

KOMPRESJA DANYCH OBRAZOWYCH

Zarys zagadnień istotnych

Artur Przelaskowski

PRZEDMOWA

Niniejsza pozycja jest owocem mojej siedmioletniej pracy w dziedzinie kompresji danych, w tym przede wszystkim danych obrazowych. Składa się na nią wiele doświadczeń związanych z aplikacją znanych algorytmów, opracowaniem nowych wersji efektywnych metod kompresji, a także uzyskaniem własnych rozwiązań pozwalających zwiększyć skuteczność kompresji w możliwie szerokiej gamie zastosowań. Różnorodność konstruowanych algorytmów, warsztatów analizy danych, narzędzi do redukcji różnego typu nadmiarowości, a także wymagań stawianych technikom kompresji w zależności od uwarunkowań konkretnych aplikacji sprawia, że nawet czysto przeglądowe opracowanie mogłoby z pewnością zostać zrealizowane w znacznie obszerniejszej formie. Stąd konieczność wyboru zagadnień szczególnie istotnych i ciekawych, oczywiście z punktu widzenia moich doświadczeń i posiadanej wiedzy. Dokładniejsza prezentacja wybranych tematów spowodowana jest ich większym znaczeniem i wartościami poznawczymi, czyli dokonaną przeze mnie selekcją zagadnień istotnych na tle choć zdawkowo zarysowanych technik i algorytmów, o których nie sposób nie wspomnieć. Przedstawione zostały również pewne nowatorskie koncepcje własne autora, nie ze względu na ich doniosłość, lecz dla lepszego zrozumienia sposobu rozwiązywania konkretnych problemów z dziedziny kompresji. Chciałbym podkreślić także szczególne zamiłowanie Autora do aplikacji medycznych, co w przypadku optymalizacji metod kompresji obrazów wydaje się szczególnie korzystne i użyteczne. Przytaczane tutaj przykłady, rozważania i konkluzje odwołują się więc nierzadko do wymagających zastosowań medycznych systemów obrazowania.

Zasadniczym celem jest więc ukazanie najistotniejszych zagadnień dotyczących technik kompresji danych, w tym szczególnie danych obrazowych, wybranych realizacji koderów stratnych i bezstratnych oraz współczesnych trendów rozwojowych w tej dziedzinie. Chciałbym zarazić Czytelnika pięknem tej tematyki i zachęcić do wysiłku na rzecz budowania nowych algorytmów, innych sposobów reprezentowania i opisu informacji, która jest transportowana w ogromnych ilościach w skali globalnej i przez wielu jest rozpoznawana jako nerw i główny czynnik rozwoju współczesnego świata.

Kompresję można obrazowo przedstawić na wiele sposobów: jako przystosowanie obiektu do możliwości przesyłania, odkurzenie z niepotrzebnych nadmiarów, trafne zawarcie informacji w kilku słowach, z których następnie odczytywane są tomy itp. Filozofia kompresji może być różnorodna i wpływa znacząco na technikę budowania nowych algorytmów.

Rozważmy przykład osoby stojącej na zielonej łące, nad którą rozciąga się błękitne niebo. Fotograf robi jej zdjęcie, dla wygody przyjmijmy, że aparatem cyfrowym. Powstaje obraz przedstawiający tę osobę na tle błękitnego nieba i zielonej łąki. Jakkolwiek zdjęcie może być bardzo piękne i ciekawe, to jest ono tylko odbiciem, zapisem w pewnej konwencji informacji o osobie otoczonej pięknym krajobrazem. Cechy sfotografowanej sceny są w dużym stopniu zależne od kunsztu fotografa, warunków oświetlenia, czułości systemu obrazowania itp. Czynniki te mogą w różny sposób zniekształcać obraz rzeczywisty (za taki uznajemy obraz rejestrowany przez prawidłowy narząd wzroku), który jest podmiotem w procesie rejestracji obrazu i stanowi treść (informację) podstawową. Jeśli ona zostanie dobrze zdefiniowana to np. ten sam efekt zmiany barwy, oświetlenia itd. można uzyskać technikami przetwarzania obrazu działając na bazie informacji podstawowej. W ten sposób w zależności od aplikacji można tworzyć różne wersje obrazu, między innymi taką, która odpowiada dokładnie zdjęciu wykonanemu przez fotografa. Dotarcie do zasadniczej treści i efektywny jej

zapis stanowią kluczowe zadanie, jakie stoi przed procesem kompresji. Zapis ten można wzbogacić o dodatkowe parametry przekształceń tworzących różne wersje reprezentacji informacji podstawowej.

Idealnym rozwiązaniem byłoby znalezienie takiego opisu obiektu przedstawionego na obrazie, który wyrazi wszystkie jego cechy i umożliwi odtworzenie w zależności od potrzeb zarówno pełnej, niezakłóconej postaci obiektu, jak też odpowiednio uproszczonej informacji zapisanej w możliwie najoszczędniejszej reprezentacji. Tak właśnie rozumiem ideę kompresji - jako docieranie do istoty zjawiska (reprezentacji fundamentalnej), a nie jej odbicia (reprezentacji pochodnej).

Szczególnie ważne jest więc opracowanie efektywnego modelu samej istoty zjawiska, umożliwiającego różne formy odtworzenia jego cech. Elementy tej filozofii można znaleźć w nowym paradygmacie kompresji formułowanym w ramach standardu JPEG 2000: stworzenie reprezentacji zawierającej upakowaną esencję zbioru danych, którą można odtwarzać różnorako, w zależności od zastosowań. Jak jednak znaleźć tę esencję, jak w kilku znakach, symbolach, skromnym alfabetem wyrazić istotę zjawiska, sedno informacji, którego odbiciem jest kompresowany zbiór danych? Mam nadzieję, że prezentowane w tym opracowaniu rozwiązania przynajmniej częściowo odpowiadają na to pytanie.

ROZDZIAŁ 1. WPROWADZENIE

Kompresja zbiorów różnego typu danych jest ostatnio jednym z najbardziej frapujących tematów z pogranicza wielu dziedzin nauki i techniki, takich jak teoria informacji, teoria stopnia zniekształceń źródeł informacji, teoria przekształceń (unitarnych, nieliniowych, z bazą nadmiarową itp.), czasowa i sprzętowa optymalizacja algorytmów itp. Jednocześnie wiele zagadnień istotnych z punktu widzenia kompresji danych ma znacznie szerszy charakter. Prace nad uściśleniem pojęcia informacji i ilościowym jej opisem, sprecyzowaniem wiedzy (dostępnej) a priori na temat analizowanych zbiorów danych i optymalną (możliwie prostą a jednocześnie zupełną) ich charakterystyką (modelowaniem), obiektywizacją wrażeń wzrokowej percepcji i ujęciem ich w ramy efektywnego modelu stanowią także kluczową tematykę wielu innych zagadnień z obszaru optymalizacji technik analizy, przetwarzania i rozpoznawania danych. Duża liczba opracowań różnorodnych metod kompresji stratnej i bezstratnej, formatów, mnogość publikacji, badań i eksperymentów oraz konkretnych implementacji, szeroko rozwinięty proces standaryzacji metod wyznaczania efektywnych reprezentacji danych spowodowane są gwałtownym rozwojem technologii informatycznych i telekomunikacyjnych oraz wzrostem liczby i rozmiarów zbiorów danych różnego rodzaju, pojawiających się niemal w każdej dziedzinie życia, gdzie dotarły komputery, oraz związanymi z tym trudnościami ich gromadzenia, przeglądania i wymiany (transmisji).

