

WSTĘP DO STANDARDU JPEG2000

Artur Przelaskowski i Dominik Rives

1. NOWE WZORCE ALGORYTMÓW KOMPRESJI (PARADYGMATY)

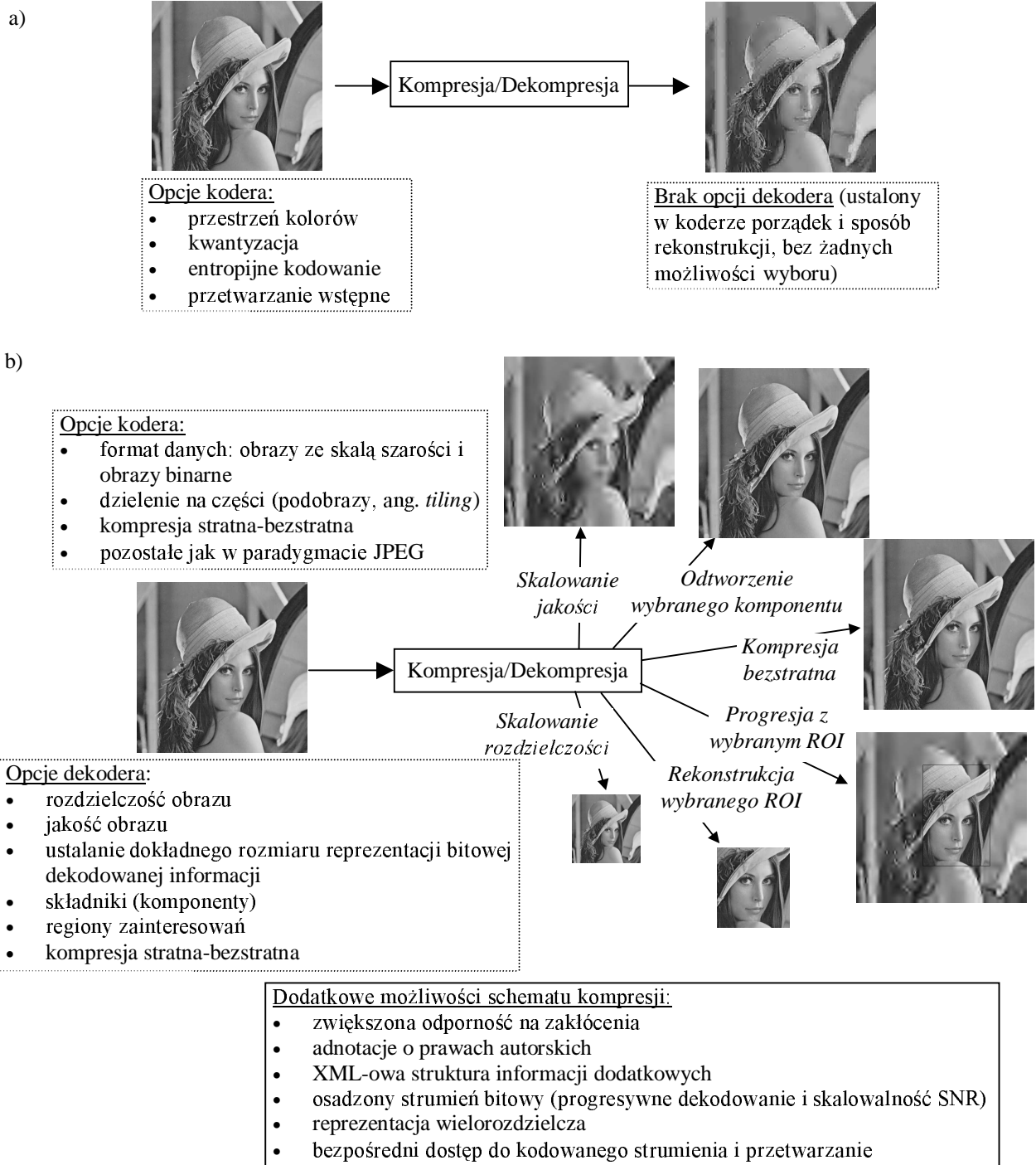
Zdolność uzyskiwania możliwie krótkiej reprezentacji danych obrazowych była dotychczas jednym z najważniejszych kryteriów decydujących o przydatności techniki kompresji. Przy obecnym rozwoju nowych technologii multimedialnych równie istotne okazały się nowe cechy funkcjonalne systemów kompresji-dekompresji, powodujące dużą użyteczność algorytmów w znacznie szerszym obszarze zastosowań. Wiele postulatów stawianych przez rosnący krąg potencjalnych użytkowników jest praktycznie nie do zrealizowania za pomocą funkcjonujących dotąd standardów kompresji obrazów, czyli przede wszystkim JPEG-a. Schemat kompresji wynikający z klasycznego paradygmatu definiuje zbiór dostępnych opcji kodera, które pozwalają z jednej strony dopasować algorytm kompresji do własności strumienia wejściowego (dynamiki danych, przestrzeni kolorów itp.), z drugiej zaś - dobrać parametry procesu kompresji (sposób kwantyzacji - np. tablice, szerokość przedziałów, czy kodowania - np. sekwencyjny, progresywny itd.). Po stronie dekodera standard nie dopuszcza żadnych możliwości wyboru sposobu odtwarzania strumienia - obowiązuje ustalony w koderze porządek i sposób rekonstrukcji, a strumień zakodowany jest rekonstruowany w całości. Realizację paradygmatu schematu kompresji ze standardu JPEG przedstawiono na rys. 1.1.a).

Koncepcja nowego paradygmatu kompresji (rys. 1.1.b)) została sformułowana w okresie intensywnych prac badawczych nad realizacją standardu JPEG2000. W porównaniu z paradygmatem z JPEG-a następuje znaczne rozszerzenie możliwości kształtowania skompresowanego strumienia danych (np. wprowadzanie regionów zainteresowań - ROI, mechanizmu podziału oryginału na części - ang. *tile*) oraz postaci dekompresowanej reprezentacji oryginału (np. sterowanie rozdzielczością, jakością obrazu, bitowym rozmiarem reprezentacji informacji rekonstruowanej).

