

Roman STAROSOLSKI  
Politechnika Śląska, Instytut Informatyki

## ALGORYTMY BEZSTRATNEJ KOMPRESJI OBRAZÓW

**Streszczenie.** Artykuł prezentuje przegląd algorytmów bezstratnej kompresji obrazów oraz standardów obowiązujących w tej dziedzinie. W artykule, po omówieniu podstawowych pojęć i metod stosowanych w bezstratnej kompresji obrazów, zawarto przegląd i klasyfikację algorytmów i standardów kompresji poszczególnych klas obrazów. Kilka algorytmów omówiono bardziej szczegółowo, w tym między innymi JPEG-LS — nowy międzynarodowy standard bezstratnej kompresji obrazów.

## LOSSLESS IMAGE COMPRESSION ALGORITHMS

**Summary.** The paper presents an overview of the lossless image compression algorithms and of the standards concerning lossless image compression. In the paper, after presenting basic methods used in lossless image compression, we review algorithms and standards of lossless compression for different classes of images. Selected algorithms are described more thoroughly, among them JPEG-LS — the new international standard of lossless image compression.

### 1. Wprowadzenie

Niniejszy artykuł zawiera przegląd metod, algorytmów i standardów stosowanych w bezstratnej kompresji obrazów. Problematyka omawiana w artykule jest bardzo obszerna. Z tego powodu omawiając algorytmy skoncentrowano się na przedstawieniu ich istoty, nie dążąc do uzyskania takiego poziomu szczegółowości, jak w niektórych pozycjach z cytowanej literatury. Omówienie podstawowych pojęć i algorytmów z zakresu kompresji danych, stosowanych w algorytmach kompresji obrazów, wykracza poza ramy niniejszego artykułu i jest przedmiotem oddzielnego opracowania [48]. Układ dalszej części niniejszego artykułu jest następujący: w punkcie 2 znajdują się definicje podstawowych pojęć i charakterystyki metod stosowanych w bezstratnej kompresji obrazów. Punkt 3 zawiera

przeгляд i klasyfikację algorytmów i standardów kompresji poszczególnych klas obrazów. W punkcie 4 omówiono bardziej szczegółowo wybrane algorytmy i standardy bezstratnej kompresji obrazów. Omówiono standard Lossless JPEG oraz algorytm FELICS, który dorównuje Lossless JPEG pod względem współczynników kompresji będąc jednocześnie algorytmem prostym i szybkim. Omówiono również nowy międzynarodowy standard bezstratnej kompresji obrazów, tj. JPEG-LS. Artykuł kończy krótkie podsumowanie.

## 2. Podstawowe pojęcia i metody

### 2.1. Obraz cyfrowy w kompresji danych

Obraz cyfrowy jest dyskretną reprezentacją naturalnego obrazu otrzymaną w wyniku procesu akwizycji bądź wygenerowaną w sposób sztuczny. Obraz cyfrowy to aproksymacja rzeczywistego obrazu za pomocą prostokątnej tablicy pikseli. Warto zaznaczyć, iż obrazy docierające do świadomości człowieka to też aproksymacje obrazów rzeczywiście pojawiających się na siatkówce oka, która składa się ze skończonej liczby czopków i pręcików. Piksel, jako element obrazu cyfrowego, reprezentuje odpowiadający mu fragment powierzchni obrazu rzeczywistego. Komputerową reprezentacją piksela jest wektor dyskretnych wartości, które opisują, z ograniczoną precyzją, wartości poszczególnych składowych odpowiadającego pikselowi fragmentu obrazu. Ze względu na liczbę składowych piksela oraz sposób ich reprezentacji i interpretacji obrazy cyfrowe dzielimy na klasy:

- obrazy binarne — piksel to bit, którego wartość określa czy dany fragment obrazu jest biały (1) czy czarny (0);
- obrazy w stopniach szarości — piksel jest nieujemną liczbą całkowitą, jej wartość interpretowana jest jako jasność obrazu. Zakres wartości piksela (liczba stopni szarości, liczba poziomów jasności) na ogół nie przekracza  $2^{16}$ . Przez obraz o głębi jasności  $N$  bitów na piksel rozumiemy obraz o liczbie stopni szarości  $2^N$ ;
- obrazy barwne — piksel to 3-elementowy lub 4-elementowy wektor określający pośrednio bądź bezpośrednio jasność, nasycenie oraz barwę elementu obrazu. Znaczenie i liczba poszczególnych składowych wektora zależne są od przyjętego modelu barwy, np. dla modelu RGB będzie to jasność składowych czerwonej, zielonej i niebieskiej, a dla modelu  $YCbCr$  jasność i para składowych pozwalająca określić barwę i nasycenie;
- obrazy  $k$ -spektralne — piksel to  $k$ -elementowy wektor, jego poszczególne składowe opisują jasność obrazu w różnych zakresach spektrum elektromagnetycznego (np. podczerwień, zakresy widzialne);

- obrazy z paletą barw — piksel to pojedyncza nieujemna liczba całkowita interpretowana jako numer barwy z pewnej palety barw. Dla obrazów z paletą barw przyjmuje się, iż nie ma bezpośredniego związku pomiędzy wartością piksela a jasnością nasyceniem ani barwą elementu obrazu.

Zakłada się, iż w obrazach w stopniach szarości, obrazach barwnych oraz  $k$ -spektralnych występuje silna korelacja wartości blisko położonych pikseli.

Algorytmy kompresji ogólnego przeznaczenia zastosowane bezpośrednio do kompresji danych obrazowych w praktyce nie sprawdzają się. W kompresji obrazów korzysta się z nich, ale po przeprowadzeniu transformacji danych obrazowych do postaci lepiej nadającej się do kompresji algorytmem uniwersalnym. Przyczyn takiego stanu rzeczy można dopatrywać się w tym [27], iż:

- obraz jest dwuwymiarowym zbiorem danych. Algorytmy kompresji ogólnego przeznaczenia projektowane były dla jednowymiarowych ciągów symboli, takich jak na przykład tekst;
- rozmiar alfabetu źródła dla obrazów jest duży (typowo 256 do 65536), co komplikuje stosowanie modeli danych wyższych rzędów w przypadku zastosowania kompresora statystycznego;
- naturalne obrazy zawierają szum, wprowadzony między innymi w procesie akwizycji, który w przypadku algorytmów słownikowych powoduje, że udział dłuższych fraz w słowniku jest niewielki;
- faktu, że dla większości pikseli typowego obrazu jasność piksela jest zbliżona do pikseli go otaczających, nie wykorzystuje się bezpośrednio w algorytmach kompresji ogólnego przeznaczenia.

## 2.2. Kompresja bezstratna a kompresja stratna

Działanie algorytmów stratnych nie jest odwracalne i dlatego ich zastosowanie ograniczone jest do przypadków, gdy dopuszczalne są niewielkie różnice pomiędzy tym, co kompresowaliśmy, a tym, co uzyskaliśmy po dekompresji. Nadają się one dla takich klas danych, jak obrazy czy dźwięki, gdzie utrata informacji, dzięki niedoskonałości ludzkich zmysłów, pozostaje niezauważona bądź może być tolerowana. Algorytmy stratne pozwalają na uzyskiwanie znacznie lepszych współczynników kompresji niż w algorytmach bezstratnych. W przypadku algorytmów stratnych możliwe jest również uzyskanie zadanego współczynnika kompresji bez względu na postać obrazu (co oczywiście czasami prowadzi do znacznej utraty informacji). Znanych jest wiele stratnych algorytmów kompresji obrazów, kompresja stratna stosowana jest często dla obrazów 'ruchomych, przegląd algorytmów można znaleźć w monografiach [38,39].

Algorytmy bezstratne stosuje się w kompresji obrazów wtedy, gdy obraz jest traktowany jako dokument (ustawodawstwo niektórych państw zabrania kompresji stratnej obrazów medycznych używanych w celach diagnostycznych [7, 38]), gdy obraz będzie podlegał dalszemu przetwarzaniu lub gdy nie można zakładać, że dopuszczalne jest zastosowanie kompresji stratnej. Kompresja bezstratna obrazów stosowana jest również w przypadku takich obrazów, których akwizycja jest kosztowna [42, 6].

W algorytmach LPL (ang. Lossy Plus Lossless) stosuje się kompresję stratną jako element kompresora bezstratnego. Pierwsze prace nad takimi algorytmami prowadził na początku lat 80. S.E. Elnahas [26]. W przypadku algorytmów LPL [51, 54, 16] obraz najpierw kompresuje się stratnie, uzyskując silnie skompresowaną „miniaturę” oryginalnych danych, którą można zastosować np. do szybkiego przeglądania archiwów obrazów. Następnie wyznacza się residuum, tj. różnice pomiędzy oryginalnymi danymi a tymi zdekompresowanymi po kompresji stratnej. Residuum kompresuje się algorytmem bezstratnym, przy czym dla lepszej kompresji residuum można wykorzystać informacje o charakterystyce obrazu uzyskane z danych skompresowanych stratnie. Na podstawie danych skompresowanych stratnie i bezstratnie skompresowanego residuum można odtworzyć oryginalne dane bez utraty informacji.

