

ROZDZIAŁ 15. NOWE STANDARDY I ALGORYTMY

Najstarszymi (1980 rok) i szeroko stosowanymi obecnie standardami kompresji obrazów cyfrowych są międzynarodowe standardy kodowania cyfrowych faksów, odpisów (ang. facsimile) Grupy 3 i Grupy 4 opracowane przez grupę konsultacyjną CCITT (*Consultative Committee of the International Telephone and Telegraph*). Standardy te dotyczą jedynie binarnych obrazów zawierających teksty i dokumenty. Wzrost liczby zastosowań wielopoziomowych obrazów cyfrowych w szerokiej gamie zastosowań nieuchronnie prowadzi do opracowywania nowych standardów kompresji. Udogodnienia związane z wprowadzeniem tych standardów dotyczą nie tylko łatwiejszej wymiany obrazów pomiędzy różnymi systemami i aplikacjami, lecz także pozwalają na znaczące ograniczenie kosztu budowy wyspecjalizowanych urządzeń cyfrowych niezbędnych w wielu systemach kompresji obrazów w czasie rzeczywistym.

15.1. Przegląd standardów.

W ostatnich latach prace nad nowymi standardami kompresji obrazów prowadzone były w trzech zasadniczych kierunkach:

- **obrazy binarne**

W 1988 roku został uformowany komitet znany jako JBIG (*Joint Bilevel Imaging Group*) pod auspicjami ISO-IEC/JTC1/SC2/WG8 i CCITT SG VIII NIC, w celu opracowania standardu kompresji i dekompresji obrazów binarnych. Grupa skoncentrowała swej wysiłki na poszukiwaniu efektywniejszego algorytmu od opracowanych wcześniej przez CCITT, w zastosowaniu do klasycznych aplikacji (n.p. ośmiu binarnych obrazów odniesienia zaproponowanych przez CCITT), a także rozszerzenia ich stosowalności do nowych aplikacji. Chodziło głównie o opracowanie algorytmów progresywnych i adaptacyjnych.

Należy też wspomnieć o opracowanej przez naukowców z IBM w 1988 roku binarnej wersji kodera arytmetycznego o nazwie Q-koder i zastosowaniu go do kompresji obrazów binarnych - technika ABIC (*Arithmetic Binary Image Compression*) [4, rozdz. 14.2].

- CCITT Group 3 i 4 - kodowanie długości sekwencji do kompresji obrazów binarnych
- JBIG (Joint Bilevel Imaging Group) - kodowanie arytmetyczne do kompresji danych binarnych

- **pojedyncze obrazy wielopoziomowe, monochromatyczne i kolorowe**

Komitet znany powszechnie pod nazwą JPEG (*Joint Photographic Experts Group*), działający jako ISO-IEC/JTC1/SC2/WG10 przy bliskiej nieformalnej współpracy z CCITT SG VIII NIC, uformowany został pod koniec 1986 roku w celu opracowania międzynarodowego standardu dla pojedynczych, wielopoziomowych, monochromatycznych i kolorowych obrazów. Zadaniem zespołu było zdefiniowanie standardu dla tak różnych zastosowań jak foto- i telegazeta, grafika komputerowa, skład komputerowy, mała poligrafia, kolorowe faksy, systemy medyczne i wiele innych. Pomimo tego, iż w tej dziedzinie nie istniały wcześniejsze standardy, członkowie JPEG byli silnie przekonani, że wymagania zdecydowanej większości tych zastosowań winny być uwzględnione w standardzie. Ostateczna propozycja standardu, która została opublikowana w 1992 jako draft standardu międzynarodowego ISO/IEC [41],[23] zawiera trzy główne składniki: 1) system podstawowy, który zawiera prosty i efektywny algorytm, adekwatny w stosunku do większości zastosowań

kompresji obrazów., 2) zbiór rozszerzeń systemu zawierający przede wszystkim algorytm progresywnego kodowania rozszerzający pole zastosowań, 3) niezależna bezstratna metoda kodowania dla zastosowań wymagających tego typu kompresji. Bardziej dokładne omówienie tego standardu, ze względu na przewidywane przez twórców jego zastosowanie także w systemach obrazowania medycznego, zostanie przedstawione w następnej części tego podrozdziału.

- sekwencja obrazów wielopoziomowych

W 1990 roku, po ponad pięciu latach intensywnych rozważań, opracowany został dokument CCITT Recommendation H.261, *Video Codec for Audiovisual Services at px64 kbits/s*, który jest standardem obejmującym algorytm kodowania dla video telefonii i video konferencji w zakresie szybkości kodowania od 64 do 1920 kbitów/s [25,55]. W nieco zmodyfikowanej wersji rekomendacja została także przyjęta w Ameryce Północnej [56].

Od 1988 roku pracuje grupa znana jako MPEG (*Moving Picture Experts Group*), obecnie jako część ISO-IEC/JTC1/SC2/WG11 w celu zdefiniowania standardu dla gromadzenia i odtwarzania ruchomych obrazów i dźwięku na różnych nośnikach cyfrowych z szybkością 1-1.5 mbitów/s. W 1992 roku ostatecznie został przyjęty standard, powszechnie znany jako MPEG-1, dotyczący głównie aplikacji z szybkością transmisji około 1.2 mbitów/s (prędkość CD-ROM) [57,58]. Może być stosowany w interakcyjnych urządzeniach multimedialnych o jakości obrazu video porównywalnej z VCR, a jakości audio porównywalnej z odtwarzaczami CD (kompresja cyfrowego sygnału audio z prędkościami 64, 128 i 192 kbitów/s) [24].

Z tego standardu wynikają dwie bardzo ważne konsekwencje: obrazy ruchome video przybrały formę danych komputerowych, tzn. danych integrujących tekst i grafikę, a ponadto ruchome video i związany z nim dźwięk może być przekazywany przez istniejące sieci komputerowe i telekomunikacyjne.

W 1994 roku zakończono prace nad draftem standardu MPEG-2 dotyczącym telewizji interakcyjnej oraz telewizji wysokiej rozdzielczości HDTV. Trwają obecnie prace nad przygotowaniem nowego standardu MPEG-4 związanego z aplikacjami o bardzo niskiej szybkości transmisji, rzędu pojedynczych kilo-bitów i mniej, takimi jak video telefony, dostęp do video baz, przekazywanie video wiadomości, zdalnego sterowania i innych. Prace mają być finalizowane już pod koniec 1994 roku [58,59].

15.2. Standard JPEG

Specyfikacja normy standardu JPEG zawiera:

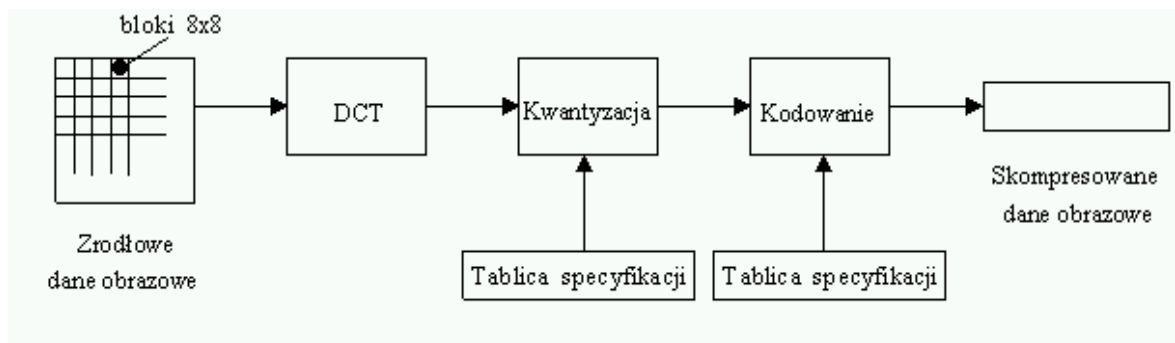
- opis procesu przetwarzania źródłowych danych obrazowych w dane obrazowe skompresowane;
- opis procesu przetwarzania skompresowanych danych obrazowych w zrekonstruowane dane obrazu;
- wskazania dotyczące praktycznych implementacji standardu;
- opis zakodowanej reprezentacji skompresowanych danych obrazowych.

Specyfikacja nie opisuje kompletnej zakodowanej reprezentacji obrazu, może ona zawierać pewne parametry zależne od aplikacji. W normie znajdują się cztery techniki kompresji, a mianowicie:

1. Podstawowy proces kodowania (ang. baseline process).

2. Rozszerzony , bazujący na DCT proces kodowania (ang. extended DCT- based process).
3. Bezstratny proces kodowania (ang. lossless process).
4. Hierarchiczny proces kodowania (ang. hierarchical process).

Kodowanie każdego składnika obrazu (luminancja, składowe chrominancji) przebiega analogicznie. W podstawowym procesie kodowania dane wejściowe muszą być ośmiobitowe. Obraz jest dzielony na bloki 8x8 i każdy blok jest transformowany za pomocą DCT, przy czym kodowanie obrazu przebiega sekwencyjnie, tzn. z lewej strony na prawą zaczynając od góry obrazu i przemieszczając się na dół. Obliczone 64 współczynniki z każdego bloku podlegają kwantyzacji. Każdy ze współczynników jest dzielony przez odpowiadającą mu wartość w tablicy kwantyzacji. Tablicę tę można dobrać w zależności od aplikacji. Współczynniki są następnie ustawiane w jednowymiarowy ciąg danych według sekwencji zygzak. Tak uszeregowane dane z poszczególnych bloków są następnie kodowane, przy czym składową stałą koduje się różnicowo, tzn. koduje się jedynie różnicę pomiędzy wartością składowej stałej obecnego bloku i poprzedniego. Do kodowania używa się kodu Huffmana, przy czym tablica Huffmana nie jest objęta normą.



Rys.15.1. Schemat blokowy algorytmu kompresji wykorzystującego DCT.

Proces dekompresji przebiega dokładnie odwrotnie, przy czym dekodery musi posługiwać się dokładnie tymi samymi tablicami specyfikacji (tablica kwantyzacji, Huffmana). Umożliwia to format zapisu, w którym tablice specyfikacji poprzedzone odpowiednimi markerami umieszczone są w pliku razem z danymi skompresowanymi. Dokładny opis formatu znajduje się w normie.