Współczesny standard kompresji powinien być tworzony z myślą o szerokiej gamie zastosowań, takich jak Internet, różnorodne aplikacje multimedialne, obrazowanie medyczne, kolorowy fax, drukowanie, skanowanie, cyfrowa fotografia, zdalne sterowanie, przenośna telefonia nowej generacji, biblioteka cyfrowa oraz e-komercja (elektroniczna komercja, głównie z wykorzystaniem Internetu). Nowy system kodowania musi więc być skuteczny w przypadku różnych typów obrazów (binarne, ze skalą szarości, kolorowe, wieloskładnikowe) o odmiennej charakterystyce (obrazy naturalne, medyczne, sztuczne obrazy grafiki komputerowej, naukowe z różnych eksperymentów, z tekstem itd.). Ponadto, powinien zapewniać efektywną współpracę z różnymi technologiami obrazowania (akwizycji/generacji/ wykorzystywania obrazów): z transmisją w czasie rzeczywistym, z archiwizacją, gromadzeniem bazodanowym (np. w cyfrowej bibliotece), w strukturze klient/serwer, z ograniczonym rozmiarem bufora czy limitowaną szerokością pasma itp. Wymagania odnośnie nowego standardu kompresji najlepiej charakteryzują stwierdzenia z 'JPEG2000 call for proposals' z marca 1997 r.: *„poszukiwany jest standard dla tych obszarów, gdzie aktualne standardy nie potrafią zagwarantować wysokiej jakości lub wydajności, standard zapewniający nowe możliwości na rynkach, które dotąd nie wykorzystywały technologii kompresji i dający otwarte narzędzia systemowe dla aplikacji obrazowych.”*

Użyteczność falek w modelowaniu i realizacji Użyteczność analizy falkowej leży głównie w możliwościach skutecznego modelowania przy ich pomocy przestrzenno-częstotliwościowej charakterystyki obrazów naturalnych. Typowa postać obrazu jest mieszaniną obszarów o znacznych rozmiarach z treścią niskoczęstotliwościową (wolno zmieniające się tło sceny naturalnej, np. niebo, ściana w pokoju, czy też jednostajne, lekko zaszumione tło w obrazie medycznym), małych obszarów ze znaczącą treścią wysokoczęstotliwościową (silne krawędzie o dużych gradientach, wyraźne, drobne struktury) oraz obszarów (obiektów) pokrytych teksturami. W bazie falkowej znajdują się elementy o skończonym nośniku, które mają dobrą rozdzielczość częstotliwościową (przy słabszej przestrzennej) w zakresie częstotliwości niskich, czyli dobrze charakteryzują tło i wolnozmiennie tekstury, a także elementy z dobrą rozdzielczością przestrzenną (przy słabszej częstotliwościowej) w zakresie częstotliwości wysokich, co daje dobrą lokalizację krawędzi w podpasmach dokładnej skali. Taka korelacja cech bazy z własnościami obrazów naturalnych prowadzi do silnej koncentracji energii

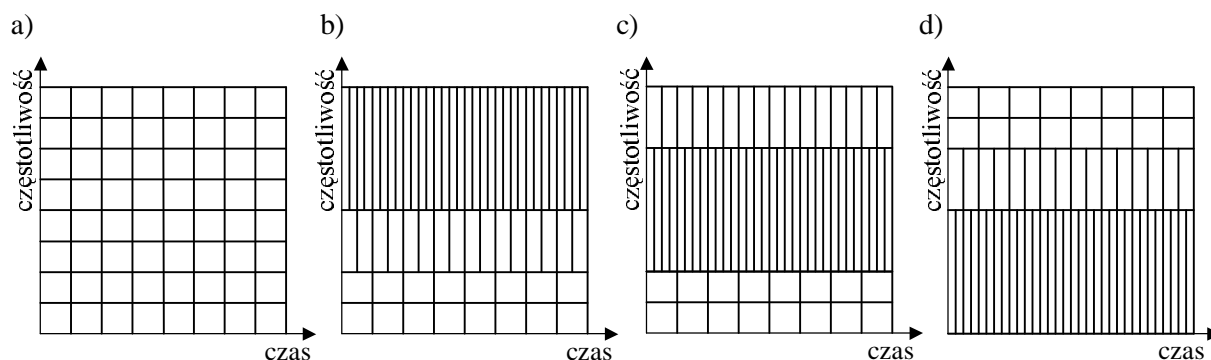
sygnału w niewielkiej liczbie współczynników, przy czym większość informacji obrazu znajduje się w niewielkim obszarze podpasma najniższych częstotliwości dziedziny falkowej dekompozycji. Pozostała część informacji zawarta jest w wartościach współczynników rozrzuconych w niewielkich grupach wokół przestrzennej pozycji silnych krawędzi w obrazach uzupełniających różnych skal. Zaż zdecydowana większość współczynników ma wartości bliskie zeru i może być zupełnie pominięta lub silnie kwantowana w procesie stratnej kompresji, bez znaczącego wpływu na jakość rekonstrukcji. Wyższość wielorozdzielczej analizy falkowej w stosunku do STFT (ang. *Short Time Fourier Transform*) lub blokowej DCT (ang. *discrete cosine transform*) jest tutaj bezdyskusyjna, ponieważ lepiej, w sposób naturalny opisuje rzeczywisty sygnał.



Rys. 1.1. Porównanie paradygmatu schematu kompresji: a) z podstawowej wersji standardu JPEG; b) z nowego standardu JPEG2000 - część I, opartego na falkowej technice kompresji; SNR (ang. *Signal to Noise Ratio*).

Dodatkową zaletą dekompozycji falkowej jest jej podatność na rozwiązania adaptacyjne, dotyczące zarówno stosowania bazy falkowej o nieskończonej ilości możliwych postaci, jak też samego schematu dekompozycji i podziału na podpasma. Adaptacyjną postać transformacji można lepiej dopasować do własności konkretnego obrazu czy grupy obrazów. Gładkość funkcji bazowych, ich rozmiar, symetryczność, decydują o możliwie najlepszym przybliżeniu lokalnych własności obrazu, a ich właściwy wybór wpływa znacząco na skuteczność kompresji. Zagadnienie doboru optymalnej bazy falkowej w kompresji konkretnego obrazu nie jest właściwie rozwiązywalne w sposób jednoznaczny, ale istnieje cały zbiór przesłanek (w postaci wiedzy *a priori*) pozwalających dobrać jądro przekształcenia w sposób prawie optymalny.