### 2.3. Linearyzacja

W bezstratnej kompresji obrazów na ogół stosuje się algorytmy kompresji ogólnego przeznaczenia. Algorytmy te przeznaczone są do kodowania ciągów symboli, dlatego też dwuwymiarowe dane obrazowe transformowane są do postaci jednowymiarowego ciągu pikseli. Transformacja ta nazywana jest linearyzacją.

Ze względu między innymi na uporządkowanie pikseli stosowane w transmisji z urządzeń akwizycji obrazu oraz w pamięci sterowników graficznych współczesnych komputerów najczęściej spotykany schemat linearyzacji to linearyzacja w porządku rastrowym. Polega ona na uporządkowaniu pikseli obrazu w następujący sposób: najpierw umieszczane są kolejne piksele górnego wiersza obrazu, poczynając od lewej krawędzi obrazu, następnie kolejne coraz niżej położone wiersze, również przeglądane od lewej do prawej.

### 2.4. Proste metody kompresji obrazu

Do kompresji obrazów, w których udział jednorodnych fragmentów jest znaczny a liczba barw niewielka, stosowano proste metody kompresji, takie jak [65]:

- Kodowanie długości sekwencji identycznych pikseli, po przeprowadzeniu linearyzacji obrazu.
- Kodowanie konturowe, w którym obraz podlega rozbiciu na tzw. kontury, czyli spójne obszary składające się z jednakowych pikseli. Opis danego konturu zawiera infor-

macje o barwie pikseli konturu, położeniu konturu w obrazie oraz kształcie konturu (zadany przez krawędź konturu opisaną tzw. kodem łańcuchowym).

- Kodowanie drzewiaste. W metodzie tej obraz reprezentowany jest przez drzewo czwórkowe. Korzeń drzewa reprezentuje cały obraz. Dany węzeł odpowiada prostokątnemu fragmentowi obrazu, jeżeli wszystkie piksele tego fragmentu są identyczne, to węzeł jest liściem, w którym pamięta się barwę pikseli. W przeciwnym przypadku węzeł ma 4 synów, każdy z nich opisuje dokładnie jedną ćwiartkę obrazu odpowiadającego temu węzłowi.

Ciąg symboli opisujący obraz zakodowany z użyciem powyższych metod charakteryzuje się zazwyczaj nadmiarowością, a co za tym idzie, nadaje się do dalszej kompresji za pomocą algorytmów statystycznych bądź słownikowych.

## 2.5. Kompresja progresywna

W niektórych zastosowaniach, takich jak [38] telekonferencje, telekonsultacje i wizualne przeszukiwanie dużych baz obrazowych stosuje się progresywne kodowanie obrazów. W kanałach transmisyjnych o małej przepustowości transmisja dużych obrazów, nawet jeżeli są skompresowane, jest czasochłonna. Odbiorcy obrazu może wystarczyć przybliżona znajomość obrazu, by podjąć decyzję o odrzuceniu obrazu i zrezygnowaniu z transmisji całości. Dlatego najpierw transmituje się przybliżoną i pozbawioną szczegółów wersję obrazu, a następnie dane pozwalające na odtwarzanie obrazu z coraz większą dokładnością, aż do wiernego jego odtworzenia.

Realizacja kompresji i transmisji progresywnej przebiega w następujący sposób: obraz zredukowany jest do obrazu o mniejszych rozmiarach, operacja redukcji może być przeprowadzana kilkakrotnie. W ten sposób z oryginalnego obrazu otrzymujemy kilka coraz mniej dokładnych jego przybliżeń o coraz mniejszych rozdzielczościach. Następnie kompresujemy i transmitujemy kolejne obrazy, poczynając od obrazu o najmniejszej rozdzielczości. Odbiorca na bieżąco dekompresuje otrzymywane dane i podejmuje decyzje o kontynuowaniu bądź przerwaniu transmisji. Jeżeli odbiorca nie przerwie transmisji, to kompresowane i transmitowane są kolejne, coraz dokładniejsze przybliżenia obrazu i ostatecznie oryginalny obraz.

Zastosowanie transmisji progresywnej nie musi prowadzić do uzyskania gorszych współczynników kompresji niż kodowanie i transmisja jedynie oryginalnego obrazu. Kodując dane przybliżenie obrazu kompresor ma do dyspozycji informacje o charakterystyce obrazu uzyskane z już przetransmitowanych, mniej dokładnych przybliżeń obrazu, co pozwala na polepszenie współczynników kompresji. Przykładami tutaj mogą być kompresory ABIC i PABIC [38], czy FELICS i Progressive FELICS [15]. Wśród algorytmów progresywnych można ponadto wymienić algorytmy, takie jak: kompresor opisany przez Wanga i Morrisa [56], MLP (ang. Multi-Level Progressive) [12, 13], algorytm Saida i Pearlmana [41, 42], MEP (ang. Mi-

nimal Entropy Pyramid) [10], a także standardy GIF, PNG, JPEG i JPEG2000. W literaturze w odniesieniu do kompresji oraz transmisji progresywnej realizowanej tak, jak to opisano, stosowane jest również określenie „hierarchiczny” [40, 49].

## 2.6. Kodowanie predykcyjne i transformacyjne

Ze względu na wymienione w punkcie 2.1 cechy obrazów bezpośrednia kompresja pikseli obrazu algorytmem ogólnego przeznaczenia nie daje zadowalających wyników. W kodowaniu predykcyjnym transmituje się do odbiorcy błędy predykcji pikseli a nie piksele obrazu. Za pomocą funkcji zwanej predyktorem wyznacza się spodziewaną wartość danego piksela, np. jasność dla obrazu w stopniach szarości. Ciąg błędów predykcji kodowany i transmitowany do odbiorcy określany jest, tak jak w przypadku algorytmów LPL, mianem residuum. Dla typowych obrazów entropia rzędu zerowego residuum jest znacznie mniejsza od entropii rzędu zerowego pikseli obrazu. Proces wyznaczania residuum nazywamy dekorelacją. Zastosowanie dekorelacji pozwala na konstrukcję prostych i szybkich algorytmów kompresji obrazów (np. [1, 9]). Użycie nawet bardzo prostych predyktorów, takich jak na przykład predyktor, którego wartość dla danego piksela równa jest wartości lewego sąsiada tego piksela, pozwala na osiągnięcie znacznie lepszych współczynników kompresji niż bez przeprowadzania dekorelacji. Dla naturalnych obrazów w stopniach szarości oraz dla poszczególnych składowych obrazów barwnych rozkład prawdopodobieństwa symboli residuum jest dobrze aproksymowany przez rozkład Laplace’a (symetryczny rozkład wykładniczy) o wartości oczekiwanej bliskiej 0 [64, 28, 45, 6, 29, 30]. W kodowaniu residuum algorytmy statystyczne zazwyczaj sprawdzają się lepiej niż algorytmy słownikowe [21, 44, 46, 47]. Dlatego też w większości przypadków residuum kompresuje się z użyciem algorytmu statystycznego. W przeciwieństwie jednak do algorytmów kompresji statystycznej ogólnego przeznaczenia, w kompresji obrazów kontekst danego symbolu często nie składa się jedynie z symboli bezpośrednio go poprzedzających.

Odmianą metodą przekształcenia danych obrazowych do postaci lepiej nadającej się do kompresji algorytmem ogólnego przeznaczenia jest zastosowanie dwuwymiarowych transformacji, takich jak np. dyskretna transformata kosinusowa. W metodach transformacyjnych kodowaniu i transmisji, zamiast pikseli obrazu, podlega macierz współczynników transformaty przeprowadzonej na obrazie w całości bądź po dekompozycji obrazu na fragmenty. Metoda ta stosowana jest przede wszystkim w stratnych algorytmach kompresji obrazów, gdzie nie jest wymagane istnienie transformaty odwrotnej rekonstruującej wiernie obraz na podstawie macierzy współczynników kodowanych ze skończoną precyzją. W stratnych algorytmach kompresji obrazu macierz współczynników transformaty kodowana jest po przeprowadzeniu kwantyzacji.

