

WYBRANE PROBLEMY BEZSTRATNEJ KOMPRESJI OBRAZÓW BINARNYCH

Urszula Wójcik, Dariusz Frejlichowski

Zachodniopomorski Uniwersytet Technologiczny
Wydział Informatyki
ul. Żołnierska 49, 71-210 Szczecin
e-mail: uwojcik@wi.zut.edu.pl, dfrejlichowski@wi.zut.edu.pl

Streszczenie: *W artykule przedstawiono podstawowe pojęcia i algorytmy związane z bezstratną kompresją obrazów binarnych. Dokonano przeglądu istniejących algorytmów bezstratnej kompresji, które są lub mogą być stosowane do obrazów binarnych, takich jak: kodowanie Huffmana, Golomba, Rice'a, kompresja JBIG, kodowanie długości sekwencji, kodowanie arytmetyczne oraz wybrane algorytmy predykcyjne.*

Słowa kluczowe: *Kompresja bezstratna, obrazy binarne*

A discussion on selected problems of lossless binary digital images compression

Abstract: *In the paper some basic notions and algorithms devoted to the problem of lossless digital binary images compression were provided. A brief survey on existing lossless algorithms was performed, concentrated on the ones that are or can be applied to binary images, e.g. Huffman, Golomb, Rice, JBIG, RLE, arithmetic coding, as well as the selected predicting approaches.*

Keywords: *Lossless compression, binary images*

należącymi do różnych dziedzin zastosowań), a wyborem najskuteczniejszej metody kompresji.

1. WPROWADZENIE

Niniejsze opracowanie, dokonujące przeglądu pojęć, metod i algorytmów z zakresu bezstratnej kompresji obrazów, jest próbą wstępnego ogarnięcia tego szerokiego obszaru. O zakresie i popularności problemu kompresji obrazów, w tym bezstratnej, może świadczyć mnogość pozycji książkowych jemu poświęconych (m.in. [1-5]).

W kolejnych rozdziałach niniejszego artykułu przedstawione zostaną wybrane teoretyczne podstawy kompresji, najważniejsze pojęcia związane z bezstratną jej wersją oraz przegląd miar jakości stosowanych do jej oceny. Trzeci rozdział dokonuje przeglądu istniejących metod kompresji bezstratnej obrazów w kontekście ich przyszłego zastosowania dla poszczególnych typów obrazów binarnych. W punkcie tym zawarta jest wstępna ocena porównawcza metod kompresji obrazów binarnych. Celem prowadzonych badań jest wskazanie zależności pomiędzy obiektem badań (obrazami binarnymi

2. PODSTAWOWE POJĘCIA

Kompresja jest sposobem reprezentacji informacji w zwężonej postaci, na podstawie znajomości jej własności strukturalnych. Jedna z definicji mówi, że ([5]): „kompresja danych to proces przekształcenia pierwotnej reprezentacji sekwencji danych w reprezentację o mniejszej liczbie bitów”. Za twórcę pojęcia kompresji uważa się Claude'a Shannona, który zajmował się metodami przesyłania i przechowywania danych. W latach czterdziestych rozwinął on dziedzinę wiedzy zwaną teorią informacji ([6]). Chociaż można wskazać wiele grup metod kompresji danych, np. na bazie charakteru kompresowanych danych lub stosowanego podejścia, najczęściej wyróżnia się dwa jej rodzaje: stratną i bezstratną.

Kompresja bezstratna odznacza się tym, że zrekonstruowany ciąg danych jest identyczny z sekwencją źródłową, a więc nie mamy do czynienia z utratą informacji. Ten rodzaj kompresji ma zastosowania głównie tam, gdzie wymagana jest wierna rekonstrukcja danych m.in. w obrazach medycznych, satelitarnych, historii operacji finansowych na kontach bankowych, czy kompresji tekstu. Natomiast kompresja stratna dopuszcza pewną utratę informacji, co nie pozwala na odtworzenie danych źródłowych z dokładnością do jednego bitu (tak jak w poprzednim przypadku), a jedynie w sposób przybliżony. Dzięki temu można uzyskać lepszy stopień kompresji, co znajduje zastosowanie na przykład w kompresji multimediiów.