W ostatnich latach rozwój technik kompresji wykorzystując coraz większą moc obliczeniową komputerów oraz nowe technologie sprzętowe, skupiony jest głównie wokół takich zagadnień jak:

- budowanie skuteczniejszych modeli opisujących kompresowane zbiory danych (statystycznych, predykcyjnych, kontekstowych wyższych rzędów, nieliniowych, itp.),
- tworzenie bardziej kompleksowych algorytmów adaptacyjnych, często nie przyczynowych z koniecznością zapisu dodatkowej informacji w nagłówku,
- poszukiwanie zbioru funkcji bazowych przekształceń (nie tylko ortogonalnych) dopasowanych do własności konkretnego zbioru danych, czy też dynamicznie dobieranych w samym procesie kompresji, wykorzystywanie nowych klas transformat (nieliniowych, przestrzeń-skala, nadmiarowych itp.),
- konstruowanie efektywnej reprezentacji skalowalnej, z pełną kontrolą ilości przesyłanej informacji, jej hierarchii, wygodnej w interpretacji, indeksowaniu, transmisji, odpornej na zakłócenia, z selekcją wybranych obszarów i orientowaniu progresji na żądany rodzaj informacji,
- rozwój koncepcji metod kompresji sekwencji obrazów w czasie rzeczywistym, w tym obiektowych metod tzw. drugiej generacji oraz technik skojarzonych z zorientowanymi treściowo bazami danych,
- tworzenie zobiektywizowanych miar jakości kompresowanych stratnie obrazów (coraz bardziej skorelowanych z oceną psychowizualną) i włączenie ich w proces konstruowania algorytmów kompresji,
- opracowywanie nowych standardów wykorzystujących techniki kompresji zarówno na poziomie pikselowym jak i obiektowym, a także semantycznym w kontekście operowania protokołami transmisji strumieni danych, informacją multimedialną, treścią bazodanową itp.

Zapotrzebowanie na kompresje

Postęp technologiczny w dziedzinie rejestracji obrazu, dźwięku, różnego typu danych, postępująca komputeryzacja niemal wszystkich dziedzin życia, gwałtowny rozwój połączeń sieciowych (internet, intranet) itd. pociągają za sobą fakt powstawania ogromnych ilości informacji w postaci cyfrowej. Liczne zbiory danych, które są zapisywane, przetwarzane, przechowywane, wymieniane, przesyłane stanowią w bardzo wielu przypadkach główny przedmiot analizy różnego typu systemów, a zdolność szybkiej manipulacji zbiorami danych (wygodne i sprawne przeglądanie, wyszukiwanie, sortowanie, klasyfikacja itd.) decyduje często o efektywności i użyteczności tych systemów.

Obok zbiorów zawierających różnego typu dokumenty, opisy, relacje, podania, programy komputerowe itp. powstaje szereg zapisów dźwięku i mowy (rejestrowane ścieżki dźwiękowe z koncertów, nagrania studyjne, archiwalne nagrania audycji, rejestracja ważnych spotkań). Wspomnieć należy także o różnorodnych danych pomiarowych (metrologicznych) z przemysłu, wielu dziedzin nauki i techniki, meteorologii, itd. Pokazną grupę stanowią dane obrazowe, w tym skanowane dokumenty, rysunki, zarejestrowane kamerą czy aparatem fotograficznym obrazy naturalne, obrazy satelitarne, medyczne czy grafiki komputerowej, itd. Powstają coraz nowocześniejsze systemy generacji, rekonstrukcji, przetwarzania i analizy obrazów w astronomii, medycynie, poligrafii, sztukach graficznych (ang. graphic arts), sądownictwie, reklamie, produkcji przemysłowej, e-komercji itd. Dodać należy także gwałtownie rozwijający się rynek multimedialny, w tym muzyczne koncerty na żywo w internecie, transmisje obrazu i dźwięku z wielkich uroczystości czy wydarzeń na skalę światową, telekonferencje, teleedukacja, sprzedaż różnego typu produktów o charakterze multimedialnym w e-sklepach, itp.

Nie sposób także nie wspomnieć o intensywnie rozwijającej się technologii metod obrazowania w medycynie i tysiącach badań wykonywanych codziennie w oddziałach radiologicznych różnego typu ośrodków medycznych: badaniach ultrasonograficznych (USG), tomografii komputerowej (CT), cyfrowej radiografii (CR), tomografii rezonansu magnetycznego (MRI) i medycyny nuklearnej (NM).

Wielkie bazy danych medycznych i kompleksowe systemy informacyjne szpitali (ang. Hospital Information Systems – HIS), w których składowane są odpowiednio zabezpieczone wyniki badań z kilku lokalnych ośrodków i zapewnione są możliwości wymiany informacji z nadzorującymi jednostkami regionalnymi i krajowymi, są bardzo pomocne w śledzeniu historii choroby i umożliwiają diagnozowanie w oparciu o pełne wyniki różnego typu badań. Są przy tym bardzo pomocne nie tylko dla lekarzy, ale także dla pacjentów i osób poszukujących informacji i pomocy w zakresie podstawowej opieki medycznej, dostarczając odpowiednio przygotowaną wiedzę oraz umożliwiając często kontakt z właściwymi służbami medycznymi. Jednak w tak rozbudowanych systemach z wieloma użytkownikami ograniczenia technologiczne powodują, że czas przeszukiwania bazy, sortowania czy dostępu do odpowiedniej informacji znacznie się wydłuża w miarę rozrastania się gromadzonych zasobów. Problemy pojawiają się także przy efektywnej wymianie informacji diagnostycznej pomiędzy pracownikami, ośrodkami medycznymi i potencjalnymi użytkownikami, przy multimedialnej obsłudze operacji transmitowanych 'na żywo', itp. hamując szybki rozwój telemedycyny ze względu na bardzo poważne ograniczenia, które wynikają z małej przepustowości sieci komputerowych, tak globalnych jak i lokalnych.

Praktycznie każdy nowoczesny system wymiany informacji wymaga zastosowania skutecznej w danym przypadku metody kompresji danych, aby zachować dobrą jakość przekazywanej informacji i zwiększyć pole aplikacji tych systemów. Bardzo istotnym jawi się

też aspekt korzyści finansowych związanych z oszczędnościami czasu, nośnika, wprowadzeniem nowych usług związanych z wymianą informacji. Szybki wzrost pojemności pamięci masowych oraz wolniejszy rozwój jakości i zdolności przepustowych łącz telekomunikacyjnych nie jest antidotum, a jedynie uzupełniającym sposobem realizacji głównej idei szybszego przekazywania informacji o lepszej jakości i nadawania jej bardziej użytecznej reprezentacji.