### 3. Obszary zastosowań kompresji obrazów

#### 3.1. Obrazy binarne

Obszar zastosowania obrazów binarnych to przede wszystkim archiwizacja dokumentów oraz transmisja przez urządzenia faksowe. W przypadku tego typu obrazów kompresja pozwala na kilkukrotne bądź nawet kilkudziesięciokrotne zmniejszenie rozmiarów plików zawierających obrazy. Zastosowanie kompresji do transmisji faksowej było przyczyną wczesnego pojawienia się standardów bezstratnej kompresji obrazów binarnych. Stosowany obecnie standard CCITT Group 3 wprowadzono w roku 1985. Pomimo iż od tego czasu opublikowano wiele nowych algorytmów i wprowadzono nowe standardy, CCITT Group 3 jest nadal stosowany ze względu na zgodność ze starszymi urządzeniami faksowymi.

Algorytm kompresji zawarty w standardzie CCITT Group 3, to kompresor stosujący predefiniowane kody przedrostkowe stałej długości do kodowania długości sekwencji białych i czarnych pikseli obrazu przeglądanego w porządku rastrowym. Kolejny standard CCITT Group 4 to rozwinięcie CCITT Group 3, uwzględniające dwuwymiarowy charakter danych obrazowych. Kodowanie danego wiersza obrazu z uwzględnieniem postaci poprzedniego wiersza pozwala na poprawę współczynników kompresji. Rozwiązanie takie niesie ze sobą niebezpieczeństwo propagacji błędu na kolejne wiersze obrazu po wystąpieniu błędu transmisji, gdyż w transmisji faksowej kanał transmisyjny to kanał z szumem. Propagację błędów ogranicza się kodując co kilka wierszy jeden wiersz algorytmem CCITT Group 3.

Lepsze współczynniki kompresji niż przy użyciu kodów przedrostkowych można uzyskać stosując kodowanie arytmetyczne. Opublikowany w roku 1988 algorytm ABIC [38] pozwala na uzyskanie współczynników kompresji lepszych o co najmniej 20 % od algorytmów CCITT Group 3 i 4. W algorytmie tym piksele obrazu przeglądanego w porządku rastrowym kodowane są arytmetycznie. Kontekstem danego piksela jest 7 już przetworzonych pikseli aktualnego i poprzedniego wiersza położonych najbliżej tego piksela.

W standardzie JBIG [20, 43] stosuje się adaptacyjną kompresję arytmetyczną. JBIG stosowany jest przede wszystkim w składowaniu i transmisji skanowanych dokumentów tekstowych. Przewaga kompresora JBIG nad CCITT Group 3 i 4 widoczna jest zwłaszcza w przypadku [20] komputerowo generowanych obrazów zawierających tekst oraz binaryzowanych z zastosowaniem technik ditheringu i halftoningu obrazów barwnych i obrazów w stopniach szarości. Nowszy standard JBIG2 [19] to kompresor, w którym, aby polepszyć uzyskiwane współczynniki kompresji, przeprowadza się wstępną analizę obrazu, dekomponując go na fragmenty zawierające np. tekst czy binaryzowane fotografie, po czym koduje się w odmienny sposób fragmenty różnych typów.

### 3.2. Obrazy w stopniach szarości

Prawdopodobnie najwięcej bezstratnych algorytmów kompresji obrazów to kompresory obrazów w stopniach szarości. Duże zainteresowanie tą klasą obrazów wynika z tego, że algorytmy te stanowią bazę dla algorytmów kompresji obrazów  $k$ -spektralnych i barwnych. Algorytmy te znajdują zastosowanie nie tylko w kompresji naturalnych obrazów, ale również obrazów medycznych i innych 2-wymiarowych zbiorów danych pomiarowych, w których występuje silna korelacja wartości blisko położonych symboli.

Do kompresji obrazów w stopniach szarości stosuje się algorytmy zaprojektowane specjalnie dla tej klasy obrazów, jak również algorytmy kompresji obrazów binarnych. Aby zastosować kompresor obrazów binarnych należy dokonać dekompozycji obrazu o głębi jasności  $N$  bitów na piksel na  $N$  płaszczyzn bitowych traktowanych jako obrazy binarne. Binarne reprezentacje zbliżonych wartości liczbowych mogą znacznie się różnić (np. liczby binarne 01111 i 10000). Aby po dekompozycji obrazu piksele o zbliżonej jasności miały taką samą wartość w większości płaszczyzn bitowych, przed dekompozycją koduje się piksele za pomocą kodu Graya. Kod Graya to binarny kod stałej długości, w którym kolejne słowa kodowe różnią się dokładnie jednym bitem. W przypadku kompresji obrazów o małej głębi jasności (poniżej 6 bitów na piksel) algorytmy kompresji obrazów binarnych mogą uzyskać współczynniki kompresji lepsze od algorytmów projektowanych dla obrazów w stopniach szarości (np. algorytmy JBIG i Lossless JPEG [2]).

Istnieje kilka standardów kompresji obrazów tej klasy, między innymi znane są standardy opracowane przez komitet JPEG: Lossless JPEG (punkt 4.1) i JPEG-LS (punkt 4.3). Najnowszym standardem dotyczącym tej dziedziny, również wprowadzonym przez komitet JPEG, jest JPEG2000 [5, 17, 25]. JPEG2000 ma wiele cech [36], dzięki którym jest bardziej uniwersalny niż np. JPEG-LS. Standard JPEG2000 definiuje między innymi progresywny schemat kodowania i transmisji, jak również stratne i bezstratne transformacyjne (transformata falkowa) metody kompresji obrazów w stopniach szarości oraz obrazów barwnych. Jednak bezstratny algorytm kompresji zawarty w tym standardzie ustępuje kompresorowi JPEG-LS zarówno pod względem prędkości, jak i uzyskiwanych współczynników kompresji [35].

Za najsilniejszy algorytm, uzyskujący średnio najlepsze współczynniki kompresji, uważa się algorytm CALIC, którego autorami są X. Wu i N. Memon [62, 63, 57]. W algorytmie tym stosuje się schemat predykcyjny, w którym po linearyzacji w porządku rastrowym, residuum kompresuje się algorytmem statystycznym. Algorytm CALIC zawiera dwa tryby działania. Tak zwany tryb binarny stosowany jest wtedy, gdy już zakodowane otoczenie danego piksela składa się z pikseli o co najwyżej dwóch poziomach jasności. W trybie binarnym koduje się informacje, czy piksel ma jasność taką, jak jaśniejsze piksele z sąsiedztwa, taką jak ciemniejsze, czy też inną – w takim przypadku następuje wyjście z trybu binarnego. Podsta-



wowy, predykcyjny tryb działania algorytmu CALIC polega na wykorzystaniu nieliniowego predyktora GAP (ang. Gradient Adjusted Predictor). Predyktor ten został skonstruowany tak, by poprawnie rozpoznawać krawędzie występujące w obrazie. Do wyznaczenia wartości predyktora używa się siedmiu wcześniej przetworzonych pikseli z aktualnego wiersza obrazu i dwóch wierszy poprzednich. Przed kodowaniem błędu predykcji przeprowadza się korektę średniej wartości błędu predykcji (metoda ta stosowana również w algorytmie JPEG-LS opisana jest w punkcie 4.3). Kontekst dla symbolu residuum wyznacza się zarówno na podstawie już przetworzonych pikseli otaczających dany piksel, jak i odpowiadających im symboli residuum. Dzięki zastosowaniu grupowania kontekstów w kubelki (podobnie jak w Lossless JPEG – punkt 4.1) ograniczono złożoność modelu danych.

### 3.3. Obrazy $k$ -spektralne i obrazy barwne

W przypadku obrazów  $k$ -spektralnych, jak również niektórych modeli barw obrazów barwnych, takich jak np. RGB (będącego obrazem 3-spektralnym), poszczególne składowe obrazu są na ogół silnie skorelowane. Jeżeli poprzez transformację obrazu do innego modelu uzyskujemy nowy, znacznie mniej nadmiarowy zbiór składowych pikseli, to taką transformację nazywamy dekorrelacją składowych. Do dekorrelacji składowych wykorzystuje się na przykład transformację do modelu barw YUV lub transformatę KL [64].

Jak wcześniej wspomniano, algorytmy kompresji obrazów  $k$ -spektralnych i barwnych konstruowane są na podstawie algorytmów kompresji obrazów w stopniach szarości. Przykładem mogą być algorytmy CALIC i IB-CALIC [55]. Często stosowana metoda polega na dekompozycji obrazu barwnego lub  $k$ -spektralnego na odpowiednią liczbę kodowanych niezależnie składowych. Do kodowania składowych stosuje się algorytm kompresji obrazu w stopniach szarości. Metoda ta, stosowana np. w JPEG-LS, jest prostsza niż analizowanie informacji o zależnościach pomiędzy poszczególnymi składowymi obrazu, co jednak wpływa negatywnie na uzyskiwane współczynniki kompresji. Na przykład algorytm SICLIC [3], oparty na JPEG-LS, wykorzystujący przy kodowaniu kolejnych składowych informacje o już zakodowanych składowych, uzyskuje współczynniki kompresji o kilkanaście procent lepsze od JPEG-LS. Za rozwiązanie pośrednie można uznać metodę, zastosowaną np. w standardzie JPEG2000, w której po przeprowadzeniu dekorrelacji składowych poszczególne składowe koduje się niezależnie.