Rozszerzony proces kompresji umożliwia kompresję zarówno 8- bitowych jak i 12- bitowych danych, przy czym kodowanie może być nie tylko sekwencyjne, ale i progresywne. W modzie progresywnym poszczególne bloki współczynników są kodowane w tej samej kolejności, ale w wielu skanach dzielących współczynniki każdego bloku na poszczególne pasma. Związane jest to jednak z zapewnieniem dodatkowego bufora pamięci do przechowania wartości współczynników z całego obrazu po fazie kwantyzacji, a przed statystycznym kodowaniem tych wartości. Są one następnie kolejno kodowane w skanach zbierających wartości w określonym paśmie ze wszystkich bloków.

W normie występują dwa rodzaje procedur progresywnego kodowania. Pierwsza, nazywana selekcją widma, dzieli współczynniki ustawione według sekwencji zygzak na kolejne pasma, które zawierają poszczególne części częstotliwościowego spektrum każdego z bloków. Druga procedura związana jest z precyzją, z jaką kodowane są współczynniki w każdym z pasm i nazywana jest sukcesywną aproksymacją. Najpierw kodowana jest pewna liczba bardziej znaczących bitów wartości tych współczynników, a następnie mniej znaczące bity. Można progresywnie kodować współczynniki jedynie przy pomocy procedury selekcji

widma, jak też z wykorzystaniem obu procedur. Wówczas mamy do czynienia z tzw. pełną progresją. Ze stwierżeń zawartych w opisie normy wynika, że zastosowanie selekcji widmowej, jakkolwiek wygodne dla wielu zastosowań, daje porównywalne bądź nieco gorsze wyniki kompresji niż sekwencyjna metoda kodowania, podczas gdy przy pełnej progresji skuteczność kompresji może się okazać nieco większa.

W rozszerzonym procesie kodowania możliwe jest także arytmetyczne kodowanie, a tablica warunków (ang. conditioning table) jest wówczas zapamiętywana jako tablica specyfikacji.

Proces kodowania bezstratnego ujęty w standardzie nie wykorzystuje DCT, lecz polega na predykcyjnym kodowaniu wartości pikseli na podstawie trzech wartości pikseli sąsiednich. Dla danego skanu można wybrać jedną z ośmiu predykcji przedstawionych na rys. 15.2. Następnie oblicza się różnicę pomiędzy wartością przewidywaną P_x , a rzeczywistą wartością pikseli i koduje się otrzymaną różnicę odpowiednio dostosowaną metodą Huffmana lub arytmetycznie. W metodzie Huffmana stosuje się 17 kategorii różnicowych wartości, a w kodowaniu arytmetycznym buduje się dwuwymiarowy model statystyczny.

					<u>Numer</u>	<u>Predykcja</u>
					0	Bez predykcji
		c	b		1	$P_x=R_a$
		a	x		2	$P_x=R_b$
					3	$P_x=R_c$
					4	$P_x=R_a+R_b-R_c$
					5	$P_x=R_a+((R_b-R_c)/2)$
					6	$P_x=R_b+((R_a-R_c)/2)$
					7	$P_x=(R_a+R_b)/2$

Rys.15.2. Przestrzenna relacja pomiędzy pikselami (a,b,c), których wartości służą do wyznaczania wartości przewidywanej w pikselu x. Obok rodzaje predykcji dla bezstratnego kodowania.

Predykcja 0 może być używana dla różnicowego kodowania w modzie hierarchicznym, 1,2 i 3 to predykcje jednowymiarowe wykorzystywane głównie na skraju obrazów, podczas gdy predykcje 4-7 są dwuwymiarowe. Można tą metodą odwracalnie kodować od 2 do 16 bitowe wartości pikseli obrazu. Można więc go z łatwością implementować do kompresji szerokiej klasy obrazów.

Hierarchiczny proces kodowania polega na kompresji obrazu poprzez zakodowanie sekwencji kadrów tego obrazu (różnicowo lub nieróżnicowo) o różnej rozdzielczości za pomocą rozszerzonego procesu kodowania (wykorzystującego DCT) lub metody bezstratnej. Można także połączyć te dwie metody i po kodowaniu opartym na DCT w ostatecznej fazie zastosować kodowanie bezstratne (bez fazy kwantyzacji). Wówczas zrekonstruowany obraz różni się od oryginalnego jedynie wskutek błędów przybliżeń wartości DCT i IDCT .

Tworzy się zrekonstruowane składniki odniesienia obrazu różnej rozdzielczości i koduje się różnice pomiędzy obrazem oryginalnym a tymi składnikami odniesienia. Stosowanie filtrów próbkujących obraz oryginalny z różną rozdzielczością tworzy charakterystyczną piramidę przestrzennej rozdzielczości.

Mod hierarchiczny może być stosowany alternatywnie, aby zwiększyć jakość zrekonstruowanych składników obrazu o danej rozdzielczości. Oferuje progresywną

prezentację czy transmisję, ale jest najczęściej użyteczna jedynie w systemach posługujących się wielorodzicielskimi wersjami danych obrazów.

JPEG-LS

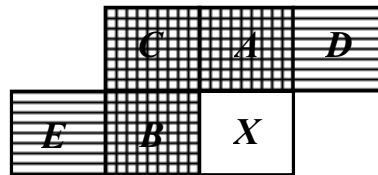
Najbardziej charakterystyczne dla standardu JPEG-LS są rozwiązania schematu predykcji oraz modelowania kontekstu w binarnym kodowaniu reszt predykcyjnych. Są one następujące:

1. Predykcja

Model predykcji jest nieliniowy, określony przez zależność:

$$\hat{x}_{i+1} = \begin{cases} \min(A, B) & \text{if } c \geq \max(A, B) \\ \max(A, B) & \text{if } c \leq \min(A, B) \\ A + B - C & \text{wpp} \end{cases} \quad (15.1)$$

Kontekst powyższego modelu predykcji, podobnie jak kodowania pokazany jest na rys. 15.3



Rys. 15.3. Kontekst w predykcji i modelowaniu kontekstu koderza binarnego; poziome kreski oznaczają kontekst dla kodowania, pionowe dla predykcji.

2. Modelowanie kontekstu

W kodowaniu używany jest szerszy kontekst, jak na rys. 15.3 i wyznaczane są następujące wartości różnicowe:

$$g_1 = D - A, \quad g_2 = A - C, \quad g_3 = C - B, \quad g_4 = B - E, \quad (15.2)$$

przy czym alfabet każdego z tych elementów jest redukowany poprzez kwantyzator równomierny do zbioru symboli $\{-T, \dots, -1, 0, 1, \dots, T\}$,

Model probabilistyczny konstruowany jest na bazie prawdopodobieństw warunkowych postaci:

$$P(\hat{x}_{i+1} | g_1, g_2, g_3, g_4). \quad (15.3)$$

W kolejnych wersjach JPEG-LS wybrano w koderze kontekst rzędu 3, czyli model:

$$P(\hat{x}_{i+1} | g_1, g_2, g_3), \quad (15.4)$$

a wartość T równą cztery.

3. Kodowanie Huffmana z prawdopodobieństwami warunkowymi lub tworzenie kodu Golomb

15.3. Standard MPEG

Od 1988 roku nad standardem kompresji sekwencji obrazów pracuje grupa znana jako MPEG (*Moving Picture Experts Group*). W ramach ISO-IEC/JTC1/SC2/WG11 opracowała ona standard dla gromadzenia i odtwarzania ruchomych obrazów i dźwięku na różnych nośnikach cyfrowych z szybkością 1-1.5 mbitów/s. W 1992 roku ostatecznie został przyjęty standard, znany jako MPEG-1 (dotyczący głównie aplikacji dla transmisji prowadzonych z szybkością około 1,2 megabitów/s - prędkość CD-ROM). W roku 1994 zakończono prace nad projektem standardu MPEG-2 dotyczącym telewizji interaktywnej oraz telewizji wysokiej rozdzielczości HDTV, nieco później opracowano standard MPEG-4 dla silnej kompresji obrazów (bardzo małe średnie bitowe) w wideotelefonii, baz danych obrazów wideo itp. Opracowując normy MPEG poszukiwano kompromisu pomiędzy efektywnością algorytmu (tak aby można było osiągać dla sekwencji wyższe stopnie kompresji, niż w kompresji każdego obrazu z osobna) a swobodą dostępu do dowolnej ramki w sekwencji (najprościej jest w przypadku oddzielnej redukcji danych w każdym z obrazów w sekwencji). W algorytmie MPEG niektóre obrazy w sekwencji kodowane są jednoobrazowo, inne zaś międzyobrazowo, przy czym występują dwa rodzaje kodowania międzyobrazowego: predykcyjny i interpolacyjny. Algorytm MPEG kompresji sekwencji obrazów opiera się na dwóch podstawowych technikach:

- blokowej kompensacji ruchu dla redukcji nadmiarowości czasowych,
- opartym na transformacie kosinusowej (DCT) schemacie stratnej kompresji dla redukcji nadmiarowości przestrzennych (wewnątrzobrazowych).

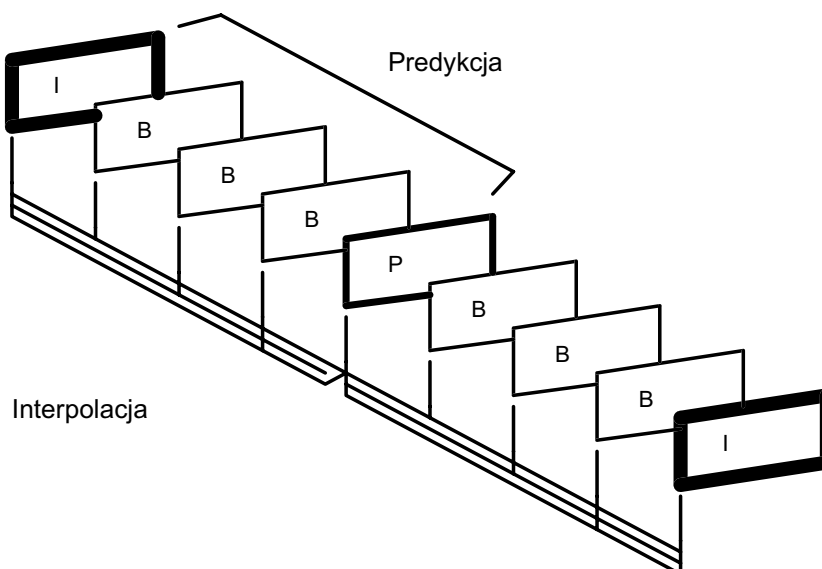
Techniki kompensacji ruchu są realizowane za pomocą modelu predykatora przyczynowego (kodowanie predyktywne) lub nieprzyczynowego (kodowanie interpolacyjne), przy czym określa się postać estymatorów ruchu dla bloków 16×16 . Parametry tych estymatorów są transmitowane (lub zapisywane) razem z informacją przestrzenną (po zakodowaniu przy pomocy metod o zmiennej długości kodu). Błędy predykcji są następnie kompresowane z wykorzystaniem techniki opartej na DCT.