Ogólna charakterystyka danych

Nie ma technik kompresji, które są optymalne w każdym zastosowaniu. Różnorodność analizowanych danych, zarówno co do ogólnej czy lokalnej charakterystyki zbioru wartości, jak też sposobu ich wykorzystania powoduje, że dla różnych typów danych należy stosować dopasowane do ich własności algorytmy w celu uzyskania maksymalnej skuteczności kompresji.

Dane jednowymiarowe

Najczęściej spotykane zbiory tekstowe zawierają bajtowe informacje o kolejnych znakach (np. zapisanych w kodzie ASCII) tworzących słowa, zdania tekstu uzupełnione znakami formatującymi i narzucającymi interpretację, rozkazy danego języka programowania, wektory danych arkuszy kalkulacyjnych, itp. Ważne są tutaj częstości wystąpień pojedynczych znaków w tekstach danego języka, formatu, dokumentu, jak też różnych kombinacji tychże znaków w określonym kontekście, wynikające często z koncepcji składniowej i semantycznego określonego zbioru danych.

Dużą grupę zbiorów stanowią zapisy dźwięku i mowy. Są to często zbiory wartości kolejno próbkowanych sygnałów analogowych, których analiza i ewentualne przetwarzanie najczęściej uwzględnia percepcyjne zdolności ucha ludzkiego, jak też sam mechanizm generacji pojedynczego dźwięku czy słowa. Odmienny charakter danych będących zapisem mowy i łagodniejsze kryteria jakościowe (koncentrujące się głównie na zachowaniu zrozumiałości wypowiedzi) implikują inną klasę rozwiązań algorytmów kompresji, a także inne modele odbiorcy i kryteria dopuszczalnych strat. Można mieć także do czynienia ze zbiorami danych zawierającymi dźwięk lub mowę generowane syntetycznie.

Przykładem danych pomiarowych będących rejestracją zmian pewnych wielkości fizycznych w czasie może być zarejestrowane badanie elektrokardiograficzne (EKG) z kilku odprowadzeń, po np. 1300 próbek dla każdego odprowadzenia. Innym przykładem jest rejestracja wahań sieci w pewnym czasie poprzez urządzenie kontrolujące (z przetwornikiem AC i zapisem próbek z pewną częstotliwością dla kilku czy kilkunastu linii). Dane pomiarowe zawierają często na istotnym poziomie różnego typu zniekształcenia, szumy, artefakty, a wydzielenie sygnału użytecznego może być niejednokrotnie zrealizowane w bardzo ograniczonym stopniu. Ponadto, rejestrowane sygnały mają nierzadko silnie niestacjonarny charakter, a różnego typu zaburzenia ewentualnych regularności występowania określonych treści w sygnale mają bardzo istotne znaczenie interpretacyjne. Dynamika sygnału może być silnie zróżnicowana, a użyteczne wahania amplitudy, częstotliwości czy fazy sygnału mają bardzo zmienny charakter wymuszając wysoką adaptacyjność algorytmów przetwarzania i kompresji danych.

Obrazy

Obrazy będące bardzo istotnym elementem współczesnych systemów informacji można podzielić na dwie zasadnicze grupy: analogowe i cyfrowe. Obrazy analogowe opisywane są funkcją jasności obrazu reprezentującą przestrzenny rozkład energii

promieniowania widzialnego, która jest czterowymiarową funkcją $C(x,y,t,\lambda)$ ograniczonych niezależnych zmiennych (współrzędne przestrzenne x i y , czas t oraz długość fali promieniowania λ), rzeczywistą i nieujemną oraz oczywiście ciągłą. Współrzędne przestrzenne ograniczone są polem widzenia odbiorcy obrazu, a pozostałe zmienne czasem obserwacji i możliwością percepcji określonego zakresu promieniowania. Odpowiedź standardowego oka ludzkiego, w zakresie intensywności i koloru, na funkcję jasności obrazu wyraża się w pojęciach luminancji i chrominancji. I tak definiuje się funkcję pola obrazu jako:

$$F_i(x,y,t) = \int_0^{\infty} C(x,y,t,\lambda) S_i(\lambda) d\lambda, \quad (1.1)$$

gdzie $S_i(\lambda)$ oznacza widmową odpowiedź i - tego czujnika. Oczywiście, jeśli dobierzemy trzy czujniki 'wrażliwe' na różne zakresy długości fal promieniowania (np. czerwony, zielony i niebieski), wówczas możemy wytworzyć sygnał luminancji i chrominancji (w różny sposób w zależności od systemu wizyjnego) zawierający informację o treści obrazu. W naszych ogólnych rozważaniach pominiemy jednak dla wygody zagadnienia wyboru pewnych podzakresów częstotliwościowych w odbiorniku promieniowania widzialnego i operujemy dalej pojęciem funkcji jasności obrazu. Ponieważ w zdecydowanej większości systemów rejestracji pole obrazu jest prostokątne (lub prostopadłościennie w systemach trójwymiarowych), prezentowany tutaj uproszczony model obrazu będzie miał zdefiniowane pole prostokątne.

Funkcja jasności obrazu może być próbkowana wzdłuż każdej współrzędnej. Jeżeli funkcja ta dla wybranej chwili czasowej jest próbkowana wzdłuż obu współrzędnych, a ponadto wartości funkcji jasności są kwantowane w każdym punkcie pola obrazu, wówczas mamy do czynienia z obrazami cyfrowymi. Proces kwantyzacji i próbkowania powtórzony w kolejnych wybranych chwilach czasowych dostarcza sekwencji obrazów. W ten sposób powstają obrazy cyfrowe reprezentujące ciągłe obrazy naturalne (ang. continuous natural images).

Próbując nieco sformalizować zapis reprezentujący obrazy cyfrowe posłużmy się następującym wyrażeniem. Niech $f_I : P \rightarrow C$ oznacza przekształcenie dyskretnego prostokątnego pola P obrazu I w zbiór wartości funkcji jasności C . Dziedziną tego przekształcenia jest zbiór wszystkich punktów pola $P = \{(x,y) \in Z^2 : x_L \leq x \leq x_H, y_L \leq y \leq y_H\}$, gdzie: $x_L = \min_{(x,y) \in I} \{x\}$, $x_H = \max_{(x,y) \in I} \{x\}$, y_L, y_H - analogicznie względem y , $W = x_H - x_L$ - szerokość obrazu, $H = y_H - y_L$ - wysokość obrazu, Z - zbiór liczb całkowitych. Ogólna postać zbioru wartości funkcji jasności wygląda następująco: $C = \{0,1,\dots,N_1-1\} \times \dots \times \{0,1,\dots,N_i-1\} \times \dots \times \{0,1,\dots,N_k-1\}$, gdzie k - liczba pasm spektralnych (wymiar przestrzeni kolorów), N_i - liczba poziomów wartości pasma i . Dla klasycznych obrazów kolorowych $k=3$ (RGB), a dla obrazów ze skalą szarości $k=1$.