### 3.4. Obrazy z paletą barw

Dla tej klasy obrazów praktyczne zastosowanie znalazły algorytmy oparte na kodowaniu słownikowym, które dla obrazów w stopniach szarości, obrazów barwnych i  $k$ -spektralnych

były mniej efektywne od kompresorów statystycznych. Obrazy tej klasy wykorzystywane są między innymi jako elementy wzbogacające strony WWW.

W standardzie GIF (ang. Graphics Interchange Format) stosuje się kodowanie LZW. Standard ten początkowo był wewnętrznym standardem opracowanym przez sieć CompuServe dla potrzeb transmisji obrazów poprzez modem, upowszechnił się jednak wraz z rozwojem sieci Internet. Format pliku GIF opisuje sposób kodowania palety barw i pikseli obrazu dla obrazów zawierających nie więcej niż 256 barw. Standard ten definiuje również bardzo prosty, progresywny schemat kompresji obrazu, który polega na linearyzacji obrazu w porządku z przeplotem. W takim przypadku kodowaniu i transmisji podlega najpierw co 8. wiersz obrazu poczynając od wiersza 0, po czym co 8. wiersz poczynając od wiersza 4, następnie co 4. wiersz poczynając od 2. wiersza i ostatecznie co 2. wiersz poczynając od wiersza 1. Standard GIF stosowany jest również do zapisu krótkich animacji, gdyż pojedynczy plik może zawierać opis więcej niż jednego obrazu. W takim przypadku kolejne obrazy prezentowane odbiorcy powinny być kolejno, w odstępach czasowych określonych w pliku zawierającym obraz zakodowany zgodnie ze standardem GIF.

Następcą standardu GIF jest PNG (ang. Portable Network Graphics) [52]. Jest interesujące, że na powstaniu nowego standardu zaważył fakt, że algorytm kompresji LZW wykorzystywany w GIF jest opatentowany i używanie go wymaga wykupienia licencji od firmy CompuServe. Kompresor PNG jest schematem predykcyjnym. W PNG stosuje się kodowanie słownikowe LZ77, przy czym do kodowania położenia i długości dopasowanej sekwencji używa się kodów przedrostkowych zmiennej długości (predefiniowanych, bądź kodów Huffmana z ograniczeniem 15 na długość słowa kodowego [39]). Standard PNG jest bardziej uniwersalny niż GIF. Oprócz kompresora obrazów z paletą barw oraz doskonalszego niż w algorytmie GIF progresywnego trybu transmisji i kodowania PNG definiuje również algorytmy kompresji obrazów barwnych i obrazów w stopniach szarości o głębi jasności do 16 bitów na piksel. Jednak w większości przypadków dla obrazów barwnych oraz obrazów w stopniach szarości współczynniki kompresji uzyskiwane przez algorytm kompresji zawarty w standardzie PNG nie dorównują współczynnikom uzyskiwanym przez JPEG-LS [37].

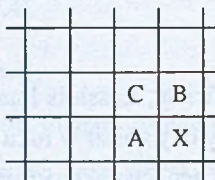
## 4. Wybrane algorytmy bezstratnej kompresji obrazów

### 4.1. Kompresja bezstratna w standardzie JPEG

W roku 1986 z inicjatywy organizacji ISO, CCITT oraz IEC powstał komitet JPEG (ang. Joint Photographic Experts Group), którego zadaniem było opracowanie międzynarodowego standardu kompresora i dekompresora naturalnych obrazów w stopniach szarości oraz

obrazów barwnych. Plonem kilkuletniej pracy komitetu jest standard JPEG [53, 39, 8], który definiuje stratną metodę kompresji opartą na transformacie kosinusowej, predykcyjną metodę kompresji bezstratnej określaną mianem Lossless JPEG, progresywne algorytmy kodowania i transmisji oraz format pliku zawierającego skompresowane dane obrazowe.

Algorytm Lossless JPEG [22] to zaadaptowany do potrzeb standardu JPEG algorytm Sunset, bezstratny algorytm kompresji obrazów opracowany w latach 80. przez G. G. Langdona, J. Rissanena i S. Todda. W algorytmie Lossless JPEG stosuje się predykcyjny schemat kodowania przeznaczony dla obrazów o głębi jasności 2-16 bitów na piksel. W schemacie tym, po przeprowadzeniu linearyzacji w porządku rastrowym i dekorelacji, residuum kompresowane jest algorytmem statystycznym. Residuum wyznaczane jest z użyciem jednego z ośmiu dostępnych predyktorów liniowych, po czym kompresowane jest statystycznie. Dostępne w algorytmie Lossless JPEG predyktory zebrano w tab. 1. Rysunek 1 ilustruje położenie pikseli, których wartości używane są do wyznaczania wartości funkcji predyktora. Poszczególne predyktory są odpowiednie dla różnych elementów występujących w obrazach, np. dla pionowych krawędzi czy powierzchni o łagodnie zmieniającej się jasności. Spośród predyktorów algorytmu Lossless JPEG za predyktor pozwalający dla typowych obrazów uzyskać średnio najlepsze współczynniki kompresji uważa się predyktor *Pred7*. W ogólnym przypadku predyktor można by dobierać adaptacyjnie — wybierać ten, który dla już przetworzonych pikseli był najlepszy. Można by również wybrać najlepszy predyktor dla części obrazu, np. dla wiersza, po czym numer predyktora przekazać przed skompresowanym fragmentem obrazu do algorytmu dekompresji. Zastosowanie obu metod zwiększyłoby złożoność algorytmu kompresji i dlatego wybrany predyktor stosowany jest do dekorelacji całego obrazu. Jednak, gdy dla danego piksela wartość wybranego predyktora nie może być wyznaczona, stosuje się prostsze predyktory (dla piksela w lewym górnym rogu obrazu używa się predyktora *Pred0*, dla pozostałych pikseli pierwszego wiersza stosuje się predyktor *Pred1*, a dla pozostałych pikseli pierwszej kolumny *Pred2*).



Rys. 1. Położenie pikseli używanych w predyktorach algorytmu Lossless JPEG  
 Fig. 1. Location of pixels used in the Lossless JPEG predictors

Tabela 1

## Predyktory algorytmu Lossless JPEG

$Pred0(X) = 0$	$Pred4(X) = A + B - C$
$Pred1(X) = A$	$Pred5(X) = A + (B - C) / 2$
$Pred2(X) = B$	$Pred6(X) = B + (A - C) / 2$
$Pred3(X) = C$	$Pred7(X) = (A + B) / 2$

Standard JPEG do kodowania residuum przewiduje zastosowanie adaptacyjnego kompresora statystycznego z arytmetycznym koderem entropijnym i modelem danych drugiego rzędu bądź kompresora statystycznego stosującego kody przedrostkowe zmiennej długości (predefiniowane, bądź wyznaczone dla danego obrazu algorytmem Huffmana) i model danych rzędu zerowego.

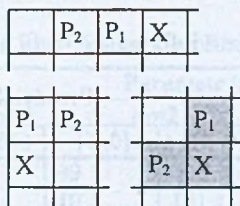
Przejęty po algorytmie Sunset model danych kompresora statystycznego stosowany w przypadku kodowania arytmetycznego to pierwszy [23] model danych, w którym zastosowano grupowanie kontekstów w tzw. kubelki. Kontekstem dla danego symbolu residuum są symbole residuum położone na lewo i powyżej danego symbolu. Liczba kontekstów, dla których trzeba by gromadzić charakterystyki symboli pojawiających się w tych kontekstach, to  $\|S\|^o$ , gdzie  $\|S\|$  jest rozmiarem alfabetu (typowo kilkaset symboli), a  $o$  – rzędem modelu danych (dla algorytmów Sunset i Lossless JPEG  $o = 2$ ). Aby zmniejszyć złożoność pamięciową modelu danych, konteksty zgrupowano w tzw. kubełkach kontekstów. Idea tego rozwiązania polega na gromadzeniu informacji o charakterystyce kompresowanych danych dla wszystkich kontekstów zgrupowanych w kubełku łącznie, dzięki czemu liczba przechowywanych w modelu danych charakterystyk równa jest liczbie kubełków i może być wielokrotnie mniejsza od liczby kontekstów. W Lossless JPEG rozmieszczenie kontekstów w kubełkach jest z góry określone, a liczba kubełków kontekstów jest niezależna od rozmiaru alfabetu i wynosi 25.