Głównymi cechami współczesnych metod kompresji są zdolność wyboru ilości, jakości i postaci informacji wyjściowej oraz elastyczność. Algorytm powinien realizować takie zadania przy możliwie skromnych kosztach obliczeniowych i sprzętowych. Metody kompresji można oceniać według różnych kryteriów. Najważniejsze to:

- **złożoność;**
- **szybkość działania;**
- **pamięć potrzebna do implementacji;**
- **rozmiar kompresji** - proporcja liczby bitów reprezentujących dane przed i po kompresji, wyrażona ułamkiem bądź procentowo;
- **stopień kompresji CR** (ang. *compression ratio*) – stosunek liczby bitów potrzebnych do reprezentacji danych przed kompresją do liczby bitów potrzebnych do reprezentacji danych po kompresji, wyrażonej w postaci $n:l$;
- **średnia bitowa BR** (ang. *bit rate*) – średnia liczba bitów potrzebna do reprezentowania pojedynczego symbolu z alfabetu źródłowego;
- **procent kompresji CP** (ang. *compression percentage*) - stopień kompresji wyrażony w formie danych procentowych. Procent kompresji jest określany wyrażeniem ([5]):

$$CP = \left(1 - \frac{1}{CR}\right) * 100\%$$

- **współczynnik kompresji** – dostarcza informacji o ilości usuniętej nadmiarowej informacji, czyli określa w jakim stopniu dane źródłowe zostały zredukowane ([8]):

$$k = \frac{L_{we} - L_{wy}}{L_{we}}$$

Przy tym: L_{we} – rozmiar danych wejściowych, a L_{wy} – rozmiar danych wyjściowych;

- **entropia wejścia**, czyli współczynnik chaosu zbioru, określa „podatność” danego zbioru na kompresję. Entropia (w teorii informacji) jest średnią ilością informacji, za pomocą której dane źródło może być zakodowane w ustalonym modelu probabilistycznym. Entropia opisywana jest wzorem ([4]):

$$E_{we} = - \sum_{i=1}^n P(z_i) * \log_2(P(z_i))$$

Przy tym: $P(z_i)$ – prawdopodobieństwo zajścia zdarzenia z_i , natomiast n – ilość różnych znaków.

Ponadto musi być spełniony warunek ([8]):

$$\sum_{i=1}^n P(z_i) = 1$$

- **minimalny rozmiar danych po kompresji**, możemy obliczyć za pomocą wzoru ([8]):

$$L_{min} = \frac{L_{we} * E_{we}}{8[\text{bitów}]}$$

- **minimalny współczynnik kompresji**, możemy obliczyć za pomocą wzoru ([8]):

$$k_{min} = \frac{L_{we} - L_{min}}{L_{we}}$$

3. PRZEGLĄD METOD BEZSTRATNEJ KOMPRESJI OBRAZÓW BINARNYCH

Ogólnie bezstratne metody kompresji możemy podzielić na takie, które muszą posiadać wiedzę o danych wejściowych przed rozpoczęciem procesu kodowania oraz takie, które kodują dane na bieżąco lub dostosowują się już w procesie kodowania (tzw. metody adaptacyjne). Uproszczony schemat procesu kompresji metodami bezstratnymi pokazano na rys.1.



Rysunek. 1 Uproszczony schemat kompresji bezstratnej, źródło: [8].

Wybór odpowiedniej metody do konkretnego zadania nie jest łatwy, gdyż niektóre algorytmy cechują się lepszymi współczynnikami kompresji, inne zaś dużą szybkością kodowania i dekodowania. Niektóre metody są bardzo czułe na błędy spowodowane niedokładnym przesłaniem określonej partii danych. Czasami wystarczy zmiana nawet jednego bitu, aby dalsza partia danych była błędnie zdekodowana, podczas gdy w innych podejściach zmiana nawet całego bajtu powoduje mało znaczące straty. Przed podjęciem decyzji o wyborze sposobu kompresji ważną sprawą jest precyzyjne określenie typu danych wejściowych.

