego} — An Object Oriented Image Modelling Framework*, Proceedings of Data Compression Conference 2001, 504.

[9] T. Moriya, D. Yang, T. Liebchen, *Extended Linear Prediction Tools for Lossless Audio Coding*, Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04), Montreal, Quebec, Canada, vol. 3, 17-21 May 2004, III_1008-1011.

[10] A. Przelaskowski, *Kompresja danych: podstawy, metody bezstratne, kodery obrazów*, Wydawnictwo BTC, Warszawa, 2005.

[11] E. Ravelli, P. Gournay, R. Lefebvre, *A Two-Stage MLP+NLMS Lossless coder for stereo audio*, Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06), Toulouse, France, vol. 5, 14-19 May 2006, V_177-180.

[12] Y. A. Reznik, *Coding of prediction residual in MPEG-4 standard for lossless audio coding (MPEG-4 ALS)*, Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04), Montreal, Quebec, Canada, vol. 3, 17-21 May 2004, III_1024-1027.

[13] R. F. Rice, *Some practical universal noiseless coding techniques*, Jet Propulsion Labolatory, JPL Publication 79-22, Pasadena, CA, March 1979.

[14] I. E. G. Richardson, *H.264 and MPEG4 video compression. Video coding for next-generation multimedia*, West Sussex, England, John Wiley & Sons Ltd., 2004.

[15] T. Robinson, *SHORTEN: Simple lossless and near-lossless waveform compression*, Cambridge Univ. Eng. Dept., Cambridge, UK, Tech. Rep., 156, 1994, 1-17.

[16] K. Sayood, *Introduction to Data Compression*, 2nd edition, Morgan Kaufmann Publ., 2002.

[17] K. Sayood (ed.), *Lossless compression handbook*, California, Academic Press USA, 2003.

[18] G. Ulacha, R. Stasiński, *A Time-Effective Lossless Coder Based on Hierarchical Contexts and Adaptive Predictors*, Proceedings of 14th IEEE Mediterranean Electrotechnical Conference MELECON'08, Ajaccio, France, 5-7 May 2008, 829-834.

[19] M. J. Weinberger, G. Seroussi, G. Sapiro, *LOCO-I: Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS*, IEEE Trans. on Image Processing, vol. 9, no. 8, August 2000, 1309-1324.

[20] X. Wu, N. D. Memon, *CALIC — A Context Based Adaptive Lossless Image Coding Scheme*, IEEE Trans. on Communications, vol. 45, May 1996, 437-444.

[21] H. Ye, *A study on lossless compression of greyscale images*, PhD thesis, Department of Electronic Engineering, La Trobe University, October 2002.

[22] R. Yu, S. Rahardja, C. C. Ko, H. Huang, *Improving coding efficiency for MPEG-4 Audio Scalable Lossless coding*, Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA, vol. 3, 18-23 March 2005, III_169-172.

[23] http://www.losslessaudio.org/

[24] http://www.monkeysaudio.com/

[25] http://uclc.info/lossless_audio_compression_test.htm

G. ULACHA

Application of an adaptive multi-contextual arithmetic encoder for lossless audio compression

Abstract. The paper presents a detailed analysis of application of an adaptive arithmetic encoder in lossless audio compression. Novel rules for contextual split are provided thanks to which there exists the possibility of proper selection for each sample, one from 20 probability distributions aiming at effective arithmetic encoding. A long-term adaptation technique is characterized. An increase in the short-term adaptation speed is obtained due to the introduction of an appropriate prediction error quantization. The separate encoding of the sign bit is also proposed. Eventually, there is a comparison of the proposed method effectiveness with other encoders known from literature. The implemented method allows us to encode audio in the real-time.

Keywords: information theory, lossless image compression, context coding, cumulated predictor error, arithmetic coder

Universal Decimal Classification: 681.3.05