Klasyczny schemat wielorozdzielczej dekompozycji Mallata ze strukturą logarytmiczną dobrze opisuje wspomniane własności typowych obrazów z wykładniczo opadającym widmem gęstości mocy. Dla obrazów o nieco innej charakterystyce, zawierających dużą ilość informacji w zakresie wysokoczęstotliwościowym (np. rozległe obszary z teksturą prążkową: czarne pasy na białym tle) bardziej efektywną dekompozycję otrzymuje się za pomocą bazy pozwalającej uzyskać dobrą lokalizację w dziedzinie częstotliwości podpasma wysokoczęstotliwościowych (ze względu na dobre zróżnicowanie zgromadzonej tam dużej ilości informacji). Konieczne jest do tego narzędzie transformacji falkowej, pozwalające dobrać schemat dekompozycji w zależności od cech obrazu (źródła informacji). Choć teoretycznie można zbudować dowolną liczbę schematów dekompozycji, rozwiązując zagadnienie optymalizacji schematu dekompozycji w każdym konkretnym przypadku, to bardziej praktycznym jest rozwiązanie, gdzie można zbudować skończoną bibliotekę reprezentatywnych transformacji, dobierając z biblioteki, dzięki szybkiemu algorytmowi, optymalną postać transformacji dla konkretnego obrazu. Takim narzędziem jest dekompozycja za pomocą pakietu falek (ang. *wavelet packet*), zwana też, bardziej algorytmicznie, schematem wyboru najlepszej bazy (ang. *best basis*). Pakiety falek stanowią dużą bibliotekę transformacji silnie zróżnicowaną pod kątem ich własności dekompozycji przestrzenno-częstotliwościowej, ze zdolnością szybkiego przeszukiwania. Funkcje bazowe pakietu falek mają własności falek klasycznych, przy czym wzbogacone są często o większą liczbę oscylacji. Wykorzystując pakiet falek, można dostosować postać transformacji do praktycznie dowolnego spektrum sygnału – zobacz rys. 1.2. Implementacja dekompozycji z pakietem falek wymaga dodatkowych nakładów obliczeniowych, głównie na proces wyszukania optymalnego schematu transformacji dla danego obrazu, a także przesłania niewielkiej informacji dodatkowej do dekodera. Kosztem większego obciążenia obliczeniowego można niemal dowolnie ‘blisko’ dopasować postać transformacji do sygnału, np. wprowadzając zmienną w czasie segmentację, pozwalając na ewolucję pakietu falek wraz z sygnałem (silnie niestacjonarnym). Z drugiej strony, procedurę takiej transformacji można znacznie uprościć, dobierając ustaloną postać dekompozycji pakietu falek dla danego typu obrazów, jak to ma miejsce np. w standardzie FBI do kompresji obrazów z odciskami palców.



Rys. 1.2. Podział dziedzin czas (przestrzeń)-częstotliwość w różnych schematach dekompozycji: a) dekompozycja równomierna, uzyskana za pomocą pakietu falek lub też STFT, b) klasyczna dekompozycja falkowa Mallata, c) dekompozycja pakietu falek, gdzie szerokość podpasma nie zmienia się ani równomiernie ani logarytmicznie, d) odwrócona dekompozycja falkowa, uzyskana z wykorzystaniem pakietu falek.

Ponadto, atrakcyjną cechą falkowej reprezentacji obrazu, od skal mało dokładnych (tj. skal dużych), z silną koncentracją informacji (energii) na bit danych, do małych skal bardzo dokładnych (z małym przyrostem informacji na bit), jest jej naturalna progresywność. Reprezentacja ta umożliwia

sterowanie kolejnością i rodzajem przekazywanej informacji, przy czym niewielkim kosztem można uzyskać strumień danych (prawie) optymalny w sensie R-D.

Kompresja konstruowana na podstawie dekompozycji falkowej jakby naturalnie odpowiada na większość wymagań stawianych w nowym paradygmacie kompresji, pozwalając je realizować w szybkich algorytmach obliczeniowych tak, że praca koderów w systemach czasu rzeczywistego jest możliwa.

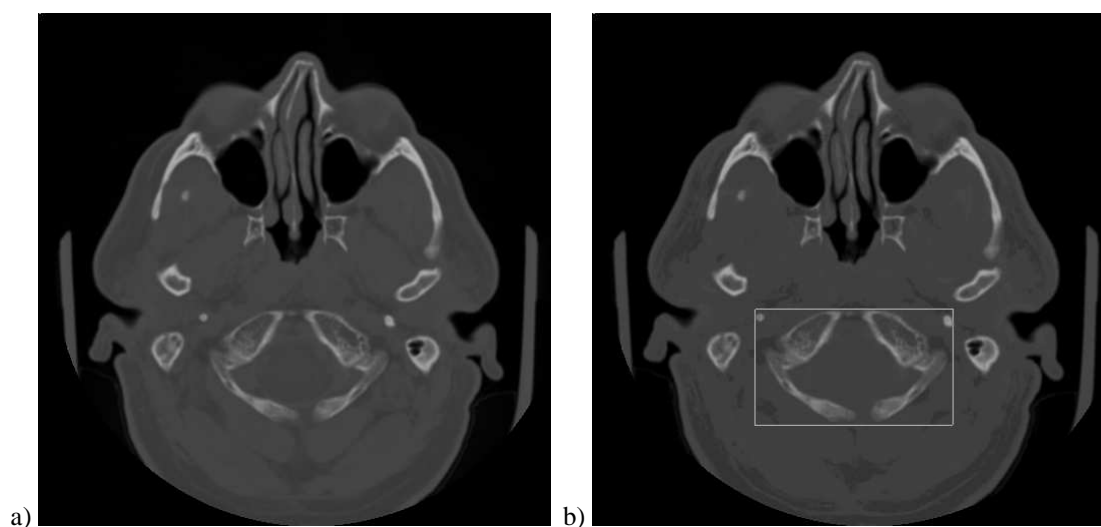
1.1. WYBRANE CECHY ELASTYCZNEGO KODERA FALKOWEGO

W wielu systemach transmisji i gromadzenia informacji bardzo dużego znaczenia nabiera oszczędna reprezentacja danych, która jednocześnie pozwala na szybki dostęp do wyselekcjonowanej informacji oraz na łatwą analizę tej informacji. Ponadto, zapewnia wysoką podatność na techniki przetwarzania uwydatniające informację użytkową w procesie dekompresji, a także małą wrażliwość przesyłanego strumienia danych na zakłócenia (przekłamanie w czasie odczytu czy inne zaburzenia transmisji). Szczególnie istotna w tym aspekcie jest elastyczna reprezentacja danych obrazowych, a to ze względu na występujące często znaczące rozmiary plików oryginalnych oraz dużą wagę zawartych w nich informacji.

Elastyczna postać skompresowanej reprezentacji danych oznacza łatwość dekodowania jedynie żądanej postaci informacji, tj. wybór rodzaju skalowania danych (progresji kodowania), możliwość zorientowania na konkretny obiekt (np. wybrany region zainteresowań), a także ewentualność reorientacji (trans-kodowania), czyli przekształcenia do nowej postaci skompresowanej reprezentacji o inaczej uporządkowanej lub wyselekcjonowanej informacji (np. w celu transmisji strumienia danych, drukowania, wyświetlenia w określonych warunkach).