## 4.2. Algorytm FELICS

### 4.2.1. Ogólny opis algorytmu

Algorytm FELICS (ang. Fast, Efficient, Lossless Image Compression System) autorstwa P.G. Howarda i J.S. Vittera przedstawiony został w roku 1993 [14, 11]. W porównaniu z algorytmem Lossless JPEG, algorytm ten charakteryzuje się zbliżonymi współczynnikami kompresji i istotnie większą prędkością działania. Algorytm FELICS przeznaczony jest do kodowania naturalnych obrazów w stopniach szarości.

W algorytmie tym obraz kodowany jest po przeprowadzeniu lineryzacji w porządku rastrowym. Sposób kodowania danego piksela  $X$  zależy od relacji jasności tego piksela do jasności jego dwóch najbliższych, już przetworzonych sąsiadów – pikseli  $P_1$  i  $P_2$  (rys. 2).



Rys. 2. Położenie najbliższych sąsiadów piksela wewnątrz i przy krawędziach obrazu  
 Fig. 2. Nearest neighbors of pixel in the center of the image and along the edges

Z wyjątkiem lewej i górnej krawędzi obrazu sąsiadami tymi są piksele położone powyżej i na lewo od danego. Dla górnej krawędzi obrazu są to 2 piksele położone na lewo od danego, a dla lewej krawędzi obrazu piksel położony powyżej i jego prawy sąsiad.

Autorzy algorytmu zaobserwowali, iż dla typowych obrazów jasność około połowy pikseli mieści się w przedziale o granicach wyznaczonych przez jasności dwóch najbliższych sąsiadów. Oznaczmy granicę dolną tego przedziału przez  $F_L$  (minimum jasności  $P_1$  i  $P_2$ ), zaś górną przez  $F_H$ . Jasności pikseli znajdujące się w przedziale  $[F_L, F_H]$  są rozłożone w przybliżeniu równomiernie, przy czym nieznacznie częściej występują wartości bliższe środkowi przedziału. Rozkład prawdopodobieństwa jasności symboli znajdujących się poza przedziałem maleje wraz z oddalaniem się od granicy przedziału; jest to w przybliżeniu rozkład wykładniczy.

Kodując dany piksel wyprowadza się pojedynczy bit określający czy jasność piksela leży w przedziale  $[F_L, F_H]$ . Jeżeli jasność piksela znajduje się wewnątrz przedziału, to odległość jasności tego piksela od początku przedziału koduje się z użyciem zmodyfikowanego kodu binarnego (punkt 4.2.2). W przeciwnym przypadku wyprowadza się kolejny bit określający czy jasność piksela znajduje się poniżej, czy powyżej przedziału, po czym koduje się odległość jasności piksela od bliższego krańca przedziału. Kodowanie odbywa się z użyciem parametrycznej rodziny kodów Golomba lub Rice'a (punkt 4.2.3) oraz adaptacyjnego modelu danych pierwszego rzędu (punkt 4.2.4).

#### 4.2.2. Zmodyfikowany kod binarny

Postać zmodyfikowanego kodu binarnego nieujemnej liczby całkowitej  $i$  zależy od założonego przedziału wartości, dla którego generujemy ten kod. Zmodyfikowany kod binarny użyty do kodowania liczb z przedziału  $[0, j - 1]$  przyporządkowuje części liczb z tego przedziału kody o długości  $\lfloor \log_2(j) \rfloor$  bitów, pozostałe liczby kodowane są na  $\lceil \log_2(j) \rceil$  bitach. Zmodyfikowany kod binarny staje się kodem binarnym stałej długości, gdy  $j$  jest potęgą dwójki. Przykłady zmodyfikowanych kodów binarnych zamieszczono w tab. 2.

Tabela 2

Zmodyfikowane kody binarne

Liczba	Przedział			
	[0, 4]	[0, 5]	[0, 6]	[0, 7]
$i=0$	00	00	00	000
$i=1$	01	01	010	001
$i=2$	10	100	011	010
$i=3$	110	101	100	011
$i=4$	111	110	101	100
$i=5$		111	110	101
$i=6$			111	110
$i=7$				111

Przyjmijmy  $N = \lceil \log_2(j) \rceil$  i  $n = 2^N$  ( $2^{N-1} < j \leq n$ ). Jeżeli  $i < n - j$ , to zmodyfikowany kod binarny  $i$  jest identyczny z  $N-1$ -bitowym kodem binarnym stałej długości  $i$ . W przeciwnym przypadku zmodyfikowany kod binarny  $i$  to  $N$ -bitowy binarny kod stałej długości liczby  $i + n - j$ . Liczba słów kodowych o długości  $N$  jest w zmodyfikowanym kodzie binarnym zawsze parzysta.

W algorytmie FELICS wśród jasności pikseli znajdujących się w przedziale  $[F_L, F_H]$  częściej występują wartości bliższe środka przedziału. Dlatego też różnica pomiędzy jasnością piksela a  $F_L$  (granica dolna przedziału  $[F_L, F_H]$ ) kodowana jest po przeprowadzeniu zmiany uporządkowania słów kodowych w zmodyfikowanym kodzie binarnym tak, aby krótsze słowa kodowe używane były do kodowania wartości położonych bliżej środka przedziału, a dłuższe dla pozostałych.

#### 4.2.3. Kody Golomba i Rice'a

Kody Golomba i kody Rice'a to parametryczne rodziny kodów przedrostkowych przeznaczone do kodowania nieujemnych liczb całkowitych. Kody z tych rodzin są optymalne w klasie kodów przedrostkowych dla wykładniczego rozkładu prawdopodobieństwa (dla niektórych wartości parametru rozkładu).

W tabeli 3 przedstawiono kilka pierwszych słów kodowych przykładowych kodów z rodzin Golomba (parametr  $m$ ) i Rice'a (parametr  $k$ ). Występujący w tabeli znak „•” nie jest częścią słowa kodowego, umieszczono go, by rozdzielić słowo kodowe na części kodowane z użyciem kodu unarnego i zmodyfikowanego kodu binarnego.

Opisana w roku 1966 przez S.W. Golomba rodzina kodów jest nieskończoną rodziną charakteryzowaną przez pojedynczy parametr [58]. Oznaczany przez  $m$  parametr kodu Golomba jest całkowitą liczbą dodatnią. Aby zakodować nieujemną, całkowitą liczbę  $x$  kodem Golomba z parametrem  $m$ , najpierw kodujemy  $\lfloor x/m \rfloor$  unarnie, po czym kodujemy  $x \bmod m$  za pomocą zmodyfikowanego kodu binarnego dla przedziału  $[0, m - 1]$ .

Tabela 3

## Kody Golomba i Rice'a

Liczba	Parametr kodu			
	$m=1$ $k=0$	$m=2$ $k=1$	$m=3$	$m=4$ $k=2$
$x=0$	0•	0•0	0•0	0•00
$x=1$	10•	0•1	0•10	0•01
$x=2$	110•	10•0	0•11	0•10
$x=3$	1110•	10•1	10•0	0•11
$x=4$	11110•	110•0	10•10	10•00
$x=5$	111110•	110•1	10•11	10•01
$x=6$	1111110•	1110•0	110•0	10•10
$x=7$	11111110•	1110•1	110•10	10•11

Rodzina kodów określana tutaj mianem rodziny Rice'a to podzbiór rodziny Golomba odkryty niezależnie przez R.F. Rice'a. Parametrem kodów Rice'a jest nieujemna całkowita liczba  $k$ . Kody Rice'a to szczególny przypadek kodów Golomba, gdy  $m = 2^k$ . Kody te są łatwiejsze do generacji, gdyż operację dzielenia można zastąpić przez przesunięcie o  $k$  bitów w prawo. Zamiast dzielenia modulo i kodowania zmodyfikowanym kodem binarnym można bezpośrednio (binarnie) zakodować  $k$  najmniej znaczących bitów  $x$ . Kody Rice'a stosowane są również do kodowania symboli residuum po przeprowadzeniu dekorrelacji obrazu (np. w algorytmie JPEG-LS), do kodowania współczynników transformacji w kodowaniu transformacyjnym obrazów [32, 24] oraz do kodowania danych z instrumentów pomiarowych satelitów i sond kosmicznych [34, 50, 4, 33].