W kolejnych podrozdziałach przedstawiony zostanie krótki opis wybranych metod, które są lub mogą być stosowane w kompresji obrazów binarnych. Warto przy tym zaznaczyć, że obrazy binarne mogą być przed kompresją traktowane różnorodnie. Można się skupić na pojedynczych wartościach pikseli (0 lub 1) albo też łączyć je w większe grupy, np. po 8, 16, itd. i dopiero wtedy poddawać je kodowaniu.

3.1. Kodowanie Huffmana

Omawiany algorytm został zaproponowany w 1952 przez Davida Huffmana jako część rozwiązania zadania na zajęciach dla studentów. Metoda ta jest jedną z najbardziej znanych technik kodowania binarnego stosowaną w kompresji obrazów. Jest ona oparta na rozkładzie prawdopodobieństwa wystąpienia znaków. Kody generowane przez ten algorytm nazywane są kodami Huffmana i są to kody prefiksowe, będące optymalnymi dla przyjętego modelu probabilistycznego ([2]).

Kodowanie Huffmana jest algorytmem kompresji statycznej. Oznacza to, że program kompresujący musi przynajmniej raz przejrzeć cały kompresowany plik w celu stworzenia histogramu. Kodowanie Huffmana jest sposobem dopasowania zapisu znaków do rodzaju kompresowanych danych. Często wykorzystywane jest jako ostatni etap w różnych systemach kompresji stratnej oraz bezstratnej (np. w kompresji JPEG jako końcowy etap przetwarzania).

Algorytm kodowania wykorzystuje strukturę drzewa binarnego do wyznaczenia słów kodowych, których długość jest odwrotnie proporcjonalna do prawdopodobieństw p_i wystąpienia poszczególnych symboli s_i , oraz do kodowania znaków zmienną długością kodu. Konstrukcja drzewa binarnego rozpoczyna się od liści w kierunku korzenia. Algorytm Huffmana polega na łączeniu dwóch liści o najmniejszych wagach w poddrzewo z wierzchołkiem rodzica o wadze równej sumie wag liści. Następnie wśród liści i wierzchołków

poddrzew wyszukiwane są i łączone kolejne wierzchołki, które mają najmniejsze wagi. W ten sposób tworzone są kolejne poziomy poddrzewa, do momentu znalezienia rodzica każdego z wierzchołków drzewa. Istnieją również metody adaptacyjne, które tworzą drzewo w czasie kodowania i nie potrzebują czasochłonnego procesu przeglądania całego pliku.

Na efektywność kompresji wpływa przede wszystkim częstość występowania znaków. Kodowanie Huffmana cechuje się uniwersalnością i wynikami bliskimi optymalnym. Bardzo dobrze radzi sobie ono z grafiką, czy tekstem.

Niestety operacja kompresji i dekompresji jest stosunkowo czasochłonna, gdyż wykonywane są operacje na strukturze drzewa. Kolejna wada to duża czułość na zniekształcenia przesyłanej zakodowanej informacji: wystarczy zmiana jednego bitu, aby spowodować nieodwracalne błędy podczas dekompresji. Dlatego też w kodowaniu metodą Huffmana należy stosować dużą liczbę zabezpieczeń przed błędami (np. kompresowanie mniejszych bloków i zapis sumy kontrolnej po każdym bloku), co zmniejsza współczynnik kompresji. W celu zwiększenia efektywności kodowania, stosuje się efekt polegający na rozszerzeniu alfabetu źródła. Utrudnia to jednakże realizację etapu modelowania.

3.2. Kody Golomba

Kod Golomba (G_m) jest unarnym kodem symboli, który wykorzystuje kod dwójkowy stałej długości. Jest to kod o nieskończonym alfabecie źródła, który opisany jest dodatnią liczbą całkowitą m , zwaną rzędem kodu ([5]). Kody Golomba - Rice'a służą do kodowania liczb całkowitych przy założeniu, że im większa liczba, tym mniejsze prawdopodobieństwo jej wystąpienia.