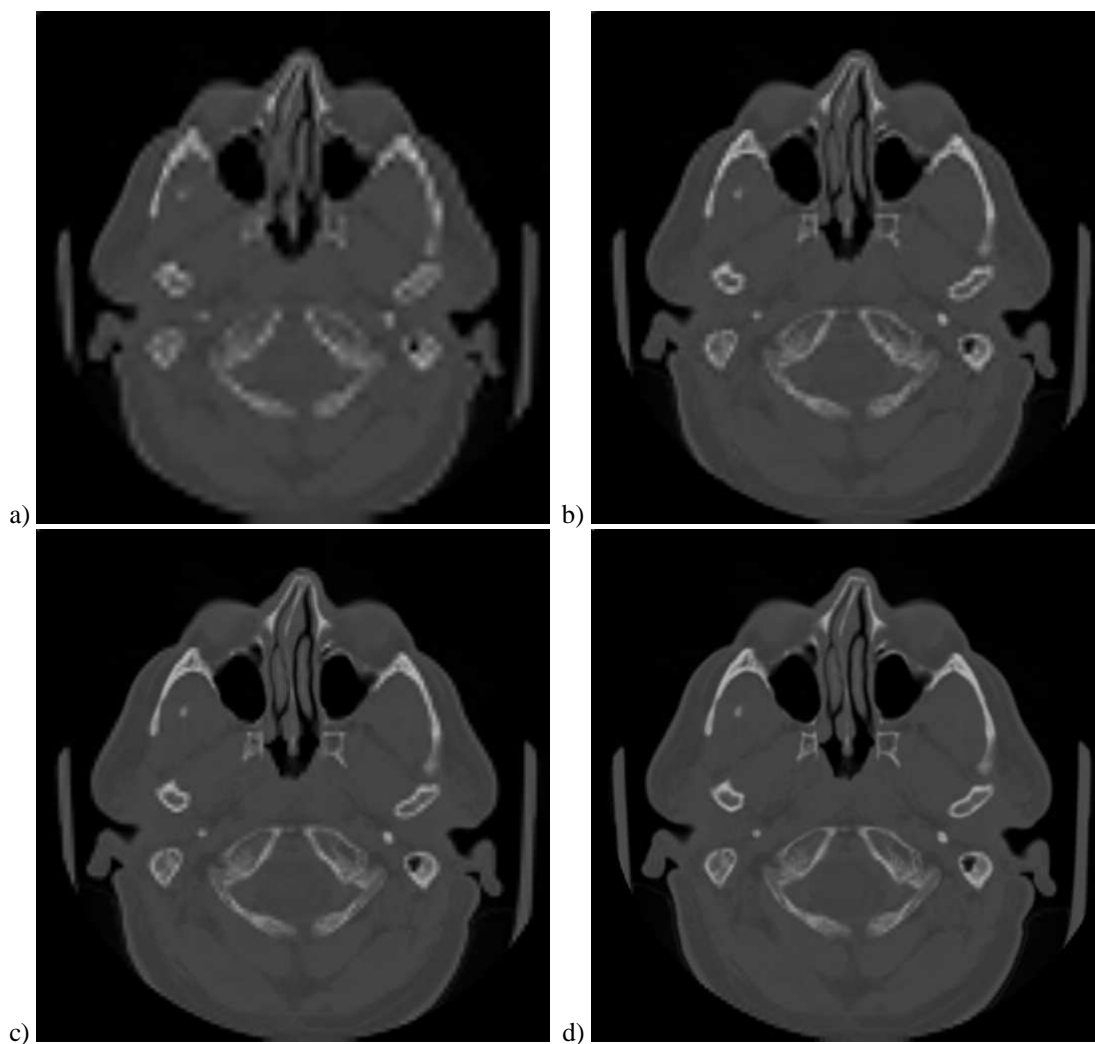
1.1.1. Progresja, osadzanie i skalowalność

Istotnym zagadnieniem przy transmisji, ale także archiwizacji zbiorów danych jest **progresywne**, czyli stopniowe przekazywanie informacji w strumieniu kodowym według ustalonego porządku: od postaci najbardziej ogólnej do najdrobniejszych szczegółów. Progresywny strumień danych może być zorientowany na jakość (optymalizacja w sensie R-D przekazywanej informacji), rozdzielczość (skalę, podpasmo, częstotliwość), komponent (np. koloru - RGB), położenie w przestrzeni, a także zadany obszar zainteresowania ROI (ang. *Region Of Interests*). Przydatne jest nieraz łączenie różnych rodzajów progresji w ustalonej hierarchii, w zależności od potrzeb konkretnej aplikacji. Przedstawione niżej przykłady kolejno transmitowanych i rekonstruowanych informacji dotyczą obrazu CT (tomografii komputerowej) z rys. 1.3.



Rys. 1.3. Przykładowy obraz CT kodowany/rekonstruowany za pomocą algorytmu falkowego: a) obraz oryginalny 512×512×8 bitów (obraz testowy z JPEG2000 obcięty do ośmiu najbardziej znaczących bitów do celów prezentacji); b) ten sam obraz z zaznaczonym ROI.

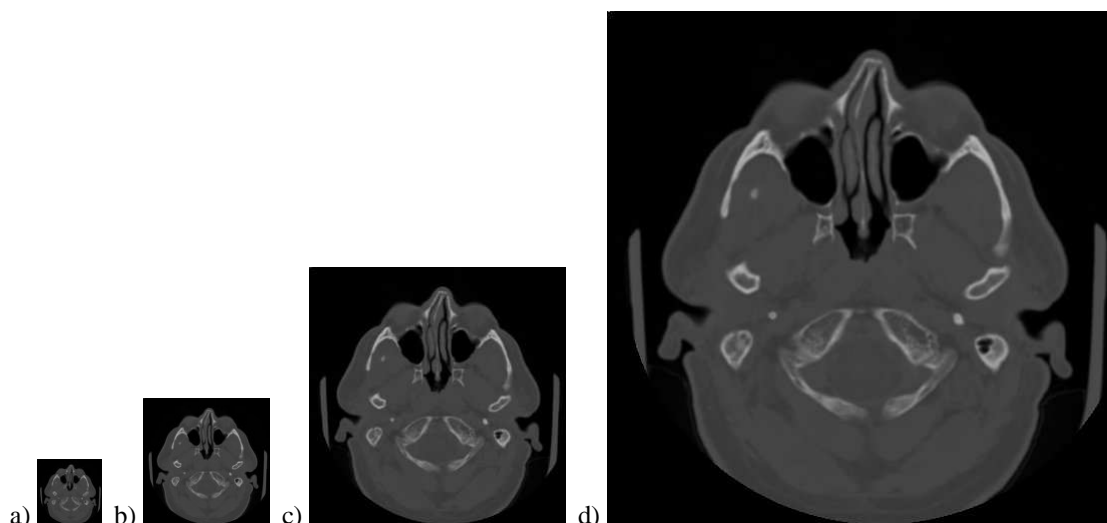
W zastosowaniach transmisyjnych progresja jakości pozwala bardzo szybko rozpoznać ogólny charakter przekazywanej informacji, gdyż w pierwszej kolejności przesyłany jest strumień dający duży przyrost jakości rekonstrukcji (według ustalonego kryterium) na bit reprezentacji danych. Efekt ten uzyskuje się poprzez kodowanie najpierw warstw najstarszych bitów wartości współczynników, potem młodszych. Transmisję uporządkowaną jakościowo przedstawiono na rys. 1.4, gdzie dekodowanie informacji z kolejnych partii strumienia kodowego powoduje stopniową poprawę jakości całego obrazu. W pierwszej rekonstrukcji przy średniej 0.1 bpp (bitów na piksel - ang. *bits per pixel*) widać wyraźne efekty rozmycia struktur czaszki, zarówno tkanki kostnej jak i miękkiej. Struktury te zostają sukcesywnie wyostnione na kolejnych obrazach, po transmisji i rekonstrukcji kolejnych szczegółów.