#### 4.2.4. Model danych algorytmu FELICS

Model danych wykorzystywany jest w algorytmie FELICS tylko wtedy, gdy jasność danego piksela znajduje się poza przedziałem  $[F_L, F_H]$ . Zadaniem modelu danych jest w takiej sytuacji dobór parametru kodowania, czyli określenie, który kod z rodziny kodów zostanie użyty do zakodowania odległości jasności piksela od bliższego z krańców przedziału  $[F_L, F_H]$ . Model danych algorytmu FELICS to model pierwszego rzędu. Kontekstem dla danego piksela jest długość przedziału  $[F_L, F_H]$ , a zatem liczba kontekstów równa jest liczbie poziomów jasności obrazu.

Dla każdego kontekstu parametr kodowania dobierany jest w modelu danych za pomocą prostego i szybkiego algorytmu. Dla każdego kontekstu i dla każdej sensownej wartości parametru w modelu danych przechowuje się licznik sumarycznej liczby bitów, jaką dałoby zastosowanie tego parametru. Do kodowania wybierany jest parametr dający dla już przetworzonych danych najkrótszy kod. Opisana metoda doboru parametru nie ogranicza się do kodów Golomba czy Rice'a, nadaje się do dowolnej parametrycznej rodziny kodów o ograniczonym rozmiarze.

#### 4.2.5. Przykład

		P <sub>2</sub> 20
P <sub>1</sub> 10	X 25	

Rys. 3. Przykład dla algorytmu FELICS

Fig. 3. An example for the FELICS algorithm

Kodujemy piksel X z rys. 3. Używamy rodziny kodów Golomba. Przyjmujemy, iż bitem 0 wartości 1 kodujemy informację, że jasność piksela X znajduje się w przedziale  $[F_L, F_H]$ , 0 – poza przedziałem. Jeżeli jasność piksela znajduje się poza przedziałem  $[F_L, F_H]$ , to wyprowadzamy 0, gdy jasność X jest mniejsza od obu jego najbliższych sąsiadów, 1 – gdy znajduje się powyżej przedziału. Przedział  $[F_L, F_H]$  wyznaczony przez jasności dwóch najbliższych sąsiadów X to  $[10, 20]$ . Jasność piksela X znajduje się poza tym przedziałem, a zatem wyprowadzamy 0. Jasność piksela X znajduje się powyżej przedziału  $[F_L, F_H]$ , wyprowadzamy więc 1. Pozostaje zakodować odległość jasności X od górnego krańca przedziału: 4 (przyjmujemy, iż pierwsza wartość położona poza przedziałem znajduje się w odległości 0 od krańca przedziału). Kontekstem dla X jest długość przedziału  $[F_L, F_H]$ , czyli w tym przypadku 10. Przyjmijmy, że w modelu danych dla kontekstu 10 minimalna wartość licznika sumarycznej długości kodu występuje dla kodu o parametrze  $m=2$ . Wyprowadzamy słowo kodowe kodu Golomba o parametrze  $m=2$  dla liczby 4, czyli 1100.

Po zakodowaniu piksela z użyciem modelu danych należy przeprowadzić aktualizację modelu. Należy zwiększyć wartości wszystkich liczników dla kontekstu (10), w którym wystąpił kodowany piksel. Licznik dla danego kodu zwiększamy o długość słowa kodowego przyporządkowywanego przez ten kod zakodowanej właśnie odległości (4) jasności piksela od krańca przedziału  $[F_L, F_H]$ .

### 4.3. Standard JPEG-LS

Standard JPEG-LS [18, 59, 61] jest nowym standardem bezstratnej kompresji obrazów ustanowionym w roku 1999 z inicjatywy komitetu JPEG. W roku 1994 komitet JPEG ogłosił wezwanie do zgłaszania algorytmów-propozycji nowego międzynarodowego standardu kompresji obrazów. Wezwanie dotyczyło przede wszystkim algorytmów kompresji obrazów w stopniach szarości o głębi jasności 2-16 bitów na piksel. Dotyczyło ono również obrazów barwnych, w których poszczególne składowe kompresowane są niezależnie, oraz algorytmów określanych jako "near-lossless", czyli w istocie kompresora stratnego będącego wariantem podstawowego algorytmu bezstratnego.



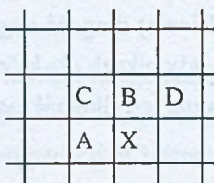
W odpowiedzi na wezwanie zgłoszono 9 propozycji, spośród których były 2 schematy transformacyjne i 7 predykcyjnych (w tym CALIC). Komitet JPEG zdecydował, iż nowy standard powstanie z uwzględnieniem kilku propozycji, spośród których jako rdzeń nowego standardu wybrano algorytm LOCO-I [60]. Algorytm LOCO-I wybrano, gdyż uzyskiwane przez ten algorytm współczynniki kompresji niewiele ustępują algorytmom kompresującym najsilniej, takim jak np. CALIC, od których algorytm LOCO-I jest znacznie szybszy. W dalszej części niniejszego podrozdziału scharakteryzowany zostanie podstawowy algorytm JPEG-LS, który różni się nieznacznie od algorytmu LOCO-I. Oprócz algorytmu kompresji standard JPEG-LS definiuje również pewne szczegóły implementacyjne oraz format pliku zawierającego skompresowany obraz.

W algorytmie JPEG-LS stosuje się schemat predykcyjny. Po linearyzacji obrazu w porządku rastrowym residuum wyznaczane jest z zastosowaniem predyktora nieliniowego. Zastosowano predyktor MED (ang. Median Edge Detector), który jest wariantem przedstawionego w roku 1990 przez S.A. Martucciego predyktora MAP (ang. Median Adaptive Predictor) [31]. Wartość predyktora MED wyznaczana jest jedynie z użyciem prostych stałopozycyjnych operacji dodawania i odejmowania oraz porównań liczb stałopozycyjnych:

$$Pred_{MED}(X) = \min(A, B, C) + \max(A, B, C) - C.$$

Położenie pikseli, których wartości używane są w funkcji predyktora, ilustruje rys. 4. Residuum kodowane jest z zastosowaniem modelu danych 3 rzędu i rodziny kodów Rice'a o ograniczonej długości słowa kodowego.

Kontekstem dla symbolu residuum  $Q_x = X - Pred_{MED}(X)$  są trzy wartości określone przez autorów algorytmu LOCO-I mianem gradientów:  $\gamma_1 = D - B$ ,  $\gamma_2 = B - C$ ,  $\gamma_3 = C - A$ . Konteksty zgrupowano w 365 kubelków kontekstów. Na podstawie informacji zgromadzonych w kubelku zawierającym kontekst danego symbolu w modelu danych wyznacza się parametr kodowania – numer kodu z rodziny Rice'a, który zostanie użyty do zakodowania symbolu.



Rys. 4. Położenie pikseli używanych w dekorrelacji i modelowaniu JPEG-LS  
 Fig. 4. Location of pixels used in the JPEG-LS for decorrelation and modeling

Dla rodziny kodów Rice'a istnieje prostsza niż znana z algorytmu FELICS metoda wyznaczania parametru kodowania. Zastosowano ją w algorytmie JPEG-LS. W jej przypadku dla każdego kubelka kontekstów wymagane jest przechowywanie tylko dwóch liczników: liczby symboli zakodowanych w tym kontekście i sumy wartości tych symboli.

Po dekorrelacji rozkład prawdopodobieństwa symboli residuum zbliżony jest do symetrycznego rozkładu wykładniczego, przy czym wartość oczekiwana rozkładu w poszczególnych kubelkach kontekstów jest różna od zera. Ze względu na sposób kodowania symboli residuum pożądanym jest, by wartość oczekiwana rozkładu prawdopodobieństwa była bliska 0. Dlatego też podczas dekorrelacji wprowadza się korektę. Symbol residuum wyznacza się nie jako  $Q_x = X - \text{Pred}_{MED}(X)$ , ale jako  $Q_x^\Delta = X - \text{Pred}_{MED}(X) - \Delta_x$ , gdzie  $\Delta_x$  jest adaptacyjnie szacowaną dla danego kubelka kontekstów wartością średnią  $Q_x$ .

Rozkład prawdopodobieństwa ciągu symboli residuum to w przybliżeniu symetryczny rozkład wykładniczy o wartości oczekiwanej 0. Aby zakodować wartość  $Q_x^\Delta$  kodem Rice'a należałoby wyprowadzić słowo kodowe kodu Rice'a dla  $|Q_x^\Delta|$  oraz pojedynczy bit określający znak  $Q_x^\Delta$ , co dla niezerowych wartości  $Q_x^\Delta$  uniemożliwiałoby uzyskanie współczynnika kompresji lepszego niż 2 bity na piksel. Dlatego też przeprowadza się zmianę uporządkowania wartości  $Q_x^\Delta$  tak, by za pomocą pojedynczego słowa kodowego kodu Rice'a kodować zarówno znak, jak i moduł wartości  $Q_x^\Delta$ .