Słowo kodu Golomba składa się z dwóch części:

- przyrostka, czyli odległości d symbolu (wyrażonej słowem kodu dwójkowego stałej długości) od początku przedziału ($B_m(d)$);
- przedrostka, czyli numeru przedziału symbolu zapisanego w kodzie unarnym ($U(u)$).

Słowo kodu Golomba można zapisać w postaci $G_m(i) = U(u) B_m(d)$, gdzie:

u – numer przedziału, $u = \lceil i/m \rceil$;

m – rząd kodu;

i – liczba całkowita ($i \geq 0$), $i = u m + d$;

d – odległość od początku przedziału, $d = i \bmod m$.

Kod Golomba cechuje optymalna postać zakodowanej reprezentacji danych przy mniejszych kosztach obliczeniowych. Duża jest też efektywność, porównywalna z kodowaniem metodą Huffmana. Kod zawiera nieskończony zbiór słów kodowych. Stopień

kompresji zależy nie tylko od częstości wystąpień symboli, ale też od ich „ustawienia”. Kod Golomba ma zastosowanie m.in. w metodzie odwracalnej kompresji JPEG-LS ([16]), oraz w metodzie adaptacyjnie dobieranych kodów Golomba ([15]).

3.3. Kody Rice’a

Kod Rice’a jest odmianą kodu Golomba, którą można traktować jako dynamiczny kod Golomba. Modyfikacja polega tu na dzieleniu zbioru liczb na podprzedziały o stałej długości (zwanej rzędem kodu), która jest potęgą dwójki. Kodowanie Rice’a z adaptacyjnym dobieraniem rzędu jest stosowane między innymi w algorytmie kompresji bezstratnej JPEG-LS oraz FLAC. Metoda ta uchodzi za bardzo wygodną w implementacji i pozwala na prostą i szybką realizację procesu kompresji.

3.4. Kodowanie arytmetyczne

Początki kodowania arytmetycznego sięgają lat sześćdziesiątych. Za jego twórców uważa się C.E. Shannona i P. Elias’a ([6]). Kodowanie arytmetyczne jest użyteczne, gdy rozmiar alfabetu jest mały, prawdopodobieństwa są zróżnicowane oraz gdy proces kodowania i modelowania są od siebie oddzielone. W opisywanym kodowaniu generowany jest unikalny znacznik, kodujący cały ciąg wejściowy, który odpowiada prawdopodobieństwu wystąpienia kodowanego ciągu, wyrażonego binarnie.

Zasada kodowania arytmetycznego opiera się na kodowaniu komunikatu jako liczby z lewostronnie domkniętego przedziału [0,1). Znajdowanie tej liczby polega na zwiększaniu jej precyzji poprzez stopniowe zawężanie przedziału, przy użyciu prawdopodobieństwa aktualnie przetwarzanego znaku ([1]).

Algorytm kodowania arytmetycznego działa na dowolnym źródle danych, zarówno dla danych tekstowych, obrazów, jak i dla plików dźwiękowych. Jest to bardzo efektywna metoda kompresji, która daje lepsze wyniki niż metoda Huffmana (od kilku do kilkunastu procent). Jednakże efektywność algorytmu zależy jedynie od rozkładu prawdopodobieństwa symboli. Ponadto nie wykorzystuje on cech właściwych dla poszczególnych typów danych. Przy praktycznych rozwiązaniach stosuje się dodatkowo struktury drzewa i kopca binarnego, co przyspiesza nawet kilkunastokrotnie działanie algorytmu.

W metodzie tej nie ma potrzeby stosowania rozszerzonego alfabetu źródła (tak jak w przypadku metody Huffmana). Pozwala to zbudować model na podstawie bogatszej statystyki danych źródła. Wygenerowanie znacznika dla konkretnego ciągu nie

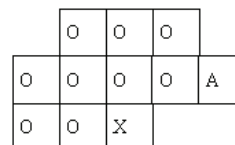
wymaga wyznaczania bądź pamiętania znaczników innych ciągów. W porównaniu z algorytmem Huffmana, kodowanie jest znacznie wolniejsze, natomiast dekodowanie jest operacją dużo prostszą. Podejście bywa jednak uważane za mało praktyczne – kosztowna arytmetyka zmiennoprzecinkowa i złożony algorytm dają w efekcie dużą złożoność obliczeniową. W przypadku długiej sekwencji danych źródłowych liczba kodowa może przekroczyć pojemność zmiennych (rejestrów) danej implementacji (procesora). Można przy tym zauważyć nieznaczne pogorszenie współczynnika kompresji w przypadku kodowania obrazów binarnych. Więcej informacji na temat kodowania arytmetycznego można znaleźć m. in. w [17].