Rys. 1.4. Obrazy rekonstruowane za pomocą dekodera falkowego (w realizacji JJ2000) przy transmisji zorientowanej warstwowo (jakościowo) bez uwytatniania ROI. Z jednego skompresowanego pliku można uzyskać obrazy o różnej jakości: a) po zdekodowaniu informacji zawartej w pierwszych 3190 bajtach strumienia (0.1 bpp); b) po zdekodowaniu 6524 bajtów (0.2 bpp); c) po zrekonstruowaniu obrazu z 9760 bajtów (0.3 bpp); d) po odtworzeniu 16313 bajtów strumienia (0.5 bpp).

Ze schematu dekompozycji wielorozdzielczej Mallata wynika naturalna hierarchia danych: od małorozdzielczej wersji obrazu oryginalnego po wersję o rozdzielczości oryginału. Takie uporządkowanie informacji zapewnia progresję zorientowaną na rozdzielczość w wyjściowym strumieniu danych z kodera. Progresja rozdzielczości jest wygodna szczególnie w systemach bazodanowych do archiwizacji obrazów, kiedy to z archiwum pobierane są obrazy w różnej skali, dopasowanej do możliwości urządzeń peryferyjnych danego systemu (drukarki, monitory o określonej rozdzielczości itp.). Progresja ta użyteczna jest także przy przeglądaniu baz danych (szybkie przeglądanie wzrokowe mniejszych baz, indeksowanie przy automatycznym przeszukiwaniu baz

dużych, różnego typu selekcja i klasyfikacja danych obrazowych itp.). W przykładzie porządkowania informacji według rozdzielczości na rys. 1.5 pokazano kolejno rekonstruowane postacie obrazu CT.



Rys. 1.5. Obrazy rekonstruowane za pomocą algorytmu falkowego (JJ2000) przy dekodowaniu skalowanym według rozdzielczości. Z jednego skompresowanego pliku można uzyskać kolejno obrazy o różnej rozdzielczości: a) po odtworzeniu informacji zawartej tylko we współczynnikach LL najwyższego poziomu (największej skali, przy dekompozycji Mallata); b) po uzupełnieniu o dodatkową informację z podpasem tej samej skali; c) po uzupełnieniu o dodatkową informację kolejnej, bardziej dokładnej skali; d) po rekonstrukcji z wykorzystaniem wszystkich zakodowanych wartości współczynników falkowych.

Bardzo korzystną cechą metody kompresji jest więc zdolność do arbitralnego wyboru typu progresji, zarówno na etapie kompresji jak i dekompresji, przy czym oba te wybory powinny być w dużym stopniu niezależne. Przykładowo, kompresja zorientowana na progresję rozdzielczości wcale nie narzuca takiej samej progresji w procesie rekonstrukcji. Aby zrealizować taki schemat transmisji informacji, trzeba czasami pomiędzy koderem i dekodery umieścić transkoder realizujący konwersję strumienia zorientowanego na jeden rodzaj progresji w nowy strumień, inaczej zorientowany.

Istotną cechą procesu kodowania w kontekście różnego typu progresji jest skalowalność. Najczęściej jest ona rozumiana jako kodowanie pozwalające uzyskać (zapisać, przesłać) jednocześnie (w tym samym strumieniu kodowym) informację o oryginale w więcej niż jednej wersji rozdzielczości i/lub jakości. Skalowalność pozwala więc zrealizować różne rodzaje progresji, umożliwia także jedynie częściowe dekodowanie skompresowanego strumienia, co jest kluczowe przy realizacji nowego paradygmatu kompresji. W kodowaniu skalującym tworzona jest reprezentacja kodowa, która umożliwia rekonstruowanie obrazów z różną rozdzielczością oraz/lub jakością, poprzez dekodowanie odpowiednich partii wejściowego strumienia danych. Rozdzielczość i/lub jakość rekonstrukcji rośnie wraz z długością dekodowanych podzbiorów tego strumienia, przy czym ilość rekonstruowanej informacji (średnio na dekodowaną daną) jest coraz mniejsza. W przypadku strumienia skalowanego mogą istnieć różnego typu dekodery, tzn. o małej złożoności (stosunkowo wolne, bez koniecznej dużej ilości pamięci do przechowywania zrekonstruowanej informacji) odtwarzające jedynie obrazy podglądowe ze strumienia o średniej np. 0.1 bpp lub też o dużej złożoności rekonstruujące dobrej jakości obrazy ze strumienia np. 1 bpp. Niezależnie od nich mogą pracować dekodery odtwarzające jedynie wersje bezstratne obrazów, oczywiście reprezentacji wytworzonej przez kodery skalujące.

Aby uzyskać elastyczną kontrolę nad tworzonym strumieniem kodowym, niezbędna jest reprezentacja, która posiada ponadto cechę **osadzenia**. Oznacza to, że reprezentacja obrazu o większej średniej bitowej składa się z reprezentacji dla mniejszej średniej bitowej uzupełnionej o informację dodatkową. Dokładniej definicja kodowania z osadzaniem jest następująca: jeśli długość dwóch zbiorów utworzonych w danym koderze wynosi odpowiednio M i N bitów, gdzie $M > N$, to zbiór o rozmiarze N jest identyczny z pierwszymi N bitami zbioru o długości M .

Przy takiej organizacji strumienia danych możliwe jest przerwanie procesu kompresji w dowolnej chwili (nawet przy zachowaniu optymalności w sensie R-D), kontynuowanie kompresji ze zmienionymi parametrami, zmiana porządku odtwarzania itp. – a wszystko przy stałej kontroli długości strumienia.