Obrazy cyfrowe można zasadniczo podzielić na dwie grupy: binarne (ang. bilevel) i wielopoziomowe (ang. continuous-tone, multilevel). Dla obrazów binarnych $N=2$ (faksy, teksty z edytorów, itd.), natomiast w obrazach wielopoziomowych wartości pikseli wyrażane są w skali szarości bądź koloru. Typowa wartość N dla obrazów monochromatycznych (wielopoziomowych) to 255 (obrazy bajtowe).

Istnieją także inne sposoby tworzenia obrazów cyfrowych, nie będących bezpośrednio odbiciem naturalnych obrazów analogowych. Są to obrazy powstające w systemach grafiki komputerowej, edytorach tekstu, bądź też rekonstruowane w systemach tomograficznych. Są to tzw. sztuczne obrazy cyfrowe. Reprezentacja obiektów przedstawionych w obrazie zawiera

zazwyczaj trzy kategorie danych: dane geometryczne (określa położenie i kształt składowych obiektu w przestrzeni o ustalonym układzie współrzędnych), topologiczne (pewne relacje pomiędzy składowymi obiektu - np. kolejność wierzchołków wielokąta na płaszczyźnie), atrybuty (różne własności składników obiektu, np. przezroczystość, chropowatość itp.). Ponadto wyróżnia się dwie zasadnicze metody reprezentacji obiektów: brzegowe i objętościowe. Używa się różnych elementów podstawowych konstruujących obraz. Przykładowo mogą to być wielokąty, wielomiany, krzywe Beziera, funkcje sklepane itd.

Obrazy medyczne, jakkolwiek przedstawiają naturalne obiekty o regularnych kształtach, to jednak tworzone są z wykorzystaniem innego rodzaju nośnika informacji niż światło widzialne (promieniowanie rentgenowskie, ultradźwięki, itp.) lub też specjalnych technologii (np. endoskopia z obrazowaniem za pomocą wziernika z układem optycznym i własnym źródłem światła) umożliwiające obserwację niewidocznych struktur wewnętrznych ludzkiego ciała. Pewna cecha charakterystyczna tkanek z penetrowanego obszaru (oporność akustyczna, poziom osłabienia promieniowania rentgenowskiego, zdolność absorpcji kontrastu, itd.) musi zostać zobrazowana w odpowiednio skontrastowanym polu obrazu wyświetlanego na ekranie monitora urządzenia obrazującego. W coraz większym stopniu medyczne systemy obrazowania wykorzystują cyfrowe systemy akwizycji. Przykładowo, planarne obrazy rentgenowskie rejestrowane są w systemach cyfrowej radiografii przy pomocy wysokiej jakości układów detekcyjnych (np. pixel detectors, imaging plates) lub też analogowe obrazy z klisz rentgenowskich przetwarzane są do postaci cyfrowej w systemach opartych na skanerach laserowych wysokiej klasy. Ten rodzaj obrazów z racji na odmienny sposób tworzenia obrazów struktur niewidocznych będziemy nazywać rekonstrukcyjnymi obrazami cyfrowymi.

Sekwencje obrazów

Filmy telewizyjne, wideo, dane z kamer przemysłowych w systemach monitoringu, analizy scen w stereoskopowym układzie kamer, systemach wizji komputerowej (ang. computer vision), a także wiele innych to informacja reprezentowana w formie sekwencji obrazów. W przypadku zastosowań medycznych taki rodzaj danych to zbiory z kolejnymi przekrojami obrazowanych narządów np. z tomografii rentgenowskiej, sekwencje czasowe poszczególnych warstw (z badań MR), dwuwymiarowe sinogramy w badaniach scyntygraficznych itp. Cechą charakterystyczną tych zbiorów jest fakt, iż zawierają one często wiele obrazów dwuwymiarowych silnie skorelowanych ze sobą, tzn. wiele treści zawartej w przestrzeni obrazu (obiekty, tło, relacje przestrzenne, itp.) powtarza się w kolejnych kadrach. Fakt ten można i należy wykorzystać w konstruowaniu efektywnych metod kompresji, przy czym przedmiotem optymalizacji są modele obiektów występujących w obrazie, ich cechy (kształt, tekstura), ruch, wzajemne relacje, itp.

Dane mieszane

Występują czasami zbiory danych, które zawierają różnego typu informacje. Dobrym przykładem mogą być bazy danych dużych ośrodków medycznych, które zawierają zbiory z wynikami kompleksowych badań pacjentów. Obok danych tekstowych przedstawiających opis choroby, charakterystykę pacjenta itp., a także koncepcję terapii i prowadzonych badań, znajdują się tam różnorodne wyniki badań: ciągi danych pomiarowych, np. EKG, pojedyncze obrazy, np. rentgenogramy płuc, a także sekwencje obrazów tomograficznych, np. z MR, zarówno sekwencje różnych warstw organów, jak też czasowych funkcji dynamicznych poszczególnych warstw. Czasami istnieje też potrzeba zapisania dźwięku, np. z badania USG. Archiwizacja tak złożonych struktur danych wymaga konstrukcji odpowiedniego formatu zapisu danych, który będzie na tyle uniwersalny, że pozwoli efektywnie stworzyć ich

kompleksową reprezentację. Wykorzystuje on kilka różnych technik kompresji właściwych poszczególnym typom danych. Przykładem takiego formatu danych jest stale rozwijany sposób zapisu danych medycznych w standardzie DICOM (ang. Digital Imaging and Communications in Medicine) [1].

Dane multimedialne

Jest to szczególny rodzaj danych mieszanych, gdzie występują długie sekwencje obrazów z przewagą scen o ograniczonej informacji, skojarzone z dźwiękiem. Warunki użytkowania nakładają przy tym dodatkowe wymagania na sposób przysyłania i prezentacji danych, a więc pośrednio na metodę kompresji. Użyteczny system archiwizacji i transmisji danych multimedialnych winien zapewnić możliwość interakcji, pracy w czasie rzeczywistym przy zmieniającej się przepływności sieci, odpowiedni poziom zabezpieczenia przed błędami transmisji, sprawne indeksowanie i przeszukiwanie bazy multimedialnej, itp.

Podstawowe pojęcia

Kompresja danych

Kompresją danych nazywany jest proces przekształcenia pierwotnej reprezentacji zbioru danych w inną reprezentację o mniejszej liczbie bitów, a odwrotny proces rekonstrukcji oryginalnego zbioru danych na podstawie reprezentacji skompresowanej nazywany jest dekompresją. Cele kompresji w zależności od charakteru danych i zastosowań mogą być różnorodne. Zazwyczaj przy projektowaniu metody kompresji chodzi jednak o uzyskanie największej efektywności, przy czym efektywność ta może być rozumiana rozmaicie.

Można wyróżnić dwie zasadnicze kategorie metod kompresji danych: bezstratne i stratne. W **kompresji bezstratnej** (inaczej odwracalnej) zrekonstruowany po kompresji zbiór danych jest numerycznie identyczny ze zbiorem oryginalnym z dokładnością do pojedynczego bitu. Ten rodzaj kompresji jest oczywiście pożądany w zastosowaniach bezwzględnie wymagających wiernej rekonstrukcji zbioru oryginalnego.