Redukcja nadmiarowości międzyobrazowych

W sekwencji MPEG występują trzy typy kadrów:

- Obrazy kodowane jednoobrazowo (I - *intraframes*), wykorzystywane jako odniesienie do predykcji i interpolacji pozostałych obrazów.
- Kadry predykcyjne (P - *predicted*) są kodowane względem poprzednich obrazów (I lub P) oraz są używane jako kadry referencyjne dla kadrów następnych.
- Kadry interpolowane (B - *bidirectional prediction*), pozwalają uzyskać najsilniejszą kompresję, ale wymagają zarówno kadrów odniesienia spośród obrazów poprzedzających jak i następnych (kadry I i P). Kadry interpolowane nie są nigdy wykorzystywane jako referencyjne.

Organizacja obrazów jest elastyczna i może być dostosowana do wymagań konkretnych aplikacji. Relacje pomiędzy trzema rodzajami kadrów przedstawia rys. 15.4.



Rys. 15.4. Schemat międzyobrazowego kodowania w MPEG.

Estymacja ruchu

We wszystkich przypadkach, kiedy obraz jest kodowany z uwzględnieniem kadru odniesienia, w celu poprawy skuteczności kompresji, stosowana jest estymacja ruchu. Polega ona na wyekstrapolowaniu z sekwencji obrazów informacji o ruchu określonych obiektów. Standard MPEG określa sposób reprezentacji informacji o ruchu jako jeden lub dwa wektory ruchu (w zależności od rodzaju predykcji) przypadające na pojedynczy obiekt, którym jest blok o rozmiarach 16×16 . Natomiast standard nie specyfikuje sposobu wyznaczania tych wektorów.

Kompensacja ruchu

Najczęściej stosowaną techniką redukcji nadmiarowości w sekwencji obrazów jest predykcja. Kompensacja ruchu poprzez predykcję zakłada, że "lokalnie" bieżący obraz może być modelowany jako translacja poprzedniego obrazu w sekwencji. Informacja o ruchu jest częścią informacji koniecznej do odtworzenia obrazu ze skompresowanej reprezentacji.

Kompensacja ruchu przez interpolację pozwala poprawić mechanizm dostępu do dowolnie wybranego kadru z sekwencji, zredukować wpływ błędów transmisji oraz szumu. Jednocześnie można osiągnąć wzrost efektywności kompresji dzięki lepszemu mechanizmowi przewidywania, uwzględniającemu zmiany informacji zawartej w obrazach pojawiających się w sekwencji później. Aby zwiększyć elastyczność techniki kompresji sekwencji obrazów, dla każdego bloku 16×16 w kadrach interpolowanych możliwy jest wybór jednego z czterech typów predykcji (brak predykcji, predykcja wprzód, predykcja w tył, interpolacja) przedstawionych w tabeli 1.

TABELA 15.1. Typy predykcji stosowane w MPEG; oznaczenia: \bar{x} - współrzędne elementu obrazu, \overline{mv}_{01} - wektor ruchu w odniesieniu do obrazu I_0 , \overline{mv}_{21} - wektor ruchu w odniesieniu do obrazu I_2 .

Typ bloku	Predyktor	Błąd predykcji
Brak predykcji	$\hat{I}_1(\bar{x}) = 128$	$I_1(\bar{x}) - \hat{I}_1(\bar{x})$
Predykcja wprzód	$\hat{I}_1(\bar{x}) = \hat{I}_0(\bar{x} + \overline{mv}_{01})$	$I_1(\bar{x}) - \hat{I}_1(\bar{x})$
Predykcja w tył	$\hat{I}_1(\bar{x}) = \hat{I}_2(\bar{x} + \overline{mv}_{21})$	$I_1(\bar{x}) - \hat{I}_1(\bar{x})$
Interpolacja	$\hat{I}_1(\bar{x}) = \frac{1}{2}[\hat{I}_0(\bar{x} + \overline{mv}_{01}) + \hat{I}_2(\bar{x} + \overline{mv}_{21})]$	$I_1(\bar{x}) - \hat{I}_1(\bar{x})$

Informacja o ruchu dla każdego bloku jest kodowana różnicowo względem informacji wyznaczonej w bezpośrednio poprzedzającym bloku. Zakres danych opisujących różnicowe wektory ruchu są dobierane dla konkretnej aplikacji - charakteru sekwencji obrazów, ich rozdzielczości, rodzaju zarejestrowanego ruchu i jest bezpośrednio związany z założonym obszarem poszukiwań optymalnego estymatora ruchu dla każdego bloku.

Redukcja nadmiarowości przestrzennych

Kodowanie obrazów oryginalnych i różnicowych (z predykcji) oraz składowych obrazu (luminancja, składowe chrominancji) przebiega podobnie. W podstawowym schemacie kompresji ośmiobitowe dane wejściowe dzielone są na bloki 8x8, w których niezależnie obliczana jest DCT. Współczynniki transformaty z każdego bloku podlegają równomiernej kwantyzacji, a następnie uszeregowaniu według sekwencji zygzak i kodowaniu. Do kodowania używa się zmodyfikowanej metody Huffmana (w połączeniu z metodą kodowania długości sekwencji).

Główne idee MPEG 4 i MPEG 7

15.4. Kompresja progresywna

15.5. Algorytmy obietowe (drugiej generacji)

15.6. Algorytm CALIC

Metoda przedstawiona w tym punkcie jest uznawana obecnie za najefektywniejszą metodę bezstratnej kompresji obrazów. Pozwala uzyskać bardzo małe wartości średnich bitowych przy stosunkowo niewielkim czasie kompresji i małych wymaganiach pamięciowych - zobacz wyniki przedstawione na końcu tej sekcji.

Technika CALIC (ang. Context-based Adaptive Lossless Image Codec) wykorzystuje adaptacyjne predycyjne kodowanie z kontekstem wyższych rzędów, a także podobną do metod interpolacyjnych analizę obrazu na coraz większym poziomie rozdzielczości. Dużą efektywność kodowania zawdzięcza nieliniowemu, odpowiednio dobranemu kontekstowemu modelowi predykcji, który wykorzystuje odpowiednio dobrane (na podstawie doświadczeń) wartości współczynników predykcji. Niskie koszty aplikacji osiąga z kolei poprzez ciekawy

pomysł kwantyzacji kontekstu, który pozwala uzyskać dużą skuteczność kodowania przy niewielkim rozmiarze tablic prawdopodobieństw warunkowych. Modulowanie kontekstu oraz predykcja algorytmicznie są proste, są zrealizowane w większości w arytmetyce liczb całkowitych i prostą logikę. Tak więc czas wykonania predykcji i modelowania (w koderze i dekoderze jest identyczny) dla obrazu 512×512 wynosi 1.52 sekundy na komputerze SUN SPARC10.

Do realizacji ostatniej fazy algorytmu CALIC można użyć entropijnego kodera dowolnego rodzaju (Huffmana, arytmetyczne), a cechy całego schematu kompresji zdają się pozwalać przy pewnych modyfikacjach na realizację tej metody w sposób progresywny do zastosowań transmisyjnych.

Twórcy techniki CALIC zdawali sobie dobrze sprawę z wymagań stawianych dobrej metodzie odwracalnej kompresji. Nic dziwnego, skoro zgłosili swoją metodę do komitetu ISO jako propozycję nowego standardu bezstratnej kompresji. Zasadniczym elementem ich schematu kompresji jest modelowanie kontekstu. Kryteria modelowania kontekstu dla bezstratnego kodowania obrazu w znaczeniu maksymalnej kompresji oraz praktycznej przydatności są następujące:

- a) silna dekorelacja - przyjęty model obrazu pozwala całkowicie określić i zredukować statystyczną nadmiarowość źródła.
- b) uniwersalność - model obrazu może szybko przystosować się (adoptować) do nieznannej statystyki źródła.
- c) niski koszt obliczeniowy - same konteksty oraz modelowanie obrazu i predykcja oparte na tych kontekstach mogą być efektywnie obliczane (wyznaczone).
- d) niskie wymagania pamięciowe - przestrzeń pamięci przeznaczona do gromadzenia kontekstów i informacji związanych z tymi kontekstami używana w modelowaniu i predykcji nie powinna być nadmierna.

Zaproponowali więc rozwiązanie, które w zadawalający sposób spełnia te kryteria, przez co uznawane jest powszechnie jako *state of art* współczesnych technik odwracalnej kompresji. Krótka charakterystyka techniki CALIC, napisana na podstawie [1], przedstawia się następująco.

1. Trzy-stopniowy schemat kodowania predykcyjnego z przeplotem.

Niech obraz oryginalny, wielopoziomowy ze skalą szarości, o szerokości W i wysokości H będzie opisany przez $I[i, j]$, $0 \leq i < W$, $0 \leq j < H$. Kodowanie predykcyjne realizowane jest w trzech kolejnych etapach, schematycznie pokazanych na rys. 15.5. Na każdym etapie przeprowadzane jest przeplatane próbkowanie obrazu oryginalnego. Celem realizacji modelu sekwencyjnej predykcji jest stworzenie możliwości predykcji kodowanych wartości przy pomocy najlepszego kontekstu. Już w drugim etapie udaje się zbudować otoczenie w kącie prawie 360 stopni, przy czym jednak jest to otoczenie dość odległe. Najlepsza sytuacja występuje w trzecim etapie, kiedy model predykcji obejmuje otoczenie z każdej strony i jednocześnie jest to otoczenie najbliższe. Warto przy tym zauważyć, że w trzecim etapie predykcji przy tak dobrym modelu kodowana jest połowa pikseli obrazu. Jest to jeden z ważnych czynników pozwalających uzyskać dużą skuteczność kompresji całego metody.