1.1.2. Kodowanie wybranego obszaru zainteresowań

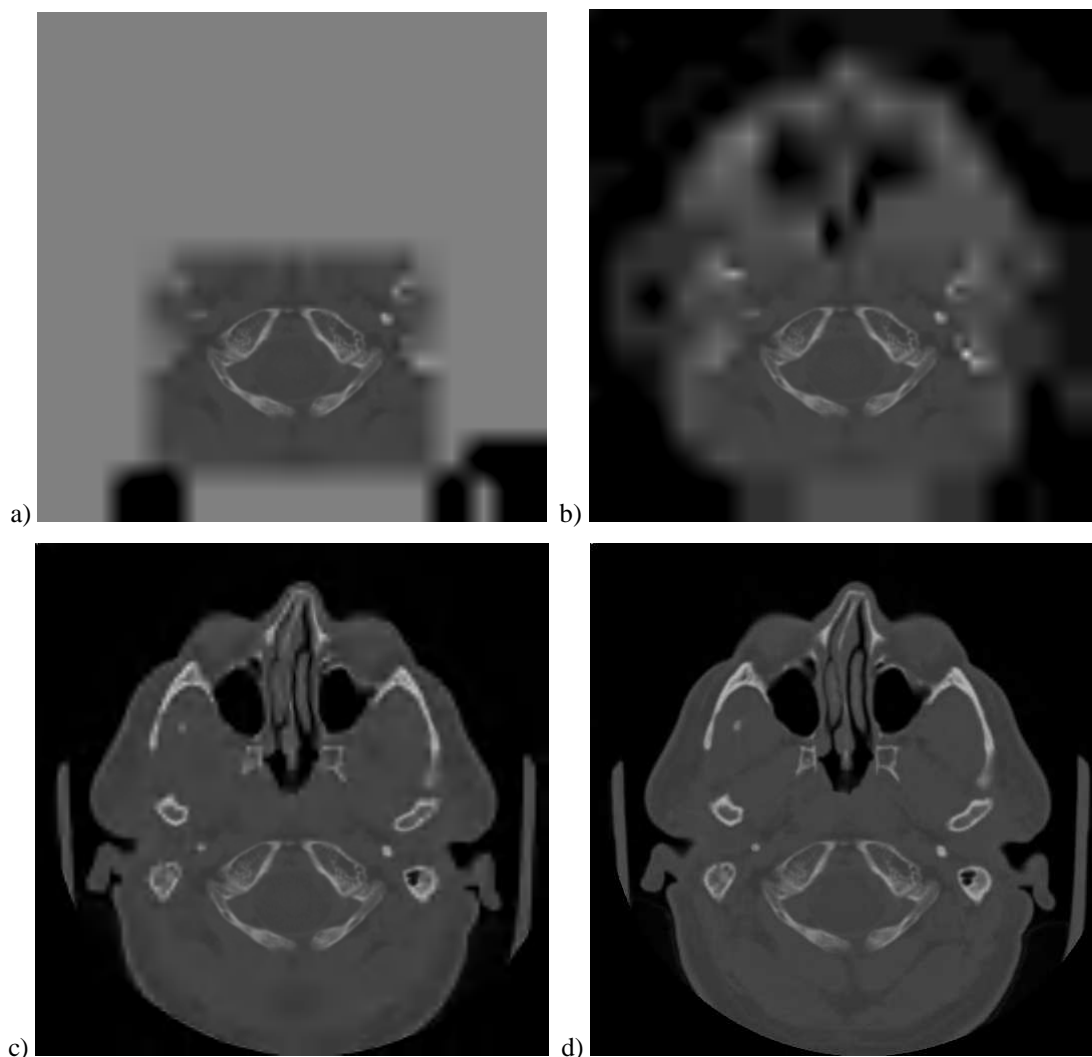
Wybór obszaru zainteresowania, który powinien być inaczej potraktowany w procesie kompresji, jest niemal niezbędną funkcją kodera w zastosowaniach medycznych. Ze względu na wysokie wymagania odnośnie jakości obrazów medycznych, wprowadzanie jakichkolwiek zmian w ważnych diagnostycznie fragmentach obrazu musi być niezwykle ostrożne. Jednocześnie rzadko zdarza się, aby użyteczna diagnostycznie informacja rozłożona była na powierzchni całego obrazu. Wstępna, obszarowa selekcja treści (informacji) zawartej w obrazie na poziomie semantycznym, dokonywana najczęściej po specjalistycznej analizie treści obrazu oryginalnego (powstającego w danym systemie obrazowania), stanowi wówczas istotny element kształtowania optymalnej postaci procesu elastycznego kodowania. Znaczącą kompresję uzyskuje się głównie poprzez silną redukcję nieistotnego diagnostycznie tła, zachowując diagnostyczną wiarygodność obrazu wskutek bezstratnej (lub niemal bezstratnej) kompresji fragmentów (regionów) istotnych.

Możliwe jest oddzielne kodowanie wybranych obszarów o istotnej diagnostycznie treści. Do kompresji obiektów o dowolnych kształtach, powstałych w wyniku wstępnej selekcji (klasyfikacji) fragmentów obrazu, stosuje się transformacje falkowe adaptujące się do kształtu (ang. *shape adaptive*) oraz określone metody kodowania kształtu (jak w MPEG-4 VTC).

W innym przypadku wyróżnianie ROI w algorytmie kompresji odbywa się na etapie ustalania porządku kodowania współczynników falkowych, jak w JPEG2000. Przykład kompresji falkowej obrazu z selekcją ROI (według rys. 1.3) w realizacji tego standardu przedstawiono na rys. 1.6. Porównanie przykładowych obrazów z rys. 1.4a) i rys. 1.6a) ukazuje, że nadanie ROI wyższego priorytetu w progresywnej postaci strumienia danych pozwala uzyskać znacznie większą wiarygodność diagnostyczną obszaru zainteresowań przy tym samym stopniu kompresji (czasie transmisji). Do korzystnych operacji na ROI należy zaliczyć także możliwość dowolnego kształtowania ROI z dokładnością do piksela (dostępna np. w części pierwszej standardu JPEG2000) oraz możliwość dekodowania jedynie ROI z ogólnej reprezentacji obrazu. Ta druga funkcja wygodna jest przykładowo w bazach danych, kiedy to mając małorozdzielczy podgląd obrazu lub też jego opis słowny można od razu zdecydować o wyborze interesującego nas fragmentu, który następnie jest dekodowany z największą dostępną jakością.

Transmisję zorientowaną na ROI można uzyskać poprzez skalowanie, czyli przesunięcie bitów wartości współczynników ROI ponad najbardziej znaczący bit pozostałych współczynników. Wtedy, zdejmując kolejne mapy bitowe jako warstwy, kodowane są najpierw wszystkie bity współczynników ROI, a dopiero potem pozostałe. Można także łagodniej wyodrębnić obszar zainteresowań wysuwając jedynie część bitów wartości jego współczynników ponad pozostałe, ustawiając przy tym współczynniki ROI na początku każdego podpasma. Trzeba jednak wówczas zapisać dodatkową informację o kształcie ROI (część II JPEG2000).

Ważnym elementem jest realizacja interaktywnej transmisji, gdy selekcja ROI możliwa jest dopiero po wstępnym przesłaniu informacji według progresji jakościowej. Odbiorca obrazu sygnalizuje przełączenie na wybrany ROI w momencie, kiedy uzyskana jakość rekonstrukcji pozwala określić ten obszar. Wskutek odpowiedniego uporządkowania informacji, przesłanie już niewielkiej jej ilości (zapisanej na kilku, kilkunastu kilobajtach dla obrazu z rys. 1.3) pozwoli zorientować się w charakterze obrazu oraz wybrać ROI istotny diagnostycznie. Dosłanie kolejnej partii danych dotyczących jedynie wybranego obszaru pozwoli zrekonstruować ROI z oryginalną dokładnością, czyli bezstratnie. Podsumowując, przy takiej interaktywnej transmisji wystarczy często przesłać wielokrotnie mniej danych, aby umożliwić proces diagnostyczny na podstawie pełnej, dostępnej informacji. Aplikacje realizujące koncepcję takiego systemu telemedycznego w architekturze klient-serwer stanowią przedmiot zainteresowań prac części 9 JPEG2000.