Zwykle w procesie kompresji występują dwie kolejne fazy, które odnoszą się do całego zbioru lub poszczególnych jego części. W fazie **modelowania** tworzona jest pewna pośrednia reprezentacja oryginalnego zbioru danych, która jest następnie efektywnie kodowana. Pierwsza faza, wykorzystując pewne metryczne zależności w oryginalnej przestrzeni danych charakteryzowanej jako źródło informacji o określonym alfabetcie i strukturze symboli, transformuje wartości danych w pewne obiekty, które można traktować jako symbole nowego źródła, często o zupełnie innym alfabetcie i strukturze. Nowe źródło ma lepsze własności statystyczne (globalne, lokalne, model prawdopodobieństw występowania pojedynczych symboli bądź ich sekwencji), które następnie wykorzystuje się w drugiej fazie **binarnego kodowania**. Oba odwzorowania są odwracalne w bezstratnych metodach kompresji.

Przykładem wyjaśniającym obie fazy procesu kompresji jest prosty algorytm *kodowania długości sekwencji* (ang. RLE - run length encoding). Niech zbiór danych w reprezentacji oryginalnej O będzie następujący: $O = \{5, 5, 5, 4, 4, 11, 11, 11, 11, 11, 7\}$. Jeśli za algorytm fazy modelowania przyjmiemy proces zastąpienia kolejnych jednakowych symboli w strumieniu danych liczbą ich kolejnych wystąpień oraz wartością symbolu, wówczas pośrednia reprezentacja zbioru O jest następująca: $M = \{3 \times 5, 2 \times 4, 5 \times 11, 1 \times 7\}$. Jeśli przyjmiemy teraz zasadę, iż ta pośrednia reprezentacja jest kodowana binarnie tak, że na pierwszych czterech bitach zapisywana jest liczba wystąpień danego symbolu, a na

następnych czterech wartości tegoż symbolu, wówczas binarna sekwencja wyjściowa K wygląda następująco: $K=\{00110101, 00100100, 01011011, 00010111\}$.

Przy okazji można zauważyć, że długość oryginalnej reprezentacji danych wynosząca 44 bity (przy założeniu 4-bitowej reprezentacji pojedynczego symbolu) została zmniejszona do 32 bitów nowej, skompresowanej reprezentacji.

W **kompresji stratnej** (inaczej nieodwracalnej) zazwyczaj transformuje się zbiór danych w zupełnie nową przestrzeń pośrednią, w której nadmiarowość reprezentacji oryginalnej jest znacznie zredukowana. Faza modelowania kończy się kwantyzacją zbioru uzyskanych wartości, który redukuje alfabet binarnie kodowanego strumienia dając znaczne oszczędności w długości kodu wyjściowego. Kwantyzacja jest procesem nieodwracalnym, a więc niemożliwą jest pełna rekonstrukcja oryginalnego zbioru danych w procesie dekompresji. Odtwarzany zbiór danych jest jedynie przybliżeniem oryginału - możliwe są nawet znaczne różnice w poszczególnych wartościach z zachowaniem jednak ogólnego charakteru danych, w wyniku czego można osiągać znacznie wyższe stopnie kompresji niż w technikach odwracalnych. Większa kompresja osiągana jest zwykle kosztem mniejszej zgodności ze zbiorem danych oryginalnych.

W przypadku kompresji obrazów wprowadza się też czasami pojęcie wizualnej bezstratności w kontekście zastosowań metod stratnych. Eliminacja części informacji z obrazu może być niezauważalna dla obserwatora w normalnych warunkach prezentacji obrazu. Przykładowo, dla danych obrazowych o 8- bitowej dynamice i 6-cio bitowego przetwornika karty graficznej używanej do przeglądania obrazów, usunięcie informacji zawartej w dwu najmłodszych bitach danych nie spowoduje żadnych zmian w obserwowanym obrazie. Definicja wizualnej bezstratności jest jednak subiektywna i trzeba zachować ostrożność przy konkretnych aplikacjach. Wystarczy bowiem zmiana warunków obserwacji obrazu, np. zmiana palety, albo użycie danych skompresowanych do rejestracji obrazu na filmie, bądź też poddanie ich przetwarzaniu (analiza obrazu, eliminacja szumów, obliczanie parametrów ilościowych z obrazu), by wystąpiła zauważalna różnica pomiędzy obrazem bez dwóch najmłodszych bitów i obrazem oryginalnym.

Efektywność kompresji

Efektywność kompresji może być rozumiana w różnoraki sposób w zależności od rodzaju skompresowanych danych, zastosowania, sprzętowych możliwości implementacji itp. Pierwszym, najbardziej powszechnym rozumieniem tego pojęcia jest zdolność do maksymalnego zmniejszenia rozmiaru nowej reprezentacji skompresowanych danych w stosunku do rozmiaru zbioru pierwotnego. Do liczbowych miar tak rozumianej efektywności należą przede wszystkim: stopień kompresji CR (ang. compression ratio), procent kompresji CP (ang. compression percentage) oraz średnia bitowa BR (ang. bite rate).

Stopień kompresji wyrażany jest przez stosunek liczby bitów reprezentacji obrazu oryginalnego do liczby bitów reprezentacji obrazu skompresowanego, procent kompresji określany jest przez wyrażenie $CP = (1 - \frac{1}{CR}) \cdot 100\%$, a średnia bitowa BR jest średnią ilością bitów skompresowanej reprezentacji danych przypadającą na element oryginalnego zbioru danych.

Efektywność czy skuteczność metod kompresji to najczęściej zdolność do osiągnięcia w procesie kompresji możliwie dużych wartości CR lub CP, czy też możliwie małej średniej bitowej BR. W pewnych zastosowaniach, np. w urządzeniach do rejestracji danych w czasie rzeczywistym, efektywność kompresji może być pojmowana jako zdolność do minimalizacji czasu kompresji (lub też kompresji/dekompresji). Innymi kryteriami efektywności mogą być:

- iloczyn: czas \times średnia bitowa;
- rodzaj powstających zniekształceń przy kompresji stratnej i ich poziom przy danym CR;
- iloczyn: miara zniekształceń \times średnia bitowa;
- odporność tworzonego strumienia skompresowanych danych na zakłócenia w czasie transmisji, szacowana przy pomocy odpowiednich miar;
- zdolność do szybkiej transmisji progresywnej (tzn. informacja o przesyłanym zbiorze w pierwszej kolejności jest całościowa i dotyczy ogólnego charakteru danych, a następnie informacje na temat poszczególnych fragmentów są uszczegóławiane - na tym samym poziomie w całym obrazie, aż do wiernego odtworzenia najdrobniejszych szczegółów w etapie końcowym), optymalizowana pod kątem możliwie dużej koncentracji informacji na bit danych przesyłanych w pierwszej kolejności (chodzi o efektywne upakowanie informacji na początku transmisji);
- zdolność do uzyskania w procesie kompresji założonej z góry wartości CR, aby algorytm pozwalał kompresować zbiory danych w stopniu, którego wartość możliwie dokładnie odpowiada założonej wartości CR; bardziej efektywny algorytm pozwoli więc na dokładniejsze sterowanie długością pliku wyjściowego, dopasowując się np. do zmiennej przepustowości łącz;
-

Krótki zarys historii technik kompresji

Pierwszym kluczowym wydarzeniem jest położenie fundamentów teorii informacji przez Claude Shannona, jednego z największych naukowców XX wieku [2]. Chodzi tu o prace Shannona z końca lat czterdziestych stulecia (szczególnie prace z roku 1948). Sformułowane tam pojęcia entropii jako miary informacji, nadmiarowości, modeli źródeł informacji, itp. stanowią zręby współczesnych technik kompresji danych. Shannon przyczynił się do powstania skutecznego algorytmu kodowania opartego na statystycznej analizie zbioru skompresowanych danych, zwanego algorytmem Shannona-Fano.