Pierwszy etap

- A. Tworzony jest podobraz o dwukrotnie zmniejszonej rozdzielczości w obu kierunkach $W/2 \times H/2$ w sposób następujący:

$$\mu[i, j] = \left[\frac{I[2i, 2j] + I[2i+1, 2j+1]}{2} \right], \quad 0 \leq i < W/2, \quad 0 \leq j < H/2 \quad (15.5)$$

B. Podobraz μ koduje się wykorzystując typowy kontekst dla DPCM, np.

$$\hat{\mu} = \frac{\mu[i, j-1] + \mu[i-1, j]}{2} + \frac{\mu[i+1, j-1] - \mu[i-1, j-1]}{4}, \quad 0 \leq i < W/2, \quad 0 \leq j < H/2 \quad (15.6)$$

Współczynniki predykcji zostały wyznaczone metodą regresji liniowej przy pomocy zbioru obrazów treningowych (zaokrąglone zostały do potęgi dwójki ze względu na prostotę obliczeń).

Drugi etap

A. Ten sam zakodowany już podobraz μ jest stosowany jako kontekst predykcji pikseli leżącej na diagonalnej elementarnej komórki czterech pikseli (rys. 15.5). Kodowane są piksele leżące wyżej na diagonalnej (jest ich $WH/4$), opisane jak niżej:

$$I[2i, 2j], I[2i+1, 2j+1], \quad 0 \leq i < W/2, \quad 0 \leq j < H/2 \quad (15.7)$$

B. Aby zakodować wartości $I[2i, 2j]$ stosuje się następujący model liniowej predykcji:

$$\hat{x} = 0.9\mu[i, j] + \frac{I[2i-1, 2j+1] + I[2i-1, 2j-1] + I[2i+1, 2j-1]}{6} - 0.05(I[2i-2, 2j] + I[2i, 2j-2]) - 0.15(\mu[i+1, j] + \mu[i, j+1]). \quad (15.8)$$

Natomiast wartości $I[2i+1, 2j+1]$ nie trzeba już kodować. Są one w dekodерze rekonstruowane z zależności:

$$I[2i+1, 2j+1] = 2\mu[i, j] - I[2i, 2j]. \quad (15.9)$$

Może przy tym wystąpić błąd obcięcia reszty ułamkowej, gdy przy liczeniu średniej diagonalnej μ suma dwóch wartości pikseli jest liczbą całkowitą nieparzystą (można go zignorować).

Trzeci etap

A. Należy zakodować dwie pozostałe wartości pikseli obrazu oryginalnego z każdego elementarnego bloku czterech pikseli, a mianowicie

$$I[2i+1, 2j], I[2i, 2j+1], \quad 0 \leq i < W/2, \quad 0 \leq j < H/2. \quad (15.10)$$

Dla zwiększenia skuteczności predykcji wykorzystano kontekst '360 stopni' składający się z czterech pikseli najbliższego sąsiedztwa w rastrze cztero-połączeniowym oraz dwóch pikseli z sąsiedztwa ośmio-połączeniowego (wszystkie one są już wcześniej kodowane oraz dekodowane, a więc dostępne w procesie rekonstrukcji).

B. Zastosowano następujący model predykcji:

$$\hat{x} = \frac{3}{8}(I[i-1, j] + I[i, j-1] + I[i+1, j] + I[i, j+1]) - \frac{I[i-1, j-1] + I[i+1, j-1]}{4} \quad (15.11)$$

2. Modelowanie i kwantyzacja kontekstu.

Dotąd do modelowania kontekstów predykcji poszczególnych pikseli x wykorzystywano K wartości sąsiednich pikseli x_1, x_2, \dots, x_K , przy czym $K = 4, 9, 6$ odpowiednio na 1, 2 i 3 etapie predykcji. Aby przy entropijnym kodowaniu uzyskać minimalną długość kodu wyjściowego dla zbioru wartości $e = x - \hat{x}$ (gdzie \hat{x} wyznaczono dla każdej wartości x w trój etapowej predykcji) potrzeba maksymalizować prawdopodobieństwo warunkowe $p(e/x_1, \dots, x_K)$. Używając standardowych metod np. kodowania arytmetycznego z założonym modelem Markowa n -tego rzędu potrzeba wówczas określić w sposób istotny statystycznie model o 2^{2^K} stanach (Z -liczba poziomów szarości), co jest problemem niebagatelnym! Do zapamiętania takiego modelu statystycznego potrzeba bardzo dużo pamięci, a nabranie przez ten model statystycznej istotności wymaga bardzo dużego zbioru danych (problem 'rozrzedzenia kontekstu'). Jest więc to rozwiązanie niepraktyczne.

Zastosowano więc następujący model kwantyzacji kontekstu w celu prostszego określenia wartości prawdopodobieństw warunkowych oraz korekcji przewidywanej wartości w pętli sprzężenia zwrotnego. Pomysł przedstawia się następująco:

A. aby ułatwić modelowanie kontekstu błędów DPCM kwantyzuje się kontekst x_1, x_2, \dots, x_K związany z przewidywaną wartością \hat{x} do liczby binarnej $t = t_K \dots t_1$ składającej się z K bitów tak, że:

$$t_k = \begin{cases} 0 & \text{if } x_k \geq \hat{x} \\ 1 & \text{if } x_k < \hat{x} \end{cases}, \quad 1 \leq k \leq K. \quad (15.12)$$

Liczba t reprezentuje wyższego rzędu przestrzenną strukturę modelującego kontekstu, czyli cechę obrazu, która może wskazywać na zachowanie się błędu predykcji DPCM, czyli wyrażać korelację otoczenia z błędem predykcji. Można w ten sposób uwzględnić krawędzie, rogi, tekstury, które nie mogły być wychwycone przez DPCM (którego współczynniki są globalne, wspólne dla całego obrazu).

B. obliczana jest także wartość wielkości zwanej dyskryminatorem mocy błędu, która charakteryzuje zmienność wartości kontekstu (jego gładkość), w sposób następujący:

$$\Delta = \sum_{k=1}^K w_k |x_k - \hat{x}|. \quad (15.13)$$

Na jej podstawie można estymować prawdopodobieństwo warunkowe błędu predykcji jako $p(e/\Delta)$ zamiast $p(e/x_1, \dots, x_K)$. Aby zlikwidować problem 'rozrzedzenia kontekstu' przy estymacji $p(e/\Delta)$ dokonywana jest kwantyzacja wartości Δ na L poziomów (w praktyce najczęściej $L=8$). Łącząc teraz, jako iloczyn kartezjański $L \times 2^K$, kwantowany do L poziomów dyskryminator Δ oraz 2^K kwantowanych wzorców tekstury kontekstu t_k otrzymujemy ostatecznie kwantyzację 2^{2^K} stanów źródła Markowa do znacznie zredukowanej liczby $L2^K$ stanów. Tak uproszczony kwantowanym kontekstem oznaczony jest następująco: $C(d, t)$, $0 \leq d < L$, $0 \leq t < 2^K$. Pozostaje oczywiście ważny problem - jak dobrać wartości współczynników w_k . Wartość Δ jest ustalana jako średniokwadratowa estymata $|e|$ (z minimalnym błędem średniokwadratowym naśladuje wartości $e = x - \hat{x}$). Określa się więc wartości współczynników w_k stosując standardową

metodę regresji liniowej. Tak więc mając przykładowy predyktor DPCM $\hat{x} = \sum_{k=1}^K a_k x_k$ jednego z trzech etapów predykcji, jest wyznaczany treningowy zbiór S wartości $|e| = |x - \hat{x}|$ oraz $|x_k - \hat{x}|$, $1 \leq k \leq K$. Następnie wartości w_k , $1 \leq k \leq K$ są obliczane metodą regresji liniowej tak, aby zminimalizować wartość:

$$\sum_S \{|e| - \Delta\} = \sum_S \left\{ |x - \hat{x}| - \sum_{k=1}^K w_k |x_k - \hat{x}| \right\} \quad (15.14)$$

na treningowym zbiorze danych.

3. Adaptacyjne, oparte na kontekście modelowanie błędu (wraz z drugą fazą kwantyzacji)

Jednak $L2^K$ stanów kontekstu w danym modelu to jednak dalej za dużo do dobrej estymacji prawdopodobieństw warunkowych, w tym przypadku $p(e/C(d,t))$. Wobec tego następuje:

- A. estymowanie warunkowej wartości oczekiwanej $E\{e/C(d,t)\}$ poprzez wyznaczenie odpowiedniej średniej próbek $\bar{e}(d,t)$ dla różnych kwantowanych kontekstów. Intuicyjnie znacznie mniej próbek potrzeba do dobrej estymacji warunkowej wartości oczekiwanej niż prawdopodobieństwa ($O(n^{-0.5})$ versus $O(n^{-0.4})$). Ponieważ średnia warunkowa $\bar{e}(d,t)$ lepiej estymuje błąd predykcji DPCM w kwantowanym kontekście $C(d,t)$, można skompensować błąd DPCM poprzez modyfikację predykcji x z \hat{x} na predykcję wartości \dot{x} z wartości zmiennej określonej następująco: $\dot{x} = \hat{x} + \bar{e}(d,t)$. Nowy predyktor \dot{x} jest adaptacyjny, nieliniowy i realizuje mechanizm sprzężenia zwrotnego błędu predykcji z opóźnieniem jednego kroku. Wykorzystując nowy predyktor mamy teraz nowy błąd predykcji: $\varepsilon = x - \dot{x}$ dla entropijnego kodowania błędu. Rozkład wartości błędu predykcji zachowuje w tym przypadku postać rozkładu Laplace'a, ale jest wyraźnie wyostrzony (co powinno poprawić skuteczność entropijnego kodowania).
- B. optymalizacja kwantyzatora Q wartości Δ . Kryterium kwantyzacji jest minimalizacja warunkowej entropii wartości błędów zależnej od $p(\varepsilon/Q(\Delta))$. Ze zbioru obrazów treningowych obliczamy zbiór wartości par (ε, Δ) i używamy standardowej dynamicznej techniki wyboru wartości $0 = q_0 < q_1 < \dots < q_{L-1} < q_L = \infty$ dzielących przedział wartości Δ na L podprzedziałów: $S_d = \{\varepsilon | q_d \leq \Delta < q_{d+1}\}$, takich że wrazenie:

$$- \sum_{\varepsilon} p(\varepsilon) \log p(\varepsilon | \varepsilon \in S_d) \quad (15.15)$$

osiąga minimum.

Dwa procesy poszukiwania optymalnego schematu kwantyzacji (szukanie współczynników i podział przedziału wartości Δ na L podprzedziałów) mogą być przeprowadzane również w sposób łączny (kwantyzacja łączna).

- C. entropijne kodowanie wartości ε - adaptacyjne entropijne kodowanie (najlepiej arytmetyczne) z użyciem tylko L prawdopodobieństw warunkowych $p(\varepsilon / Q(\Delta) = d)$, $0 \leq d < L$ dla poszczególnych wartości ε .