3.5. Kompresja JBIG

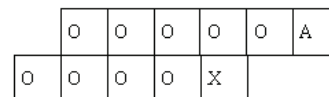
Standard kodowania obrazów binarnych JBIG (*ang. Joint Bi-level Image experts Group*) stanowi połączenie dwóch algorytmów ([2]): algorytmu kompresji bezstratnej i transmisji progresywnej kodowania obrazów binarnych.

Sposób kodowania piksela jest określony na podstawie „kontekstu”, na który składa się 9 bitów (oznaczonych na rys.2. literą **O**), położonych w sąsiedztwie kodowanego piksela. Znajdują się one na lewo i/lub wyżej (zakładamy kompresję od lewej do prawej i z góry na dół), zatem piksele te znane są dekodownikowi w momencie dekodowania. Wartości kolejnych pikseli kodowane są w oparciu o kontekst pikseli sąsiednich. Dodatkowo na kontekst składa się dziesiąty bit (litera **A** na rys.3), również na lewo i/lub w górę, którego położenie względem kodowanego symbolu może być dowolne (ale takie samo w obrębie całego obrazka; jest ono przesyłane wraz z plikiem) – służy to uwzględnieniu periodyczności w obrazku, nie mieszczących się w kontekście.

Kontekst trzywierszowy



Kontekst dwuwierszowy



Rysunek. 2 Rodzaje kontekstów. Oznaczenia: O – stały element kontekstu, A – zmienny element kontekstu, X – kodowany piksel, źródło: [2].

Po kompresji obrazu następuje drugi etap – transmisja progresywna. Polega ona na tworzeniu małych, mniej dokładnych wersji obrazu. Obrazy te mogą być zastosowane np. do szybkiego przesłania podglądu pliku. Zadanie to realizuje się poprzez zastępowanie kolejnych bloków pikseli jednym pikselem, wyznaczonym jako średnia wartość czterech pikseli składających się na dany blok. Schemat ten można stosować wieloetapowo, tworząc w ten sposób kilka obrazów, każdy o mniejszej rozdzielczości. Obrazy te nazywane są poziomami.

Liczba koderów używanych przy kompresji JBIG wynosi od 1024 do 4096. Zależy ona od tego, czy wykonywane jest kodowanie o niskiej, czy wysokiej rozdzielczości. W standardzie JBIG wszystkie 1024 kodery są wariantami koderów arytmetycznego nazywanego koderem QM ([2]). Kompresja JBIG umożliwia uszczegóławianie obrazu, bez zwiększenia rozmiaru danych (względem obrazu oryginalnego).

3.6. Kompresja JBIG2

Standard JBIG2 opiera się na tym samym algorytmie kodowania co JBIG. Jest jednakże rozszerzeniem swojego poprzednika, w którym różne fragmenty obrazu mogą być kodowane różnymi metodami.

Koder JBIG2 dzieli obraz binarny na tzw. **regiony**. Wyróżnia się **regiony tekstowe**, **regiony w półtonie** oraz **regiony ogólne**. Kodowanie bloków ogólnych opiera się na tym samym algorytmie kodowania arytmetycznego, którego używano w JBIG ([5]). Regiony tekstowe kompresowane są metodą słownikową. W segmencie umieszczony jest słownik symboli, który utworzony jest z map bitowych. Kodowanie polega na zapisaniu informacji o położeniu znaków oraz indeksu do elementu w słowniku symboli. W przypadku kompresji stratnej symbole nie muszą dokładnie pokrywać się z danymi na obrazie. Dzięki temu można osiągnąć mniejszy rozmiar słownika. Procedura kodowania regionów w półtonie jest także oparta na metodzie słownikowej. Kodowanie przebiega podobnie jak w przypadku segmentów tekstowych, z tym, że słownik zamiast symboli przechowuje wzorce półtonu o określonym rozmiarze.