Kolejnym, bardzo istotnym wydarzeniem było opublikowanie w 1952r. przez D.A. Huffmana pracy "A method for the construction of minimum-redundancy codes" [3]. Przedstawia ona algorytm tworzenia optymalnej reprezentacji kodowej dla zbioru danych przy założeniu przyporządkowania każdemu symbolowi alfabetu źródła modelującego ten zbiór danych oddzielnego słowa kodowego o długości (w bitach) w przybliżeniu odwrotnie proporcjonalnej do prawdopodobieństwa wystąpienia tego symbolu w strumieniu danych wejściowych. Metoda ta była dominującą aż do lat osiemdziesiątych w dziedzinie bezstratnych metod kompresji. Ponieważ metoda Huffmana dotyczy zasadniczo sposobu realizacji drugiej fazy schematu kompresji - binarnego kodowania, podejmowano szereg prób poprzedzania kodowania Huffmanowskiego różnymi technikami predykcyjnej redukcji nadmiarowości oryginalnej reprezentacji zbioru danych.

W tym samym okresie (lata 60-te i 70-te) opracowano szereg technik kompresji stratnej, która polegała na wydobyciu i uwypukleniu pewnych istotnych (tzn. szczególnie ważnych z punktu widzenia zawartej informacji) cech (tj. tekstur, konturów, niektórych struktur czy składowych) zbioru danych i zapobieżeniu ich degradacji w procesie stratnej kompresji. Metody te zwane ekstrakcyjnymi były stosowane głównie do kompresji obrazów medycznych. Pozostawiano w obrazie wszelką informację istotną z punktu widzenia procesu diagnozy i terapii kodując odpowiednie segmenty obrazu w sposób odwracalny, podczas gdy

pozostałe obszary obrazu zawierające informacje nieistotne w opinii specjalistów archiwizowano przy pomocy metod silnej kompresji stratnej. Kompresja była niekiedy połączona z tworzeniem cyfrowej postaci obrazów analogowych. Przykładowo, na kliszy rentgenowskiej lekarz zaznaczał obszar istotny diagnostycznie, który następnie podlegał skanowaniu i bezstratnej kompresji.

W latach 1977 i 1978 panowie Lempel i Ziv opublikowali dwa algorytmy bezstratnej kompresji, które stały się podstawą nowej grupy technik tzw. słownikowego kodowania [4,5]. Metody te charakteryzuje dość rozbudowana faza modelowania, polegająca na budowaniu czy też określaniu słownika będącego podstawową strukturą wykorzystywaną w procesie tworzenia wyjściowej sekwencji kodowej. Dość skromna faza binarnego kodowania sprowadza się do zapisania kolejnych indeksów fraz słownika odpowiadających sukcesywnie analizowanym danym strumienia wejściowego. Od pierwszych liter nazwisk autorów oraz lat publikacji algorytmy te nazwano LZ 77 i LZ 78. Pomimo dość jasnej koncepcji, doczekały się one praktycznej realizacji dopiero w 1984r autorstwa T. Welcha - algorytm LZW [6]. O popularności i dużej efektywności tych metod niech świadczy fakt, iż stały się one podstawą tak znanych i powszechnie stosowanych kompresorów jak : Unix_Compress, ARC, PKZIP, LHarc, ARJ.

W latach osiemdziesiątych nastąpił gwałtowny rozwój technik kompresji. Na uwagę zasługują prace nad coraz doskonalszymi modelami adaptacyjnymi implementowanymi w różnych metodach, opracowanie skutecznych algorytmów kodowania arytmetycznego [7,8], zasadniczo najefektywniejszej obecnie metody bezstratnej kompresji (stale modyfikowanej np. [9]), a także rozwój metod kodowania transformacyjnego do kompresji stratnej. Podejmowano próby wykorzystania różnych transformat, np. Fouriera, Walsha-Hadamarda, sinusową, Karhunen-Loevego uzyskując najlepsze rezultaty dla dyskretnej transformaty kosinusowej DCT (ang. discrete cosine transform). Znalazło to odbicie w pierwszych standardach do kompresji obrazów cyfrowych wielopoziomowych opracowanych na bazie blokowej DCT na początku lat dziewięćdziesiątych - standardy JPEG i MPEG.

Koniec lat osiemdziesiątych i początek lat dziewięćdziesiątych to pojawienie się dwu nowych, efektywnych technik stratnej kompresji obrazów opartych na dość złożonym aparacie matematycznym. Pierwsza to metoda wykorzystująca przekształcenia fraktalne, pozwalająca uzyskać dużą skuteczność kompresji szczególnie dla obrazów naturalnych o niezbyt złożonej treści. Istotne zasługi w rozwoju tej techniki położyli między innymi Barnsley, Jacquin, Hurd. Drugą techniką, z którą związane są takie nazwiska jak: Mallat, Daubechies, Villasenor, Vetterli, Strang i inni, jest metoda transformacji falkowej (ang. wavelet transform) należąca do szerszej grupy technik kodowania pasmowego (ang. subband coding). Metody falkowe desygnowane do kompresji sygnałów niestacjonarnych mają szereg zalet, z których chyba najważniejszą jest możliwość łatwej i szybkiej adaptacji do konkretnego charakteru zbioru danych w wymiarze skali i przestrzeni, co pozwala zwiększyć skuteczność kompresji obrazów do wartości często niemożliwych do uzyskania przy pomocy innych technik.

W drugiej połowie lat dziewięćdziesiątych daje się zauważyć rosnącą dominację technik kompresji obrazów statycznych, które wykorzystują przekształcenia falkowe do dekompozycji danych oryginalnych. Świadczą o tym kolejne modyfikacje algorytmu EZW (ang. embedded zerotree wavelet) autorstwa J. M. Shapiro [10], jak również coraz bardziej złożone metody kwantyzacji i kodowania współczynników falkowych, pozwalające blisko dwukrotnie zwiększyć skuteczność kompresji (dwa razy mniejsza długość kodu przy tym samym poziomie zniekształceń) w stosunku do standardu JPEG. Znalazło to wyraz w pracach nad nowym standardem kompresji JPEG 2000 opartym na koncepcji falkowej dekompozycji

obrazów, elastycznym kształtowaniu strumienia danych kompresowanych w zależności od potrzeb użytkownika, oraz kompresji stratnej-do-bezstratnej (ang. lossy-to-lossless), pozwalającej w fazie początkowej na efektywną kompresję stratną, która przechodzi ostatecznie w formę kompresji odwracalnej po dołączeniu wszystkich informacji do pliku wyjściowego kodera.