4. Przerzucanie znaku błędu predykcji

Znaki warunkowych średnich próbek $\bar{\varepsilon}(d,t)$ mogą być użyte do wyostrenia prawdopodobieństw warunkowych $p(\varepsilon / Q(\Delta) = d)$, a więc redukcji warunkowej entropii. Dla

dwóch różnych kontekstów $C(d, t_1)$ i $C(d, t_2)$, wartości warunkowych średnich próbek $\bar{\varepsilon}(d, t_1)$ i $\bar{\varepsilon}(d, t_2)$ mogą mieć przeciwne znaki i odpowiednio różne wartości $p(\varepsilon|C(d, t_1))$ i $p(\varepsilon|C(d, t_2))$. Dla ustalonej wartości $0 \leq d < L$ można rozdzielić prawdopodobieństwo $p(\varepsilon|Q(\Delta)=d)$ na dwa:

$$p_+(\varepsilon|d) = p(\varepsilon|d, \bar{\varepsilon}(d, t) \geq 0), \quad p_-(\varepsilon|d) = p(\varepsilon|d, \bar{\varepsilon}(d, t) < 0). \quad (15.16)$$

Oczywistym jest, że oba prawdopodobieństwa warunkowe p_+ i p_- dają niższą entropię niż $p(\varepsilon|Q(\Delta)=d)$ (bardziej wyróżniona jest statystyka błędu). Wydaje się jednak że wówczas podwajają się: zużycie pamięci oraz rozmycie kontekstów.

Można zaobserwować, że $p_+(\varepsilon|Q(\Delta)=d)$ oraz $p_-(\varepsilon|Q(\Delta)=d)$ są w przybliżeniu swoim lustrzanym odbiciem względem wartości oczekiwanej (równej zero) rozkładu $p = p_+ + p_-$. Można więc przetrzucić p_- symetrycznie względem osi zerowej ($p_-(-\varepsilon|d)$) i nałożyć na p_+ otrzymując obciążony estymator prawdopodobieństwa $\hat{p}(\varepsilon|Q(\Delta)=d)$ o mniejszej wariancji w stosunku do $p(\varepsilon|Q(\Delta)=d)$, co widać wyraźnie na rys. 15.6 ($\hat{p}(\varepsilon|Q(\Delta)=d)$ jest bardziej wyostżony). Zastosowanie tego estymatora pozwala zmniejszyć średnią bitową kodu bez dodatkowej pamięci oraz bez rozrzedzenia kontekstu. Podsumowując algorytm przetrzucania znaku błędu jest następujący: przed kodowaniem wartości $\varepsilon = x - \hat{x}$, koder sprawdza, czy $\bar{\varepsilon}(d, t) < 0$ na podstawie kontekstu $C(d, t)$. Jeśli błąd predykcji jest ujemny, wówczas kodowana jest wartość $-\varepsilon$, a w pozostałych przypadkach ε . Ponieważ dekodery także zna $C(d, t)$ i $\bar{\varepsilon}(d, t)$ może, jeśli to konieczne, odwrócić znak w celu poprawnej rekonstrukcji wartości ε . Mamy więc tutaj do czynienia z przewidywaniem znaku błędu predykcji.

5. Ogólny opis algorytmu

Algorytm składa się z trzech głównych składników:

- adaptacyjna predykcja,
- modelowanie kontekstu,
- kodowanie z entropią warunkową,

przy czym modelowanie kontekstu wpływa zarówno na predykcję jak i entropijne kodowanie.

Algorytm 14.1. Kompresja obrazu metodą CALIC

Dla każdego z trzech etapów:

INICJALIZACJA: $N(d, t)=1, S(d, t)=0, 0 \leq d < L, 0 \leq t < 2^K$.

PARAMETRY: wyznaczone wcześniej optymalne wartości $a_k, w_k, 1 \leq k \leq K$.

Dla każdego piksela x z danego etapu

$$0. \quad \hat{x} = \sum_{k=1}^K a_k x_k \quad \text{dla kontekstu predykcji } x_1, x_2, \dots, x_K;$$

$$1. \quad \Delta = \sum_{k=1}^K w_k (x_k - \hat{x});$$

$$2. \quad d = Q(\Delta);$$

$$3. \quad \text{wyznacz wzorzec tekstury } t = t_K \dots t_1:$$

$$\text{IF } (x_k < \hat{x}) t_k = 0 \quad \text{ELSE } t_k = 1, 1 \leq k \leq K;$$

4. $\bar{e} = S(d,t) / N(d,t)$;
 5. $\hat{x} = x + \bar{e}$;
 6. $\varepsilon = x - \hat{x}$;
 7. $S(d,t) = S(d,t) + \square$; $N(d,t) = N(d,t) + 1$;
 8. IF $N(d,t) \geq 128$
 $S(d,t) = S(d,t) / 2$; $N(d,t) = N(d,t) / 2$;
 9. IF $S(d,t) < 0$ koduj $(-\square, d)$ ELSE koduj (\square, d) ;
 END;
 END.

Do najważniejszych zalet algorytmu techniki CALIC można zaliczyć złagodzenie problemu rozrzedzenia kontekstu poprzez:

- rozróżnienie kontekstu $C(d,t)$ dla modelowania (adaptacyjnej korekcji) błędu predykcji oraz kontekstu $Q(\Delta)$ wykorzystywanego do warunkowego entropijnego kodowania wartości błędu predykcji;
- estymatę pojedynczego parametru wartości oczekiwanej zamiast szeregu prawdopodobieństw warunkowych dla danego kontekstu.

Pozwoliło to uzyskać dużą liczbę kontekstów przy modelowaniu błędu, aby uchwycić bardziej złożone struktury błędu (złożone korelacje) w celu zwiększenia efektywności kodowania.

Ocena efektywności techniki CALIC

W tabeli poniżej przedstawiono porównanie skuteczności kompresji metody CALIC z innymi, najbardziej znanymi i najefektywniejszymi technikami kompresji odwracalnej.

Obraz	UCM	ECECO W	S+P	JPEG-LS	Sunset	CALIC
Balloon	2.81	2.86	2.97	2.90	2.89	<u>2.78</u>
Barb 1	4.44	4.34	4.53	4.69	4.68	<u>4.33</u>
Zelda	3.80	<u>3.71</u>	3.84	3.89	3.81	3.72
Hotel	4.28	4.38	4.53	4.38	4.46	<u>4.22</u>
Barb 2	4.57	4.57	4.71	4.69	4.77	<u>4.49</u>
Board	3.57	3.62	3.82	3.68	3.73	<u>3.50</u>
Girl	3.81	3.81	3.96	3.93	3.91	<u>3.71</u>
Gold	4.45	4.42	4.56	4.48	4.56	<u>4.38</u>
Boats	3.85	3.86	4.03	3.93	4.01	<u>3.77</u>
Lena	4.17	4.09	4.16	4.24	4.21	<u>4.04</u>
Średnia	3.97	3.98	4.12	4.08	4.10	<u>3.89</u>

Podkreślono najmniejszą średnią bitową dla każdego z obrazów oraz najmniejszą średnią. Techniki prezentowane w tej tabeli to:

UCM - uniwersalny algorytm modelowania kontekstu (Universal Context Modelling algorithm) opracowany przez Weinbergera [2], który ze względu na duży stopień złożoności i duże koszty czasowo-pamięciowe nie nadaje się właściwie do praktycznych zastosowań.

ECECOW - metoda zagnieżdżonego kodowania współczynników falkowej transformaty z wykorzystaniem entropii warunkowej i modelowaniem kontekstu wysokiego rzędu (Embedded Conditional Entropy Coding of Wavelet Coefficients) opracowana ostatnio przez Xiaolina Wu [3]. Technika ta w wersji bezstratnej wykorzystuje odwracalną transformację falkową (z całkowitymi współczynnikami), prosty zagnieżdżony algorytm sukcesywnej kwantyzacji, a koncentruje się na modelowaniu prawdopodobieństw warunkowych przy kodowaniu hierarchicznego drzewa współczynników. Wykorzystuje tutaj podobny pomysł jak w metodzie CALIC: modelowanie kontekstu wysokiego rzędu i jego kwantyzacja, przy czym szczególną uwagę zwraca na oddzielne kontekstowe kodowanie znaku współczynników.

S+P - bezstratna wersja techniki kompresji Saida i Pearlmana, której zasadniczy pomysł polega na partycjonowaniu podzbiorów hierarchicznego drzewa współczynników zwiększając przy tym efektywność ich kodowania [4].