Użycie kodera entropijnego, w którym stosuje się kody przedrostkowe do kodowania poszczególnych symboli residuum, powoduje, iż w najlepszym przypadku współczynnik kompresji wynosi 1 bit na piksel. W przypadku gdy prawdopodobieństwo wystąpienia danego symbolu bliskie jest 1, możliwe jest uzyskanie wielokrotnie lepszych współczynników kompresji przez użycie kodera arytmetycznego, co wiąże się z pogorszeniem złożoności czasowej algorytmu, lub przez kodowanie w odmienny sposób sekwencji identycznych symboli. W algorytmie JPEG-LS zastosowano drugie z tych rozwiązań. W przypadku gdy dla gradientów  $\gamma_1, \gamma_2$  i  $\gamma_3$  zachodzi  $\gamma_1 = \gamma_2 = \gamma_3 = 0$ , to prawdopodobieństwo, że jasność danego piksela równa jest jasności jego lewego sąsiada, jest znaczne. W takiej sytuacji kodowana jest (również adaptacyjnie za pomocą kodów Rice'a) długość ciągu pikseli identycznych z lewym sąsiadem danego piksela. W przypadku gdy piksel, dla którego zachodzi  $\gamma_1 = \gamma_2 = \gamma_3 = 0$ , różni się od swojego lewego sąsiada, kodowana jest długość ciągu 0.

## 5. Podsumowanie

Wraz ze wzrostem liczby i rozmiarów obrazów cyfrowych przetwarzanych przez systemy komputerowe zwiększają się wymagania dotyczące pojemności pamięci masowych oraz przepustowości kanałów transmisyjnych. Algorytmy kompresji pozwalają na zmniejszenie objętości plików zawierających obrazy, umożliwiając w ten sposób zwiększenie możliwości danego systemu pod względem ich składowania i transmisji.

Dzięki coraz lepszemu zrozumieniu charakteru danych obrazowych możliwe jest konstruowanie algorytmów kompresujących z coraz lepszymi współczynnikami kompresji i coraz większymi prędkościami. Przykładem takiego algorytmu może być FELICS, który w porównaniu z Lossless JPEG działa znacznie szybciej uzyskując zbliżone współczynniki kompresji. Przykładem również może być algorytm JPEG-LS, który działając z prędkością zbliżoną do algorytmu FELICS uzyskuje współczynniki kompresji znacznie lepsze, zbliżone do uzyskiwanych przez najsilniejszy obecnie znany algorytm, tj. CALIC.

Mnogość wymagań stawianych algorytmom kompresji obrazów przez rzeczywiste systemy informatyczne powoduje, że algorytmy te optymalizowane są nie tylko pod względem współczynników kompresji, ale również prędkości kompresji i złożoności pamięciowych. W niektórych zastosowaniach istotne są takie cechy algorytmu, jak możliwość transmisji progresywnej, odporność na zakłócenia, czy istnienie standardu definiującego algorytm i postać pliku zawierającego skompresowany obraz. W konsekwencji nie można wybrać jednego uniwersalnego algorytmu bezstratnej kompresji obrazów. Istnieje wiele algorytmów, przeznaczonych dla różnych zastosowań i różnych klas obrazów. Algorytmy te są i nadal będą intensywnie rozwijane.

**Podziękowanie.** Artykuł niniejszy powstał w wyniku projektu badawczego nr BW-444/Rau2/2001 zrealizowanego w roku 2001 w Instytucie Informatyki Politechniki Śląskiej.

## LITERATURA

1. Allred, L. G.; Kelly, G. E.: A lossless image compression technique for infrared thermal images. Proceedings of the SPIE, 1992 USA, Vol. 1702, pp. 230-7.
2. Arps, R. B.; Truong, T. K.: Comparison of international standards for lossless still image compression. Proceedings of the IEEE, June 1994 USA, Vol. 82(6), pp. 889-99.
3. Barequet, R.; Feder, M.: SICLIC: A Simple Inter-Color Lossless Image Coder. Proceedings DCC'99, USA, March 1999, pp. 501-12.

4. Consultative Committee for Space Data Systems: Lossless Data Compression. CCSDS Report Concerning Space Data System Standards, Green Book CCSDS 120.0-G-1 and Blue Book CCSDS 121.0-B-1, issue 1, Washington, USA, May 1997.
5. Christopoulos, C.; Skodras, A.; Ebrahimi, T.: The JPEG2000 Still Image Coding System an Overview. IEEE Transactions on Consumer Electronics, November 2000, Vol. 46(4), pp. 1103-27.
6. Carpentieri, B.; Weinberger, M. J.; Seroussi, G.: Lossless compression of Continuous-Tone Images. Proceedings of the IEEE, November 2000, Vol 88(11), pp. 1797-809.
7. Denecker, K.; Van Assche, S.; Philips, W.; Lemahieu, I.: State of the art concerning Lossless Medical Image Coding. IEEE Prorisc Workshop CSSP-97, Nov. 1997, pp. 100-5.
8. Drozdek, A.: Wprowadzenie do kompresji danych. WNT, Warszawa 1999.
9. Hu, Y. -C.; Chang, C. -C.: A new lossless compression scheme based on Huffman coding for image compression. Signal Processing: Image Communication, nr 16.
10. Houlding, D.; Vaisey, J.: Low entropy image pyramids for efficient lossless coding. IEEE Transactions on Image Processing, Aug. 1995 USA, Vol. 4(8), pp. 1150-3.
11. Howard, P. G.: The design and analysis of efficient lossless data compression systems. Rozprawa doktorska, Department of Computer Science, Brown University, Providence, Rhode Island 02912, June 1993.
12. Howard, P. G.; Vitter, J. S.: New methods for Lossless image compression using arithmetic coding. Proceedings DCC'91. IEEE Comput. Soc. Press Los Alamitos, CA, USA 1991, pp. 257-66.
13. Howard, P. G.; Vitter, J. S.: Error modeling for hierarchical lossless image compression. Proceedings DCC'92. IEEE Comput. Soc. Press Los Alamitos, USA 1992, pp. 269-78.
14. Howard, P. G.; Vitter, J. S.: Fast and efficient lossless image compression. Proceedings DCC '93. IEEE Comput. Soc. Press Los Alamitos, CA, USA, pp. 351-60.
15. Howard, P. G.; Vitter, J. S.: Fast progressive lossless image compression. Proceedings of the SPIE, 1994 USA, Vol. 2186, pp. 98-109.
16. Inglis, S. J.: Lossless Document Image Compression. Rozprawa doktorska, Department of Computer Science, The University of Waikato, Hamilton, New Zealand, March 1999.
17. ISO/IEC JTC1/SC29 WG1 (ITU-T SG8), FCD 15444-1, JPEG2000 Part I Final Committee Draft Version 1.0: JPEG2000 Image Coding System, 16 March 1997.
18. ISO/IEC JTC1/SC29 WG1 FCD 14495, public draft: Lossless and near-lossless compression of continuous-tone still images (JPEG-LS). ISO Working Document ISO/IEC JTC1/SC29/WG1 N522, July 1997.
19. ISO/IEC JTC1/SC29 WG1 (ITU-T SG8), FCD 14492, Final JBIG Committee Draft: Information Technology — Coded Representation of Picture and Audio Information — Lossy/Lossless Coding of Bi-Level Images, 16 July 1999.