W przeciwieństwie do JBIG, w omawianym algorytmie możliwy jest wybór metody kodowania, która daje najlepszą kompresję dla przetwarzanego typu danych. Dodatkowo można stosować różne metody kompresji dla różnych fragmentów obrazu.

3.7. CALIC

Istnieją dwa rodzaje adaptacyjnych modeli predykcji: adaptacja w przód (ang. *forward adaptation*) oraz wstecz

(ang. *backward adaptation*). Adaptacja w przód wykorzystuje dostęp do całej sekwencji źródłowej oraz importuje parametry modelu predykcji, które dynamicznie zmodyfikowano w trakcie kodowania. Algorytm kompresji (dekompresji) jest niesymetryczny. Dane wejściowe są dzielone na kilka części. W przypadku obrazów są to często bloki prostokątne lub sąsiednie fragmenty danych. Adaptacja wstecz polega na poszukiwaniu efektywnego modelu na podstawie informacji zakodowanej lub zdekodowanej. W tym przypadku model predykcji nie wykorzystuje całej informacji źródłowej, przez co jest znacznie uboższy.

Metoda CALIC (ang. *context-based adaptive image coder*) wykorzystuje pełne sąsiedztwo najbliższych punktów podczas kodowania pikseli obrazu, co zapewnia większą efektywność kompresji obrazów.

3.8. JPEG-LS

Kodowanie stosowane w formacie JPEG-LS polega na przesyłaniu par liczb – wartości próbki i liczby powtórzeń tej wartości dla kolejnych próbek. Podejście takie jest szczególnie efektywne dla obrazów nienaturalnych, np. zrzutów ekranu komputera. Podczas modelowania kontekstu określany jest rozkład prawdopodobieństw użyty do kodowania aktualnej wartości. Istnieją dwa rodzaje trybów kodowania:

- tryb regularny, jeśli jest małe prawdopodobieństwo wystąpienia serii próbek o identycznej wartości;
- tryb sekwencji próbek identycznych, jeśli jest duże prawdopodobieństwo wystąpienia serii próbek o takiej samej wartości.

3.9. Metoda kodowania długości serii (RLE)


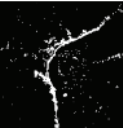



Metoda kodowania długości serii - RLE (ang. *Run-Length Encoding*) należy do grupy metod predykcyjnych. Jest to prosta metoda bezstratnej kompresji danych. Algorytm polega na wyszukiwaniu ciągów tych samych znaków i opisywaniu ich za pomocą licznika powtórzeń. Metoda najbardziej skuteczna okazuje w kompresji obrazów binarnych. Stosuje się ją również do kodowania dokumentów przesyłanych za pomocą faksu (najczęściej w jednym z formatów: PCX, BMP, TGA). Sprawdza się wszędzie tam, gdzie występuje duża powtarzalność tych samych elementów następujących po sobie, takich jak kolor tła, czy inne duże obszary jednolitego zabarwienia. Metoda charakteryzuje się dużą szybkością kodowania i dekodowania.

Istnieje wiele metod zwiększania efektywności kompresji RLE. Przykładem jest wektorowe kodowanie długości serii (ang. *vector run-length encoding*).

Kodowane są tutaj serie bloków pikseli o wymiarach $m \times m$, zamiast pojedynczych pikseli binarnych. Można dzięki temu uzyskać nawet kilkudziesięcioprocentową poprawę efektywności kodowania w porównaniu z jednowymiarową metoda RLE ([5], [10]).