Rys. 1.6. Obrazy rekonstruowane za pomocą dekodera falkowego (JJ2000) z uwytłaczonym ROI. Obraz z rys. 3.3 transmitowany jest przy założeniu przesyłania w pierwszej kolejności informacji z wybranego ROI. Kolejne wartości średnich bitowych strumienia, z których rekonstruowano obrazy a), b), c) i d) to odpowiednio 0.1 bpp, 0.2 bpp, 0.3 bpp oraz 0.5 bpp. Efekt nieznacznego rozszerzenia ROI wynika z zastosowanej struktury bloków kodowych 16×16 według aplikacji zgodnej ze standardem JPEG2000, cz. I).

1.1.3. Odporność na zakłócenia

W przypadku teletransmisji, zwłaszcza wobec silnie rozwijającej się ostatnio transmisji bezprzewodowej, dużej wagi nabiera odporność przesyłanego strumienia danych na zakłócenia, szczególnie w aplikacjach medycznych. Najdrobniejsze bowiem zniekształcenia postaci bitowej reprezentacji kodowej obrazu wskutek błędów transmisji mogą powodować zupełną nieczytelność dużych partii rekonstruowanego obrazu (tzn. oczywistą nieprzydatność diagnostyczną tego obrazu). Podczas telekonsultacji czynnik czasu jest nieraz bardzo istotny, a możliwość odtworzenia obrazu przy błędach transmisji, nawet nieco zniekształconego w stosunku do oryginału, może znacznie przyspieszyć pracę w porównaniu z oczekiwaniem na kolejną – bezbłędną transmisję całego obrazu.

Różne techniki zwiększenia odporności na błędy powodują nieznaczne wydłużenie strumienia danych kodowanych. Jednak uzyskana poprawa jakości obrazów rekonstruowanych przy dużym poziomie zakłóceń jest zdecydowana i często niewspółmierna do tych nakładów. Powinien więc istnieć solidny system detekcji i ostrzegania przed każdym zaistniałym przekłamaniem informacji podczas transmisji, jak również możliwie niezawodny i sprawny mechanizm korekcji błędów. Ważne są tutaj sposoby zabezpieczenia zastosowane w wykorzystywanej technologii transmisji (nie będące przedmiotem tych rozważań). Niebanalną rolę odgrywa również taki sposób konstruowania strumienia danych, który zapewni małą propagację błędów.

Odporność na zakłócenia buduje się na różnym poziomie reprezentacji kodowej. Jednym z rozwiązań jest kodowanie danych niezależnie, w odseparowanych segmentach, oddzielonych informacją dodatkową, która pozwala na przynajmniej częściową rekonstrukcję informacji zawartej w zgubionych segmentach czy pakietach. Ostatni etap falkowej kompresji, czyli binarne kodowanie metodami entropijnymi, winien być realizowany niezależnie, w niewielkich blokach (adaptacyjny model statystyczny jest zerowany przy przejściu od jednego bloku kodowanego do drugiego), co znacznie ogranicza propagację błędu na całe pasmo przy przekłamaniach bitowych. Podobnie, przy zmianie charakteru kodowanych danych, np. na znaki wartości współczynników falkowych, dookreślające bity kolejnych wartości modułów czy symbole mapy znaczeń, przełączany jest nowy model sterujący koderem arytmetycznym. Dodatkowo stosuje się tryb tzw. leniwego kodowania, czyli emisję bitów niosących informacje bez żadnego kodowania. Rozwiązanie to zwiększa odporność na błędy poprzez zminimalizowanie udziału kodowania arytmetycznego, które, jako metoda z grupy kodów o zmiennej długości, jest bardzo wrażliwe na wszelkie przekłamania bitowe.

Ważna jest także organizacja danych określonego fragmentu obrazu, warstwy, składnika, skali i korelacji przestrzennej w pakiety, które mają niewielkie rozmiary, a ponadto wyposażane są w markery resynchronizacji z kolejnym numerem w sekwencji. Pozwala to na ustalenie podziału przestrzeni i szybką, powtórna synchronizację procesu dekodowania w przypadku wystąpienia zakłóceń.

W testach porównawczych odporności na błędy transmisji strumieni dwóch standardów JPEG2000 i JPEG wykazano, że przy wyższych średnich bitowych i transformacji odwracalnej jakość rekonstrukcji strumienia JPEG2000 jest lepsza nawet o 6–7 dB wartości *PSNR* (szczytowy stosunek sygnału do szumu). Symulowano symetryczny, binarny kanał transmisyjny z losowymi błędami i oceniano średnią jakość rekonstruowanych obrazów przy regulowanym poziomie błędów.

1.1.4. Kompresja stratna-bezstratna

Zatarcie pojęć kompresji stratnej i bezstratnej w jednorodnym schemacie kompresji pozwala bardzo swobodnie kształtować postać strumienia wyjściowego. Szczególnie widoczne jest to w kompresji obrazów medycznych, kiedy konieczna postać dokładnej rekonstrukcji oryginału, znajdująca się w odpowiedniej 'bazie odniesienia', może służyć jako 'generator' różnych postaci obrazu w zależności od potrzeb (przeglądania, porównania obliczeniowego, zdalnej konsultacji, drukowania, wyszukiwania itp.).

Koder falkowy, zachowując dużą skuteczność kompresji w szerokim zakresie średnich bitowych, także reprezentacji odwracalnej, pozwala zwiększyć swe walory użytkowe w szeregu aplikacjach. Dzięki temu, rola koderów falkowych wydaje się być fundamentalna w dalszym rozwoju technik kompresji obrazów medycznych, o czym świadczy chociażby fakt włączenia falkowej kompresji według JPEG2000 w rozwój standardu formatu, protokołu transmisji i organizacji medycznych baz danych DICOM.