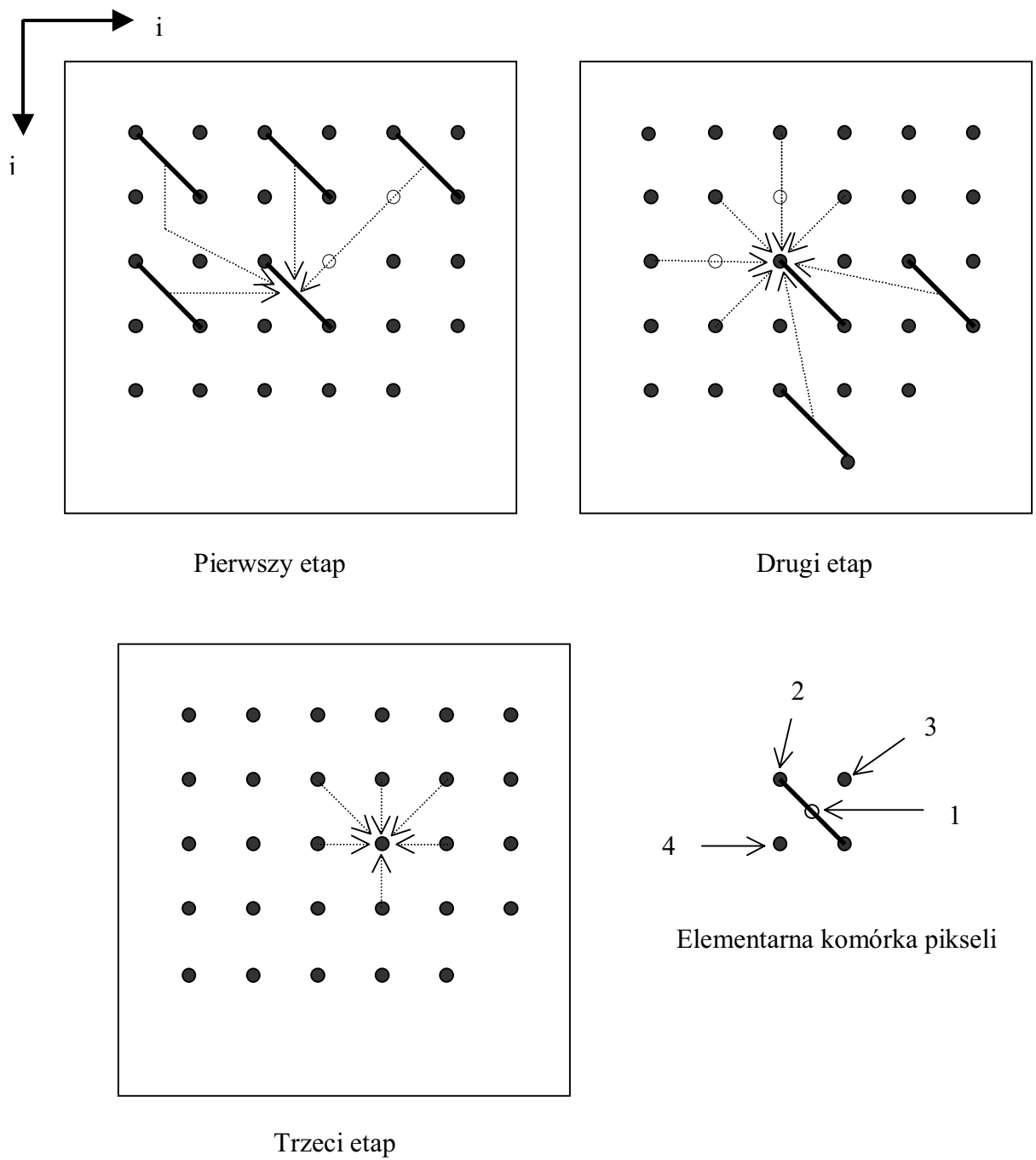
JPEG-LS - nowy standard bezstratnej kompresji [5] o znacznie zwiększonej efektywności w stosunku do poprzedniej wersji odwracalnego JPEG-a.

Sunset - odwracalny koder Langdona i Haidinyaka, oparty zasadniczo na kształtowaniu i parametryzacji zależnego od kontekstu rozkładu wartości kodowanych danych [6].

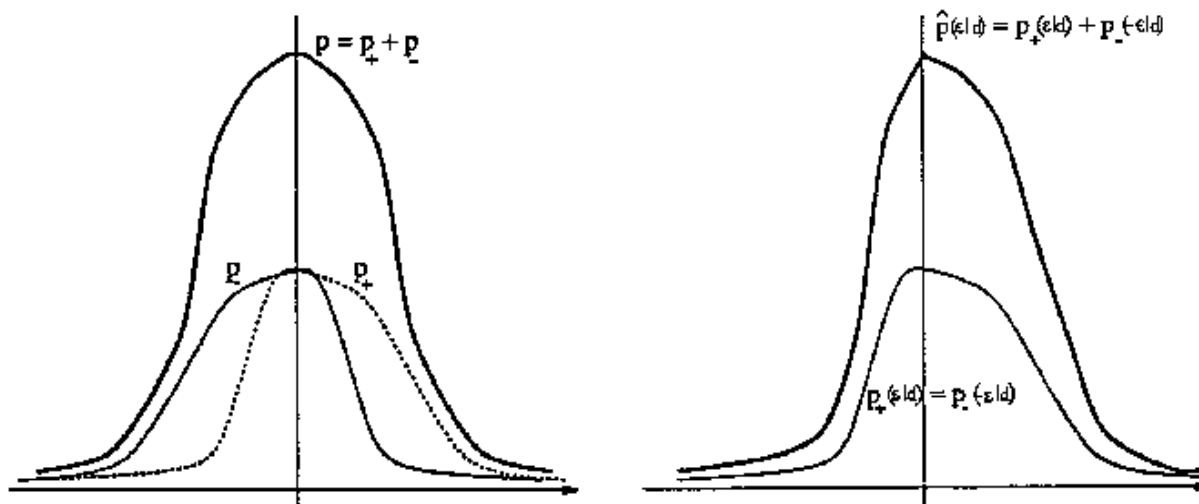
Przedstawione wyniki potwierdzają bardzo dobrą skuteczność techniki CALIC. Prawie dla każdego z dziesięciu testowych obrazów standardu JPEG metoda ta dała najmniejszą średnią bitową skompresowanej reprezentacji.

Literatura

1. X. Wu, "Lossless compression of continuous-tone images via context selection and quantization", IEEE Trans. on Image Processing, vol. 5, no. 6, pp. 656-664, 1996.
2. M. J. Weinberger, J. Rissanen, R. B. Arps, "On universal context modeling for lossless compression of the gray-scale images", pre-print, 1994.
3. X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression", pre-print, 1997.
4. A. Said, W. Pearlman, "An image multiresolution representation for lossless and lossy compression", IEEE Trans. on Image Processing, vol. 5, no. 9, pp. 1303-1310, 1996.
5. ISO/IEC JTC 1/SC 29/WG 1, "JPEG LS image coding system", ISO Working Document ISO/IEC JTC 1/SC 29/WG 1 N399 - WD14495, czerwiec 1996.
6. G. Langdon, C. Haidinyak, "Context-dependent distribution shaping and parametrization for lossless image compression", Applications of Digital Image Processing XVII, SPIE, pp. 62-70, 1994.



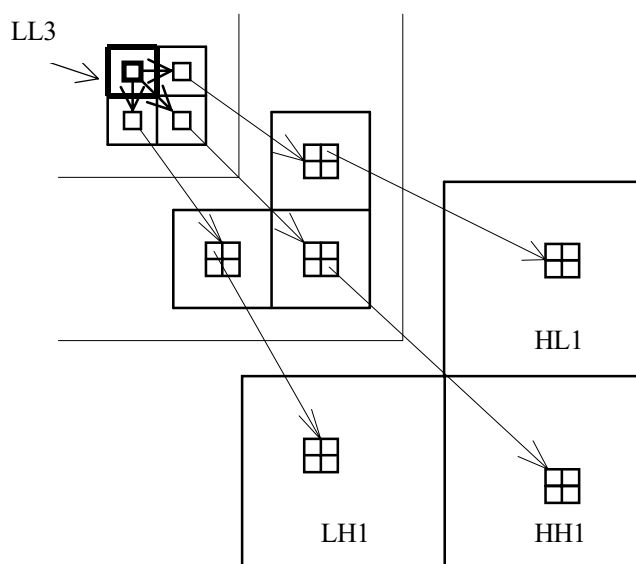
Rys. 15.5. Schemat modelowania kontekstu do trzy etapowej predykcji kodowanych wartości w CALIC-u. Cyfry przy elementarnej komórce pikseli oznaczają kolejność ich kodowania.



Rys. 15.6. Wyostrenianie prawdopodobieństwa warunkowego $p = p_+ + p_-$ poprzez przeczucanie p_- .

15.7. Algorytm EZW

Nazwa EZW oznacza algorytm oparty na zagnieżdżonym kodowaniu falkowym z drzewem zer (EZW- ang. embedded zerotree wavelet). Każdy współczynnik waveletowej transformaty jest indywidualnie i deterministycznie porównywany z tym samym zbiorem progów, aby ocenić jego znaczenie. Stąd każdy współczynnik jest traktowany jako istotna, potencjalnie ważna część danych w odniesieniu do rozważanej skali obrazu i żadne obliczenia statystyczne dla danego pasma nie są stosowane. W wyniku tego mała liczba deterministycznie znaczących współczynników określonej skali nie jest ignorowana ze względu na jej statystycznie nieistotne wartości.



Rys. 15.7. Oznaczenie pasm w drzewie transformaty wavelet obrazu.

Skrótowny opis algorytmu

Rozważany jest istotny aspekt kodowania pozycji tych współczynników transformaty, które będą transmitowane czy zapisywane jako niezerowe. Stosując skalarną kwantyzację i kodowanie entropijne, w celu uzyskania dużej efektywności kompresji prawdopodobieństwo najczęściej występującego symbolu alfabetu (po kwantyzacji jest to wartość zero) musi być bardzo duża. W typowych rozwiązaniach dużą część nowej reprezentacji danych zawiera zakodowana mapa znaczeń (ang. significance map) opisujących kolejne pozycje dwuwymiarowej mapy obrazu jako zerowe lub niezerowe. Powoduje to, że znaczące zmniejszenie długości kodu tej mapy pozwala na znaczne zwiększenie efektywności kompresji.

A. Kodowanie mapy znaczeń (ang. Significance Map Encoding)

Podstawowe określenia:

Waveletowy współczynnik jest nieznaczący względem pewnego progu T , jeśli $|x| < T$. W przeciwnym przypadku współczynnik jest znaczący.

Współczynnik wyższego poziomu drzewa dekompozycji (o mniejszej skali) nazywany jest rodzicem, a odpowiadające mu współczynniki na kolejnym poziomie o większej skali – dziećmi, natomiast na jeszcze wyższych poziomach – potomkami. Przodkami są wszystkie współczynniki leżące odpowiednio na wyższych poziomach w stosunku do danego współczynnika, powyżej rodzica.

Współczynnik x jest nazywany elementem drzewa zer dla danego progu T , jeśli on sam i wszystkie jego współczynniki potomne są nieznaczące.

Element drzewa zer dla progu T jest korzeniem drzewa zer jeśli nie jest potomkiem poprzednio znalezionego korzenia drzewa zer, czyli nie jest przewidywalnie nieznaczący z punktu widzenia poprzednio znalezionego korzenia drzewa zer w mniejszej skali przy tym samym T . Korzeń drzewa zer jest kodowany przy pomocy specjalnego symbolu, wskazującego że wszyscy jego potomkowie są nieznaczący.

Mapa znaczeń opisująca drzewo dekompozycji obrazu opisana jest więc alfabetem o czterech symbolach:

- 1) korzeń drzewa,
- 2) izolowane zero, czyli współczynnik nieznaczący posiadający wśród potomków współczynnik znaczący,
- 3) znaczący dodatni
- 4) znaczący ujemny

Dla współczynników pasma najniższych częstotliwości oraz największej skali (nie mających dzieci), rezerwuje się alfabet jedynie trójelementowy: nieznaczący, znaczący dodatni i ujemny.