Trudności w aplikacji metod falkowych do kompresji sekwencji obrazów powodują, że obecnie stosowane kodery najczęściej wykorzystują transformatę DCT z blokową estymacją i kompensacją ruchu ze standardów kompresji sekwencji obrazów i skojarzonego dźwięku rozwijanych od początku lat dziewięćdziesiątych (H.261, MPEG-1, MPEG-2, H.263). Ponadto, prace nad algorytmami kompresji drugiej generacji (z obiektową analizą scen w sekwencji obrazów i bardziej elastyczną kompensacją ruchu) są podstawą konstrukcji nowych standardów kompresji danych multimedialnych – MPEG-4 i opisu danych multimedialnych przy pomocy różnego typu deskryptorów – MPEG-7. W roku 2000 rozpoczęto prace nad nowym standardem MPEG-21, który ma być ‘wielkim obrazem’ ogromnej infrastruktury wymiany i konsumpcji treści multimedialnych, który uwzględni mnogość istniejących już i rozwijanych narzędzi, określając ich wzajemne relacje i porządkując całą przestrzeń multimediiów. Aktualny stan prac nad standardami JPEG i MPEG można monitorować na stronach odpowiednio [11] i [12].

Kilka słów o praktycznej realizacji algorytmów kompresji

Różne pomysły konstruowania efektywnych metod kompresji, jakkolwiek oparte często na słusznym założeniach i oczekiwaniach, natrafiają jednak niekiedy na duże problemy aplikacyjne, co czyni je w wielu przypadkach nieprzydatnymi. Wydaje się, że w dziedzinie kompresji danych szczególnie istotne w ocenie różnych koncepcji jest ich praktyczna przydatność i możliwości realizacyjne.

Bez względu na narzędzie (C, C++, Java, makroasembler, itd.) i sposób realizacji (programowa, sprzętowa) algorytmu kompresji istnieją pewne podstawowe problemy, jakie należy rozwiązać przy konstrukcji własnego kodera.

Konstrukcja od początku

Można rozpoczynać budowę kodera od początku, tj, od procedur realizujących podstawowe operacje bitowe i bajtowe na strumieniu wejściowym, czytanie i zapisywanie strumienia, przeszukiwanie, liczenie częstości wystąpień poszczególnych symboli, itd. Aby zrealizować nawet najprostszy koder bitowy wygodnie jest zbudować proste struktury danych, które okażą się bardzo pomocne także przy realizacji dużo bardziej złożonych algorytmów. Ważną sprawą jest dobre zorganizowanie operacji we/wy działających na pojedynczych bitach, bowiem zdecydowana większość technik misternie konstruuje efektywny kod składając z pojedynczych bitów czy też sekwencji bitów słowa i sekwencje kodowe zapisując lub transmitując paczki bitów o zmiennej długości.

Przykładowa struktura definiująca typ zbioru, na którym można dokonywać operacji bitowych wygląda następująco (przyjęto konwencję zapisu według języka C):

```
typedef struct zbior_bit {
    FILE *zbior;
    BYTE maska;
    int pamiec;
} ZBIOR_BIT;
```

Struktura taka obok wskaźnika do zbioru, na którym będą wykonywane bajtowe operacje zawiera element `maska`, w którym przesuwany jest wskaźnik położenia aktualnie ustawianego bądź czytanego bitu. W chwili początkowej ustawiana jest w nim wartość `0x80` i po operacji na najstarszym bicie jedynek jest przesuwana o jedną pozycję w prawo. Efekt operacji na bicie jest z kolei umieszczany w elemencie `pamiec`. Jest to element typu `int` ze względu na łatwość posługiwania się zestawem funkcji: `int =getc(FILE)` i `putc(int,FILE)` do czytania i pisania kolejnych bajtów. W chwili początkowej bufor ten jest wyzerowany. W przypadku operacji pisania do pliku bitowego kilku bitów do `pamiec` wchodzi kolejne bity na miejsca wskazywane przez pozycję jedynek w `maska`. W momencie kiedy określony jest najmłodszy bit następuje zapisanie całego bajtu do pliku `zbior`, wpisanie do `maska` wartości `0x80` oraz wyzerowanie `pamiec`. Proces pisania kolejnych bitów do pliku może być więc kontynuowany.

Operacja czytania bitów ze zbioru przebiega analogicznie. W chwili początkowej wczytywany jest do `pamiec` nowy bajt z pliku `zbior`. Elementy `maska` i `pamiec` są ustawiane tak samo jak przy pisaniu. Przy czytaniu najstarszy bit z `pamiec` jest kopiowany na wyjście, a jedynek w `maska` jest przesuwana o jedną pozycję w kierunku bitów młodszych. Następnie czytany jest kolejny bit z `pamiec` itd. W momencie odczytania najmłodszego bitu do `pamiec` wczytywany jest kolejny bajt ze `zbior`, a w `maska` ustawiana jest ponownie jedynek na najstarszej pozycji.

Struktura `ZBIOR_BIT` może zawierać dodatkowe elementy, pomocne przy realizacji algorytmów kompresji. Bardzo przydatnym okazuje się często licznik wczytanych czy zapisanych bajtów. Umożliwia on kontrolę strumienia danych poprzez np. generację znaczników utworzenia kolejnej porcji kodu wyjściowego, dzielenie tego strumienia na założone z góry części itd. Bardzo istotną aplikacją takiego licznika są kodery tworzące sekwencję kodową o założonej wstępnie długości - dotyczy to najczęściej metod stratnego kodowania danych. Proces wysyłania kolejnych porcji informacji zostaje przerwany w momencie osiągnięcia założonego rozmiaru kodu wyjściowego. Taka kontrola rozmiaru strumienia danych umożliwia również konstrukcję systemów przesyłania danych o określonej szybkości.

Innym częstym problemem w konstrukcji algorytmów kodowania jest budowa struktury drzewa, która jest pomocna przy konstruowaniu kodu Huffmana (zobacz rozdz. 3), budowaniu statystycznego modelu prawdopodobieństw warunkowych czy też słownika do szybkich przeszukiwań. Szczególnie istotną wydaje się efektywna struktura drzewa książki kodowej w wektorowej kwantyzacji, która może być nie tylko szybko przeglądana, ale i modyfikowana. Takich przykładów jest znacznie więcej. Wydaje się więc, że ze względu na aspekty skutecznej implementacji jest to jedno z najważniejszych zagadnień.

Przykładowo najprostsza struktura węzła drzewa binarnego, stosowanego w statycznym koderze Huffmana, może wyglądać następująco:

```
typedef struct wezel {
    UINT waga;
    int dziecko_0;
    int dziecko_1;
} WEZEL;
```

Zawiera ona wskaźniki do dwu bezpośrednich węzłów podrzędnych - dzieci oraz wagę danego węzła, która jest czynnikiem decydującym o położeniu tego węzła w strukturze drzewa. Struktura ta może zostać uzupełniona poprzez dodatkowe elementy: wskaźnik do bezpośredniego węzła nadrzędnego - rodzica oraz symbol skojarzony z tym węzłem w przypadku, gdy węzeł ten jest węzłem zewnętrznym drzewa - liściem.