20. International Telecommunication Union: Information Technology — Progressive Bi-Level Image Compression. ITU-T Recommendation T.82, March 1993.
21. Lee C. -C.: Lossless adaptive differential coding of images. Proceedings of the SPIE, 1995 USA, Vol. 2418, pp. 2-7.
22. Langdon, G.; Gulati, A.; Seiler, E.: On the JPEG model for lossless image compression. Proceedings DCC'92, IEEE Comput. Soc. Press Los Alamitos, USA 1992, pp. 172-80.
23. Langdon, G. G.; Haidinyak, C. A.: Experiments with lossless and virtually lossless image compression algorithms. Proceedings of the SPIE, 1995 USA, Vol. 2418, pp. 21-7.
24. Memon, N.: Adaptive coding of DCT coefficients by Golomb-Rice codes. Proceedings 1998 International Conference on Image Processing. ICIP98, IEEE Comput. Soc. Press, Vol.1, 1998, Los Alamitos, CA, USA, pp.516-20.
25. Marcellin, M. W.; Gormish, M. J.; Bilgin, A.; Boliek M. P.: An Overview of JPEG-2000. Proceedings IEEE DCC'2000, March 2000, pp. 523-44.
26. Melnychuck, P. W.; Rabbani, M.: Survey of lossless image coding techniques. Proceedings of the SPIE, 1989 USA, Vol. 1075, pp. 92-100.
27. Memon, N. D.; Sayood, K.: Lossless image compression: A comparative study. Proceedings of the SPIE, 1995 USA, Vol. 2418, pp. 8-20.
28. Motta, G.; Storer, J. A.; Carpentieri, B.: Adaptive Linear Prediction Lossless Image Coding. Proc. of the 1999 IEEE Data Compression Conference, Snowbird, Utah, USA, March 1999, pp. 491-500.
29. Merhav, N.; Seroussi, G.; Weinberger, M. J.: Coding of sources with two-sided geometric distributions and unknown parameters. IEEE Trans. Inform. Theory, Vol. IT-46(1), January 2000, pp. 229-36.
30. Merhav, N.; Seroussi, G.; Weinberger, M. J.: Optimal prefix codes for sources with two-sided geometric distributions. IEEE Trans. Inform. Theory, Vol. IT-46(1), January 2000, pp. 121-35.
31. Memon, N; Wu, X.: Recent developments in Context-Based Predictive Techniques for Lossless Image Compression. The Computer Journal, Great Britain 1997, Vol. 40(2-3), pp. 127-36.
32. Ordentlich, E.; Weinberger, M. J.; Seroussi, G.: A low-complexity modeling approach for embedded coding of wavelet coefficients. Proc. of the 1998 IEEE Data Compression Conference, Snowbird, Utah, USA, March 1998, pp.408-17.
33. Rice, R. F.: Lossless Coding Standards for Space Data Systems. Conference of Thirtieth Asilomar Conference on Signals, Systems and Computers. IEEE Comput. Soc. Press, Los Alamitos, CA, USA, vol.1, 1997, pp. 577-85.

34. Rice, R. F.; Yeh, P. S.; Miller, W.: Algorithms for High-Speed Universal Noiseless Coding. Proceedings of the AIAA Computing in Aerospace 9 Conference, San Diego, California, October 19-21, 1993.
35. Santa-Cruz, D.; Ebrahimi, T.: A Study of JPEG2000 Still Image Coding versus Other Standards. Proceedings of the X European Signal Processing Conference (EUSIPCO), Tampere, Finland, September 2000, Conference, Vol. 2, pp. 673-6.
36. Santa-Cruz, D.; Ebrahimi, T.: An analytical study of JPEG2000 functionalities. Proc. of the IEEE International Conference on Image Processing (ICIP-2000), Vancouver, Canada, September 2000, Vol. 2, pp. 49-52.
37. Santa-Cruz, D.; Ebrahimi, T.; Askelöf, J.; Larsson, M.; Christopoulos, C. A.: JPEG2000 Still Image Coding versus Other Standards. Proc. of the SPIE, Vol. 4115, pp. 446-54.
38. Skarbek, W.: Metody reprezentacji obrazów cyfrowych. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993.
39. Praca zbiorowa pod redakcją W. Skarbka: Multimedia algorytmy i standardy kompresji. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998.
40. Sun F. -K.; Maragos, P.: Experiments on image compression using morphological pyramids. Proceedings of the SPIE - The International Society for Optical Engineering, 1989 USA, Vol. 1199(pt.3), pp. 1303-12.
41. Said, A.; Pearlman, W. A.: Reversible image compression via multiresolution representation and predictive coding. Visual Communications and Image Processing '93, Proceedings of SPIE, Vol.: 2094, pp. 664-74.
42. Said, A.; Pearlman, W. A.: An Image Multiresolution Representation for Lossless and Lossy Image Compression. IEEE Transactions on Image Processing, Sept. 1996, vol. 5, pp. 1303-10.
43. Shen, L.; Rangayyan, R. M.: Improved joint bi-level image experts group (JBIG) data compression of continuous-tone images. Proceedings of the SPIE, Vol. 2727, 1996 USA, pp. 54-65.
44. Starosolski, R.: Fast, robust and adaptive lossless image compression. Machine Graphics and Vision, Warszawa 1999, Vol. 8(1), pp. 95-116.
45. Starosolski, R.: Szybkie, bezstratne oraz adaptacyjne metody kompresji obrazów w odcieniach szarości. Zeszyty Naukowe Politechniki Śląskiej, s. Informatyka, z. 37, Gliwice 1999, ss. 121-45.
46. Starosolski, R.: Fast and adaptive lossless grayscale image compression using the LZW algorithm. Archiwum Informatyki Teoretycznej i Stosowanej, Tom 11 (1999), z. 2, Katowice 1999, pp. 171-93.
47. Starosolski, R.: Bezstratne algorytmy kompresji obrazów. Rozprawa doktorska, Instytut Informatyki, Politechnika Śląska, Gliwice 2001.

48. Starosolski, R.: Algorytmy bezstratnej kompresji danych (w przygotowaniu).
49. Torbey, H. H.; Meadows, H. E.: System for lossless digital image coding/decoding. Proceedings of the SPIE, 1989 USA, Vol. 1199(2), pp. 989-1002.
50. Tilton, J. C.; Seiler, E.: CRUSH: a comparative lossless compression package. Proceedings IGARSS '94, Vol.1, IEEE 1994 New York, NY, USA, pp. 310-12.
51. Takamura, S.; Takagi, M.: Lossless image compression with lossy image using adaptive prediction and arithmetic coding. Proceedings DCC'94. IEEE Comput. Soc. Press, Los Alamitos, CA, USA, 1994, pp. 166-74.
52. W3C Recommendation: PNG (Portable Network Graphics) Specification, Version 1.0, 1 October 1996, dokument dostępny z: <http://www.w3.org/TR/REC-png.html>.
53. Wallace, G. K.: The JPEG Still Picture Compression Standard. Communications of the ACM, April 1991, Vol 34(4), pp. 30-44.
54. Witten, I. H.; Bell, T. C.; Emberson, H.; Inglis, S.; Moffat, A.: Textual image compression: two-stage lossy/lossless encoding of textual images. Proceedings of the IEEE, June 1994 USA, Vol. 82(6), pp. 878-88.
55. Wu, X; Choi, W. K; Memon, N: Lossless Interframe Image Compression via Context Modeling. Proc. of the 1998 IEEE Data Compression Conference, Snowbird, Utah, USA, March 1998, pp. 378-87.
56. Wang, L.; Goldberg, M.: Progressive image transmission using vector quantization on images in pyramid form. IEEE Transactions on Communications, Dec. 1989, USA, Vol. 37(12), pp. 1339-49.
57. Wu, X.; Memon, N.: Context-based, Adaptive, Lossless Image Codec. IEEE Transactions on Communications, vol. 45(4), April 1997, pp. 437-44.
58. Witten, I. H.; Moffat, A.; Bell, T. C.: Managing Gigabytes. Van Nostrand Reinhold, USA 1994.
59. Weinberger, M. J.; Seroussi, G.; Sapiro, G.: The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Trans. Image Processing, August 2000, Vol 9(8), pp. 1309-24.
60. Weinberger, M. J.; Seroussi, G.; Sapiro, G.: LOCO-I: A low complexity, context-based, lossless image compression algorithm. Proceedings DCC'96. IEEE Comput. Soc. Press Los Alamitos, CA, USA 1996, pp. 140-9.
61. Weinberger, M. J.; Seroussi, G.; Sapiro, G.: From LOCO-I to the JPEG-LS standard. Proc. of the 1999 IEEE International Conference on Image Processing, Kobe, Japan, October 1999.
62. Wu, X.: An Algorithmic Study on Lossless Image Compression. Proceedings of the IEEE DCC'96, USA, Mar. 1996, pp. 150-9.

63. Wu, X.: Efficient Lossless Compression of Continuous-tone Images via Context Selection and Quantization. IEEE Trans. on Image Processing, May 1997, Vol. IP-6, pp. 656-64.
64. Yovanof, G. S.; Sullivan, J. R.: Lossless predictive coding of colour graphics. Proceedings of the SPIE, 1992 USA, Vol. 1657, pp. 68-82.
65. Praca zbiorowa pod redakcją J. Zabrodzkiego.: Grafika komputerowa metody i narzędzia. WNT, Warszawa 1994.

Recenzent: Dr hab. inż. Władysław Skarbek Prof. Pol. Warszawskiej

Wpłynęło do Redakcji 20 czerwca 2002 r.

## Abstract

The paper presents an overview of the lossless image compression algorithms and of the standards concerning lossless image compression. In the paper basic methods used in lossless image compression are presented. For each of the image classes (binary, grayscale, color etc.) algorithms and standards of lossless compression are reviewed. Selected algorithms are described more thoroughly: the Lossless JPEG standard, FELICS — the simple and fast algorithm that acquires compression ratios close to significantly slower Lossless JPEG and the JPEG-LS — new international standard of lossless image compression.