B. Kwantyzacja sukcesywnej aproksymacji i entropijnego kodowania

Metoda sukcesywnej aproksymacji (SAQ) wykorzystuje kolejne wartości progów z sekwencji T_0, \dots, T_{N-1} aby określić znaczenie współczynników, gdzie progi są dobrane następująco: $T_i = T_{i-1} / 2$. Początkowa wartość progu T_0 jest dobrana następująco: $|x_j| < 2T_0$ dla wszystkich współczynników transformaty x_j .

Podczas kodowania (dekodowania), dwie oddzielne listy współczynników muszą być pamiętane. W każdym momencie lista dominująca (dominant) zawiera współrzędne tych współczynników, które jeszcze nie okazały się znaczące według ustalonej kolejności przeglądania (kolejne pasma od najniższych do najwyższych częstotliwości oraz

współczynniki w pasmach jak na rys. 15.7). Druga lista, zwana zależną (subordinate) zawiera wartości tych współczynników, które okazały się znaczące. Dla kolejnych wartości progu każda z list jest przeglądana raz. Podczas dominującego przeglądu współczynniki z listy dominującej są porównywane z progiem T_i w celu określenia ich znaczenia i ewentualnie znaku. Ta lista następnie jako mapa znaczeń jest kodowana w sposób opisany wyżej. Za każdym razem kiedy współczynnik jest określony jako znaczący, jego wartość bezwzględna jest dopisywana do listy zależnej, a współczynnik ten w tablicy wartości waveletowych współczynników jest zerowany. Przeгляд zależny następuje zaraz po przeglądzie dominującym. Polega on na doprecyzowaniu wartości współczynników dla dekodera o kolejny bit dokładności. Dla kolejnych współczynników to doprecyzowanie wartości może być kodowane binarnie. Poprzednia wartość progu wyznacza jednocześnie przedział niepewności wpisanych wartości. Zmniejszona o połowę wartość progu powoduje dookreślenie tych wartości z dwukrotnie mniejszym poziomem niepewności. Czyli jeśli rzeczywista wartość wpada w górną połowę starego przedziału niepewności, wówczas kodowana jest '1', a gdy w dolną – '0'. Ciąg binarnych symboli po takim przeglądzie jest następnie entropijnie kodowany, a doprecyzowane wartości na podrzędnej liście są sortowane w kolejności malejącej, co jest możliwe do powtórzenia także w dekodерze.

Te dwa przeglądy są naprzemiennie powtarzane z kolejnymi, malejącymi wartościami progów.

Dekodując współczynniki podczas kolejnych przeglądów dominującego i zależnego doprecyzowuje się ich wartość zmniejszając przedział niepewności, w którym rzeczywiste wartości współczynników mogą się pojawić. Zrekonstruowana wartość może być dowolną w ostatnim przedziale niepewności. Stosuje się tu różne rozwiązania, np. dla minimalizacji błędu średniokwadratowego można użyć metodę centroidu z założonym modelem funkcji gęstości prawdopodobieństwa lub też optymalną w sensie MINMAX metodę środka przedziału niepewności (takie rozwiązanie zastosował Shapiro).

Proces kodowania kończy się, gdy wyczerpuje się założony na skompresowaną reprezentację limit bitów.

C. Hierarchia ważności kodowanej informacji

Informacja kodowana w algorytmie EZW jest uporządkowana zgodnie z pewną hierarchią, która decyduje o tym, co jest przesyłane (kodowane) w pierwszej kolejności jako informacja najbardziej znacząca, co w drugiej itd. Struktura ważności jest następująca:

1. Numeryczna precyzja wartości współczynników, określona przez próg (przedział niepewności).
2. Wartość współczynników określona z dokładnością do ostatniego przedziału niepewności.
3. Skala czyli założona kolejność kodowania pasm wielorozdzielczej analizy obrazu – od najniższych częstotliwości (najmniejszej skali) do pasm wysokoczęstotliwościowych (największej skali).
4. Położenie w przestrzeni określone przez kolejność skalowania współczynników danego pasma w czasie przeglądu dominującego.

PRZYKŁAD 15.1. Rozważmy prosty przykład trójpoziomowej (o trzech skalach) waveletowej transformacji miniaturowego obrazu o rozmiarach 8×8 pokazano na rys. 15.8. Wartości współczynników falkowych przybliżono do najbliższej liczby całkowitej. Prześledzony zostanie algorytm sukcesywnego kodowania informacji zawartej w tych wartościach.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Rys. 15.8. Współczynniki transformaty falkowej przykładowego obrazu o wymiarach 8×8 .

Ponieważ największa wartość współczynnika wynosi 63, jako początkową wartość progu możemy przyjąć dowolną wartość z przedziału $(31.5, 63]$. Przyjmujemy więc $T_0 = 32$. Tabela poniżej obrazuje pierwszy przegląd dominujący z wartością progu równą 32. Oznaczenia są następujące:

POZ – znaczący dodatni, NEG – znaczący ujemny, IZ – izolowane zero, KDR – korzeń drzewa, Z – zero zamiast IZ i KDR na poziomie 1.

Tablica 15.2. Kodowanie współczynników falkowych techniką EZW: przegląd dominujący.

Komentarz	Pasmo	Wartość współczynnika	Symbol	Wartość rekonstruowana
(1)	LL3	63	POZ	48
	HL3	-34	NEG	-48
(2)	LH3	-31	IZ	0
	(3)	HH3	23	KDR
		HL2	49	POZ
(4)	HL2	10	KDR	0
		HL2	14	KDR
	HL2	-13	KDR	0
	LH2	15	KDR	0
(5)	LH2	14	IZ	0
		LH2	-9	KDR
	LH2	-7	KDR	0
(6)	HL1	7	Z	0
		HL1	13	Z
	HL1	3	Z	0
	HL1	4	Z	0
	LH1	-1	Z	0
(7)	LH1	47	POZ	48
		LH1	-3	Z
	LH1	-2	Z	0

Natomiast pierwszy zależny przegląd pokazuje następująca tabela:

Tablica 15.3. Kodowanie współczynników falkowych techniką EZW: przegląd zależny.

Wielkość współczynnika	Symbol	Wielkość zrekonstruowana
63	1	56
34	0	40
49	1	56
47	0	40

Na końcu kolejność tych wielkości zostaje zmieniona na (63,49,34,47) według informacji dostarczanej dla dekodera.

Następny przegląd dominujący przeprowadzany jest z progiem 16. Teraz jedynie współczynniki dotąd nieznaczące są skanowane. Ponadto współczynniki już określone jako znaczące są traktowane jako zero przy szukaniu korzeni drzewa zer. Tak więc w drugim dominującym przeglądzie kodowane są: -31 z LH3 jako NEG, 23 z pasma HH3 jako POZ, wszystkie współczynniki z HL2 oraz LH2 są KDR. Przegląd kończy się w momencie, gdy pozostałe do przejrzania współczynniki są przewidywane jako nieznaczące.

Drugi przegląd listy zależnej zawierającej teraz sześć elementów (63,49,34,47,31,23) dotyczy trzech przedziałów niepewności: [48,64), [32,48) oraz [16,31), każdy o szerokości 16. Doprecyzowanie każdej wielkości polega na określeniu dla każdego z tych przedziałów dwóch nowych przedziałów niepewności. Na końcu tego przeglądu kolejność wielkości współczynników jest następująca (63,49,47,34,31,23). Odpowiadają im zrekonstruowane wartości (60,52,44,36,28,20). Kodowanie jest dalej kontynuowane poprzez naprzemienne przeglądanie nadrzędne i zależne współczynników i może być zatrzymane w dowolnym czasie.

14.8. Algorytm SPIHT

Algorytm ten jest modyfikacją metody EZW lub inaczej - jej doskonalszą realizacją. Wykorzystuje przede wszystkim pomysł podziału list zawierających odpowiednio sklasyfikowane współczynniki falkowe, zmieniając nieco zasadę kodowania map znaczeń. Powstaje strumień bitowy zarówno w części dominującej jak i zależnej. A oto krótka charakterystyka algorytmu podziału zbiorów drzewa hierarchicznego (SPIHT - ang. set partitioning in hierarchical trees).

Transmisja wartości współczynników

Podstawowym zagadnieniem, potrzebnym do zrozumienia algorytmu, jest reprezentacja danych przyjęta w metodzie podziału zbiorów. Przedstawia ją tabela 15.4.

TABELA 15.4. Reprezentacja danych wykorzystana w algorytmie SPIHT.

znak		z	z	z	z	z	z	z	z	z	z	z	z	z	z	
msb	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
	4	→		1	1	0	0	0	0	0	0	0	0	0	0	
	3					→	1	1	1	1	0	0	0	0	0	
	2										→	1	1	1	1	
	1															→
lsb	0															→

Binarna reprezentacja współczynników uporządkowanych według wartości wielkości bezwzględnej.

Zakładamy, że współczynniki są porządkowane według minimalnej liczby bitów potrzebnej do reprezentacji ich wartości bezwzględnej, czyli określone jest przekształcenie $\eta : I \rightarrow I^2$, takie że:

$$\lfloor \log_2 |c_{\eta(k)}| \rfloor \geq \lfloor \log_2 |c_{\eta(k+1)}| \rfloor, \quad k = 1, \dots, N, \quad (15.16)$$

gdzie k - indeks kolejnego współczynnika c , N - liczba współczynników transformaty wavelet.

Oznaczmy przez μ_n liczbę współczynników takich, że $2^n \leq |c_{i,j}| < 2^{n+1}$. Dla danych z tabeli 15.4 mamy więc kolejno $\mu_5 = 2$, $\mu_4 = 2$, $\mu_3 = 4$, $\mu_2 = 7$. Kodując tak uszeregowane wartości wystarczy zapisać wartości μ_n , a następnie bity kolejnych wartości poniżej jedynek na pozycji najstarszego bitu dla danej wartości.

Progresywna transmisja tak przygotowanych wartości może być opisana przez następujący algorytm.

Algorytm 15.2. Ogólna zasada kodera SPIHT.

- 1) wysłać $n = \lfloor \log_2 (\max_{(i,j)} \{|c_{i,j}|\}) \rfloor$ do dekodera;
- 2) wysłać μ_n wraz ze współzrędnymi i znakami kolejnych współczynników, takich że $2^n \leq |c_{\eta(k)}| < 2^{n+1}$; jest to tzw. **przeгляд porządkujący** (ang. sorting pass);
- 3) wysłać n -ty najbardziej znaczący bit wszystkich współczynników, takich że $|c_{i,j}| \geq 2^{n+1}$ (tzn. tych współczynników, których współzrędnne zostały przesłane w poprzednim przeglądzie porządkującym, z zachowaniem tej samej kolejności); jest to tzw. **przeгляд uściślający** (ang. refinement pass);
- 4) zmniejszyć n o jeden i skoczyć do 2).