Struktura opisująca całe drzewo w adaptacyjnym koderze Huffmana może być zdefiniowana następująco:

```
typedef struct drzewo {
    int liscie[liczba_symboli];
    int wolny_wezel;
    struct wezel {
        UINT waga
        int rodzic;
        BOOL wezel_to_lisc;
        int dziecko;
    } WEZLY[liczba_symboli*2-1];
} DRZEWO;
```

Węzeł w tym drzewie opisany jest tylko poprzez jeden wskaźnik do węzła potomnego. Wynika to z właściwości binarnego drzewa Huffmana, że nigdy powyżej korzenia nie występują pojedyncze węzły, lecz zawsze wszystkie węzły wewnętrzne (pomiędzy korzeniem a liśćmi) i zewnętrzne występują w parach mając węzły lustrzane. Stąd też przyjmuje się, że np. na pozycji nieparzystej występuje pierwszy węzeł, a na parzystej drugi węzeł lustrzany. Stąd też mając wskaźnik do jednego dziecka jednocześnie wiemy, gdzie znajduje się drugie.

W strukturze drzewa pojawiły się obok tablicy z węzłami dwa dodatkowe elementy: wskaźnik do pierwszego wolnego węzła (jest to wersja adaptacyjna, a więc drzewo jest sukcesywnie rozbudowywane poprzez dodawanie kolejnych węzłów) oraz tablica wskaźników do liści drzewa. Wprowadzenie dodatkowej struktury znacznie ułatwi wyszukiwanie liści drzewa do potrzeb kodowania czy porównywania. Są one bowiem rozrzucone po całej tablicy węzłów w zależności od tego, kiedy pierwszy raz pojawiły się w kodowanym strumieniu. Wtedy bowiem zostały wprowadzone dopiero do struktury drzewa na końcu aktualnej listy węzłów.

Ze struktury drzewa korzysta się także przy realizacji kodera słownikowego (patrz rozdz. 5), aby zwiększyć prędkość przeszukiwań. Z kolei koder arytmetyczny (patrz rozdz. 4) wymaga optymalnej struktury przechowującej model statystyczny prawdopodobieństw warunkowych, tym większy, im wyższy rząd modelu Markowa opisuje charakter kodowanego strumienia. Wykorzystuje się tutaj techniki mieszania (ang. hashing), by przyspieszyć algorytm oraz zaoszczędzić ilość pamięci koniecznej do realizacji takiej adaptacyjnej struktury. Wszystkich zainteresowanych głębszą analizą praktycznych aplikacji algorytmów kompresji, szczególnie bezstratnej, odsyłam do pozycji M. Nelsona [13].

W przypadku kompresji stratnej istotnym czynnikiem przy konstrukcji kodera jest możliwie skuteczna realizacja metody dekompozycji (transformacji) danych oryginalnych, odpowiednio szybka i jednocześnie dokładna. Czasochłonny w niektórych rozwiązaniach jest także algorytm kwantyzacji (szczególnie w koderach wektorowej kwantyzacji, niektórych koderach falkowych i DCT). Na problemy te zwrócono uwagę w kolejnych rozdziałach niniejszego opracowania.

Konstrukcja na bazie użytecznych pakietów

Ze względu na dużą złożoność zaawansowanych metod kompresji i niebanalnych rozwiązań optymalizacji czasowej i sprzętowej poszczególnych algorytmów wygodniej jest posługiwać się gotowymi procedurami poszczególnych elementów schematów kompresji, których wcale niemało pojawiło się w ostatnich latach w internecie jako oprogramowanie *freeware*-owe (idea Free Software Foundation: <http://www.gnu.org/fsf/fsf.html>) do celów

eksperymentalnych, badawczych i edukacyjnych. Z procedur tych, jak z cegiełek można budować różne wersje koderów (zapewniając oczywiście odpowiednie sprzężenie pomiędzy nimi), optymalizować klasyczne sposoby kompresji danych, wstawiać bloki dodatkowe, zamienne, wykonywać szereg testów porównawczych na różnego typu koderach, itd.

Bardzo zachęcam do korzystania z takiej formy budowania własnych koderów i poszukiwania optymalnych rozwiązań zagadnienia kompresji różnego typu danych. Poniżej przedstawiono kilka takich zbiorów oprogramowania do kompresji. Warto zwrócić uwagę na fakt, iż filozofia otwartego oprogramowania przyświeca także twórcom nowego standardu JPEG2000. Część piąta tego standardu będzie zawierać referencyjne oprogramowanie w C i Javie kodera i dekodera, co zapewni z jednej strony jego popularność i powszechność, z drugiej zaś strony wpłynie na bardziej świadome wykorzystanie jego zalet i przyczyni się do jego rozwoju, dalszej optymalizacji w skali niemal globalnej. Taka filozofia rozwoju technik i standardów kompresji jest bardzo bliska Autorowi tej pozycji.

Wybrane źródła oprogramowania do kompresji danych i szereg cennych uwag aplikacyjnych można znaleźć między innymi w:

- QccPack: Quantization, Compression, and Coding Library & Utilities: <http://qccpack.sourceforge.net/>
- The Data Compression Library: <http://www.dogma.net/DataCompression/>
- Wavelet Image Compression Construction Kit: <http://www.cs.dartmouth.edu/~gdavis/wavelet/wavelet.html>
- JJ2000 – An implementation of Jpeg2000 standard in Java: <http://jj2000.epfl.ch/>
- Jasper - JPEG2000 codec: <http://www.imagepower.com/products/ubcform.htm>

Bibliografia

1. Dicom standard: <http://medical.nema.org/dicom.html>.
2. N. Sloane and A. Wyner (eds.), *Claude Elwood Shannon : collected papers* (New York, 1993).
3. D. A. Huffman, *A Method for the Construction of Minimum Redundancy Codes*, Proc. IRE, 40:1098-1101, 1952.
4. J. Ziv, and A. Lempel, *A Universal Algorithm for Sequential Data Compression*, IEEE Trans. Information Theory, 23(3):337-343, 1977.
5. J. Ziv J., and A. Lempel, *Compression of Individual Sequences via Variable-Rate Coding*, IEEE Trans. Information Theory, 24(5):530-536, 1978.
6. T. Welch, *A Technique for High-Performance Data Compression*, IEEE Computer, 17(6):8-19, 1984.
7. G. Langdon, *An Introduction to Arithmetic Coding*, IBM J. Res. Dev., 28(2):135-149, 1984.
8. I. Witten, R. Neal, and J. Cleary, *Arithmetic Coding for Data Compression*, Communications of the ACM, 30(6):520-540, 1987.
9. P. G. Howard, J. S. Vitter, *Arithmetic Coding for Data Compression*, Proc. IEEE, 82(6):857-865, 1994.

10. J. M. Shapiro, *Embedded Image Coding Using Zerotrees of Wavelet Coefficients*, IEEE Trans. Signal Proces., 41(12): 3445-3462, 1993.
11. <http://www.jpeg.org/>
12. <http://www.cselt.it/mpeg/>
13. M. Nelson, *The Data Compression Book*, M&T Books, 1991.