4. PODSUMOWANIE

W artykule dokonano przeglądu podstawowych pojęć i metod z zakresu bezstratnej kompresji obrazów binarnych. W przyszłej pracy omawiane algorytmy będą badane na rzeczywistych klasach obiektów binarnych, takich jak semacody, odciski palców, czy obrazy radarowe. Dobór materiału badawczego celowo koncentruje się na aktualnych problemach badawczych, takich jak bezpieczeństwo czy różnego rodzaju zastosowania techniczne. Na rys. 3 pokazano przykłady danych testowych, należące do pięciu klas zastosowań. Będą one poddawane działaniu różnych algorytmów kompresji, by sprawdzić, jaki wpływ na ich efektywność ma charakter danych wejściowych.

Nazwa grupy	Dane
 Odciski palców	Ilość próbek: 51 Rozdzielczość: 256 x 256 px Typ plików: bmp Źródło pochodzenia: fingerp Wielkości plików: 8,06 KB Głębina w bitach: 1
 Obrazy radarowe	Ilość próbek: 402 Rozdzielczość: 228 x 228 px oraz 225 x 225 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 51,8 KB oraz 51,1 KB Głębina w bitach: 1
 Semacody	Ilość próbek: 50 Rozdzielczość: 216 x 216 px oraz 184 x 184 px Typ plików: bmp Źródło pochodzenia: QR-Code Generator Wielkości plików: 137 KB oraz 100 KB Głębina w bitach: 1
 Skany tekstu	Ilość próbek: 30 Rozdzielczość: 5104 x 7016 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 34,1 MB Głębina w bitach: 1
 Kontury kształtów	Ilość próbek: 110 Rozdzielczość: 128 x 128 px oraz 100 x 100 px Typ plików: bmp Źródło pochodzenia: zbiory własne Wielkości plików: 17 KB oraz 10,8 KB Głębina w bitach: 1

Rysunek. 3 R Przykłady klas danych obrazów binarnych do testowania algorytmów kompresji, źródło: opracowanie własne.

Literatura

1. Drozdek A., "Wprowadzenie do kompresji danych", WNT, Warszawa 1999
2. Sayood K., "Kompresja danych wprowadzenie", RM, Warszawa 2002
3. Skarbek W., „Metody reprezentacji obrazów cyfrowych”, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1993
4. Skarbek W. (red.), „Multimedia algorytmy i standardy kompresji”, Akademicka Oficyna Wydawnicza PLJ, Warszawa 1998
5. Przelaskowski A., "Kompresja danych, podstawy, metody bezstratne, kodery obrazów", Wydawnictwo BTC, Warszawa 2005
6. Shannon C. E., "A Mathematical Theory of Communication", Bell System Technical Journal, 1948, vol. 27, pp. 379-423
7. Starosolski R., "Algorytmy bezstratnej kompresji obrazów", Studia Informatica, 2003, vol. 24, Nr 1(52), ss. 138-158
8. Heim K., "Metody kompresji danych", Wydawnictwo MIKOM, Warszawa 2000
9. Huffman D. A., "A method for the construction of minimum-redundancy codes", Institute of Radio Engineers, 1952, vol. 40, no.9, pp. 1098-101
10. Wang, Y., Wu, J.-M., "Vector run-length coding of Bi-level images", Proc. of Data Compression Conference, 1992, pp. 289 – 298
11. Knuth D. E., "Dynamic Huffman coding", Journal of Algorithms, 1985, vol. 6, pp. 163-180
12. Starosolski R., "Algorytmy bezstratnej kompresji obrazów", Studia Informatica, 2002, Vol. 23, Nr 4(51), ss. 277-300
13. Pennebaker W. B., Mitchell J. L., "JPEG: Still Image Data Compression Standard", Springer, 1993
14. Stateczny A., "Nawigacja porównawcza", Gdańskie Towarzystwo Naukowe, Gdańsk 2001
15. Seroussi, G., Weinberger, M.J., "On adaptive strategies for an extended family of Golomb-type codes", Proc. of the Data Compression Conference, 1997, pp. 131 – 140
16. Weinberger M., Seroussi G., Sapiro G., "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", Hewlett-Packard Laboratories Technical Report No. HPL-98-193R1, 1998
17. Langdon G.G., "An introduction to arithmetic coding", IBM Journal Research and Development, 1984, vol. 28, pp. 135-148