Algorytm sortowania z dzieleniem zbiorów

Główną zaletą tego algorytmu jest fakt, iż informacje o położeniu poszczególnych współczynników nie są bezpośrednio transmitowane. Kierunek przeglądania wartości współczynników jest ustalony w algorytmie kodera i jeśli jest dokładnie powtórzony w dekodерze, wówczas przestrzenne położenie poszczególnych współczynników może być wiernie odtworzone.

Ponadto nie trzeba w tym algorytmie sortować wszystkich współczynników (jak w klasycznym EZW). Potrzeba jedynie selekcji współczynników do poszczególnych grup μ_n , gdzie wartość n jest zmniejszana w kolejnych iteracjach. Dla danego n , jeśli $|c_{i,j}| \geq 2^n$ wówczas współczynnik $c_{i,j}$ jest określany jako znaczący. W przeciwnym wypadku mamy do czynienia ze współczynnikiem nieznaczącym.

Algorytm sortowania dzieli zbiór wartości współczynników na podzbiory T_m , dla których wykonywany jest test określenia znaczenia zbioru:

$$S_n(T_m) = \begin{cases} 1, & \max_{(i,j) \in T_m} \{|c_{i,j}|\} \geq 2^n \\ 0, & \text{wpp.} \end{cases} \quad (15.17)$$

Jeśli $S_n(T_m) = 0$ (tzn. podzbiór jest nieznaczący), to wówczas można stwierdzić, że wszystkie należące do niego współczynniki są nieznaczące. Jeśli podzbiór jest znaczący, wówczas stosowana jest pewna reguła podziału podzbioru T_m na nowe podzbiory $T_{m,l}$, dla których wykonuje się test określenia znaczenia podzbioru. Taki proces podziału zbiorów jest kontynuowany aż do określenia znaczenia wszystkich pojedynczych podzbiorów i do identyfikacji każdego znaczącego współczynnika.

Aby zredukować liczbę koniecznych porównań przy określaniu znaczeń podzbiorów zdefiniowano zasadę podziału zbioru, która wykorzystuje spodziewany porządek wartości współczynników, wynikający z hierarchicznej struktury danych w piramidzie pasmowej dekompozycji obrazu. Części przestrzennego drzewa dekompozycji są kolejnymi podzbiorem w algorytmie sortowania. Spodziewane są duże nieznaczące podzbiory oraz znaczące podzbiory zawierające tylko jeden element.

Przyjmijmy następujące oznaczenia, wynikające z hierarchicznego drzewa falkowej dekompozycji obrazu:

- $\mathbf{O}(i,j)$: zbiór współrzędnych wszystkich węzłów dziecięcych węzła (i,j) ;
- $\mathbf{D}(i,j)$: zbiór współrzędnych wszystkich węzłów potomnych węzła (i,j) ;
- \mathbf{H} : zbiór współczynników korzeni wszystkich drzew przestrzennej zależności współczynników (węzły na najwyższym poziomie piramidy).
- $\mathbf{L}(i,j) = \mathbf{D}(i,j) - \mathbf{O}(i,j)$.

Reguły dzielenia zbiorów są następujące:

- 1) Początkowy podział formowany jako zbiory $\{(i,j)\}$ i $\mathbf{D}(i,j)$ dla wszystkich $(i,j) \in \mathbf{H}$;
- 2) Jeśli $\mathbf{D}(i,j)$ jest znaczący, wówczas jest dzielony na podzbiór $\mathbf{L}(i,j)$ i cztery zbiory jednoelementowe współczynników o współrzędnych $(k,l) \in \mathbf{O}(i,j)$;
- 3) Jeśli $\mathbf{L}(i,j)$ jest znaczący, jest wtedy dzielony na cztery zbiory $\mathbf{D}(k,l)$, gdzie $(k,l) \in \mathbf{O}(i,j)$.

Algorytm kodowania

W praktycznej implementacji przedstawionych rozwiązań informacja o znaczeniu współczynników przechowywana jest na trzech uporządkowanych listach:

- lista nieznaczących zbiorów (ang. list of insignificant sets LIS),
- lista nieznaczących współczynników (ang. list of insignificant pixels LIP),
- lista znaczących współczynników (ang. list of significant pixels LSP).

W każdej z tych list poszczególne elementy są identyfikowane przez współrzędne (i,j) , przy czym elementem na listach LIP i LSP są pojedyncze współczynniki, podczas gdy na liście LIS występują zbiory $\mathbf{D}(i,j)$ (element typu A tej listy) lub $\mathbf{L}(i,j)$ (element typu B).

Podczas przeglądu sortującego (porządkującego) elementy zbioru LIP (te, które były nieznaczące w poprzednim przeglądzie) są testowane, a te które okażą się znaczące, są przenoszone do listy LSP. Podobnie zbiory z listy LIS są kolejno (według ustalonego porządku) testowane i w przypadku, kiedy zbiór jest znaczący, usuwa się go z listy LIS i dzieli. Nowe podzbiory z większą niż jeden liczbą elementów są dopisane do listy LIS,

natomiast jednoelementowe zbiory współczynników są dopisywane na koniec odpowiednio list LIP lub LSP, w zależności od ich znaczenia. Zbiór LSP zawiera współrzędne danych, które są przeglądane w trybie uściślającym.

Prowadzi to do następującego algorytmu progresywnego kodowania danych.

Algorytm 15.3. Kodowanie współczynników falkowych metodą SPIHT.

- 1) Inicjalizacja: wyślij $n = \lfloor \log_2(\max_{(i,j)} \{|c_{i,j}|\}) \rfloor$; ustawić LSP jako listę pustą oraz dodać współrzędne $(i, j) \in \mathbf{H}$ do LIP, a także współrzędne tych współczynników, które mają węzły potomne wpisać do listy LIS, jako elementy typu A .
 - 2) Przegląd porządkujący:
 - 2.1) dla każdego elementu (i, j) z LIP rób:
 - 2.1.1) wyślij $S_n(i, j)$;
 - 2.1.2) jeśli $S_n(i, j) = 1$ to przenieś (i, j) do LSP i wyślij znak współczynnika $c_{i,j}$;
 - 2.2) dla każdego elementu (i, j) z LIS rób:
 - 2.2.1) dla elementu typu A rób:
 - wyślij $S_n(\mathbf{D}(i, j))$;
 - jeśli $S_n(\mathbf{D}(i, j)) = 1$ to
 - * dla każdego $(k, l) \in \mathbf{O}(i, j)$ rób:
 - wyślij $S_n(k, l)$;
 - jeśli $S_n(k, l) = 1$ dodaj (k, l) do LSP i wyślij znak współczynnika $c_{k,l}$;
 - jeśli $S_n(k, l) = 0$ dodaj (k, l) na koniec listy LIP;
 - * jeśli $\mathbf{L}(i, j) \neq 0$ przenieś (i, j) na koniec listy LIS jako element typu B i idź do punktu 2.2.2); jeśli jest pusty to usuń element (i, j) z listy LIS;
 - 2.2.2) jeśli element jest typu B to
 - wyślij $S_n(\mathbf{L}(i, j))$;
 - jeśli $S_n(\mathbf{L}(i, j)) = 1$ to
 - * dodaj każdy $(k, l) \in \mathbf{O}(i, j)$ na koniec listy LIS jako element typu A ;
 - * usuń (i, j) z listy LIS.
- 3) Przegląd uściślający: dla każdego elementu (i, j) z LSP, z wyłączeniem tych, które zostały przyłączone w ostatnim procesie sortowania (dla tego samego n) wyślij n -ty najbardziej znaczący bit wartości $|c_{i,j}|$;
- 4) Korekcja poziomu kwantyzacji: zmniejszenie wartości n o jeden i skok do punktu 2.

Algorytm dekodera jest identyczny poza jedną zasadniczą zmianą: słowa wyślij trzeba zastąpić słowem czytaj. Proces kodowania i dekodowania ma więc ten sam stopień złożoności. Oczywiście musi być wykonane także odświeżanie obrazu rekonstruowanego w ten sposób, że dla danego n wiadomo, że wartość bezwzględna współczynników znaczących jest z przedziału $2^n \leq |c_{i,j}| < 2^{n+1}$, stąd rekonstruowana wartość bezwzględna będzie wynosić $|\hat{c}_{i,j}| = 1.5 \cdot 2^n$. Podczas przeglądu uściślającego dekoderek dodaje lub odejmuje wartość 2^{n-1} do wartości $|\hat{c}_{i,j}|$.

W ostatniej fazie kodowania, dla zwiększenia skuteczności kompresji, można użyć koder arytmetyczny, z różnymi modelami kontekstów lub też rozszerzonymi alfabetami (nie binarnymi).

Charakterystyka algorytmu

Podstawową zaletą tej metody kompresji jest tworzenie zagnieżdżonego kodu, w pełni progresywnego, umożliwiającego efektywną transmisję obrazów w sieciach, a także odtwarzanie z kodu wielu postaci obrazu oryginalnego o różnym poziomie zniekształceń. Proces kodowania można zakończyć w każdej chwili, uzyskując żadaną długość kodu lub też założony poziom zniekształceń. Jest więc ten algorytm implementacją techniki EZW o dwóch zasadniczych usprawnieniach: pozwala uzyskać mniejsze wartości średnich bitowych przy tym samym poziomie zniekształceń, a także zwiększyć efektywność czasową wyraźnie minimalizując czas kompresji i dekompresji.

Taka skuteczność osiągnięta jest głównie poprzez wprowadzenie zasady budowania drzew zer nie na całym obszarze określoności współczynników waveletowych w każdym kroku iteracyjnym, lecz jedynie w tym obszarach drzew zer, które jeszcze zachowały się w kolejnych poprzednich krokach iteracji. Ponadto zrezygnowano z kilku-symbolowego opisu rozkładu drzew zer w przestrzeni danych po waveletowej dekompozycji, lecz wprowadzono binarny opis kolejnych decyzji dzielenia podzbiorów. Korzyść z tego jest jednak niewielka, o czym świadczy wprowadzanie większej liczby symboli w efektywniejszej wersji algorytmu SPIHT z kontekstowym kodowaniem arytmetycznym.

Rozsądnym rozwiązaniem wydaje się wprowadzenie tej pierwszej zasady do klasycznej postaci algorytmu EZW.