1.1.5. Realizacja elastycznych koderów falkowych

W tabeli 1.1 zebrano różne funkcje kodera falkowego decydujące o jego użyteczności w wielu aplikacjach, w tym medycznych. Jako przykłady rozwiązań dobrze realizujących te funkcje podano wymienione wcześniej standardy, a także efektywne techniki kompresji. Do najbardziej pożądanых cech algorytmu kompresji należy zaliczyć gotowość do pracy w następującym schemacie: podgląd małej (bardzo niskorozdzielczej) wersji obrazu (miniaturki - ang. *thumbnail*) przy przeglądaniu bazy obrazowej, możliwość szybkiej rekonstrukcji pełnej wersji obrazu (przy różnych kryteriach dotyczących zarówno kolejności odtwarzania, jak i jakości wersji finalnej) albo uzyskanie wybranego fragmentu ROI o najlepszej, możliwej jakości (przy alternatywnych sposobach kolejności rekonstrukcji ROI), z silną redukcją tła. Kluczem, który pozwala uzyskać taką elastyczność procesu odtwarzania obrazu, jest odpowiednie uformowanie skompresowanej reprezentacji obrazu, uzupełnione wielowariantową dekompresją. Podkreślić należy także niezawodność całego systemu, czyli odporność na różnego typu zakłócenia.

Wszystkie te rozwiązania funkcjonalne oraz przyjęte koncepcje ich realizacji, oparte na falkowej reprezentacji danych, winny prowadzić do procedur kompresji i dekompresji sprzętowo i programowo wykonywalne w możliwie krótkim czasie. Chodzi bowiem o to, by znalazły zastosowanie w

najbardziej kluczowych obszarach wykorzystania współczesnych koderów, tj. w systemach transmisji i przetwarzania olbrzymich ilości danych obrazowych w czasie rzeczywistym, zgodnie z tempem ich rejestracji.

Tabela 1.1. Elementy składające się na funkcjonalność kodera falkowego i decydujące o jego elastyczności w perspektywie wielorakich zastosowań. Przy każdej funkcji wymieniono przykładowe techniki czy standardy kompresji falkowej, w których jest ona skutecznie realizowana.

Funkcja kodera falkowego	Techniki i standardy kompresji
Duża efektywność kompresji bezstratnej	JPEG2000, SPIHT, MBWT
Duża efektywność kompresji stratnej	FD, SC&CE, TCSFQ, MBWT, JPEG2000, C/B, PACC i inne
Tworzenie progresywnego strumienia danych (wybór typu progresji)	JPEG2000, CREW
Różnorodne kodowanie i dekodowanie ROI	JPEG2000, CREW, MPEG-4 VTC (oddzielny mod)
Kodowanie obiektów o dowolnym kształcie	MPEG-4 VTC
Osadzanie strumienia kodowego i kontrola jego długości	JPEG2000, CREW, SPIHT, TCSFQ, FD, SC&CE i inne
Zwiększona odporność na błędy transmisji	JPEG2000, MPEG-4 VTC

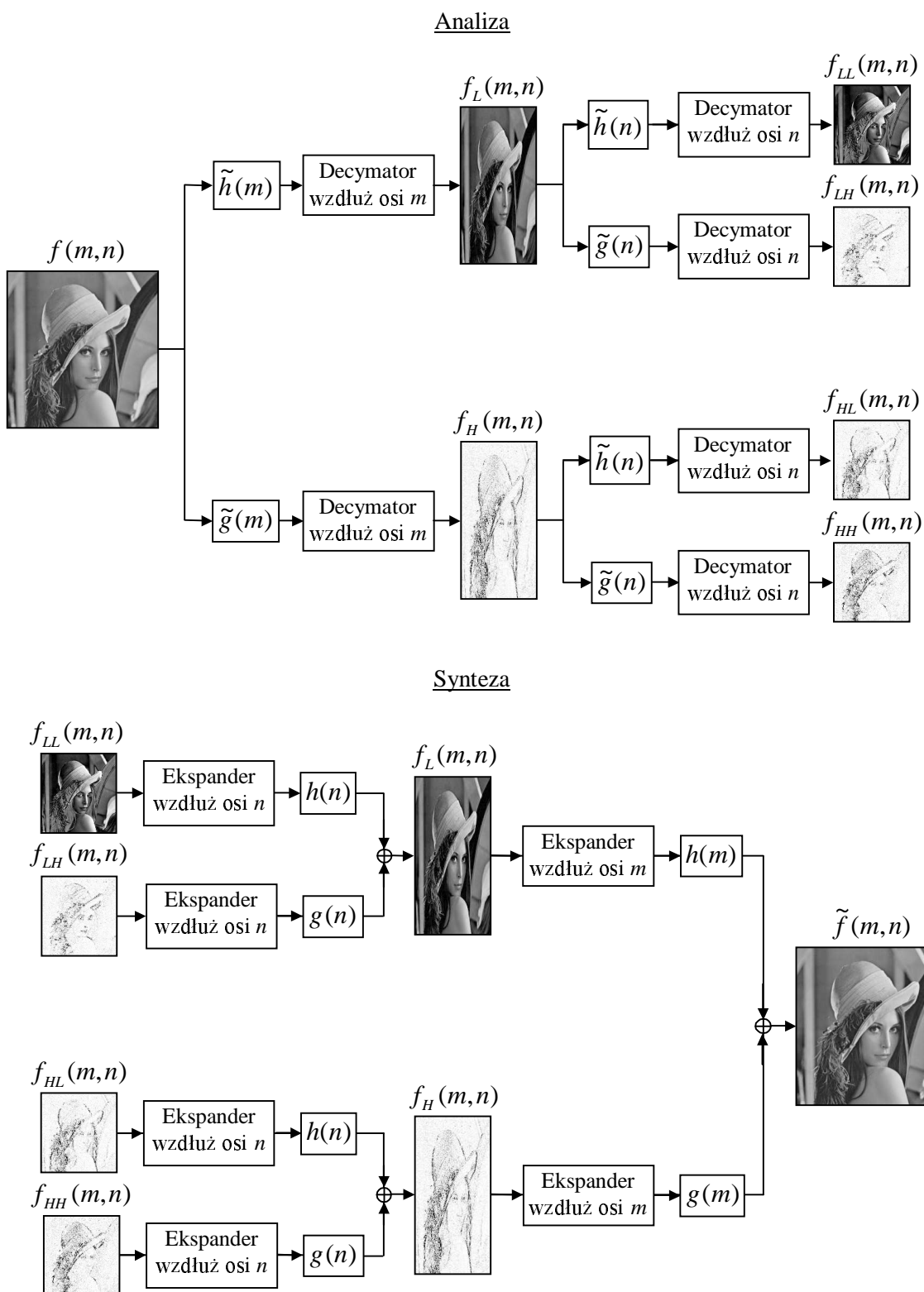
1.2. FALKOWA DEKOMPOZYCJA OBRAZÓW

Podstawowy schemat falkowej dekompozycji: analizy i syntezy obrazu przedstawia rys. 1.7. Jest to jeden z trzech zasadniczych elementów algorytmu kompresji falkowej, do których zaliczyć należy także: kwantyzację zorientowaną w dziedzinie przestrzeń-skala oraz binarne kodowanie (entropijne) odpowiednio formowanego strumienia danych, które nadaje ostateczny kształt nowej reprezentacji kompresowanego zbioru danych.

Możliwie pełna dekokorelacja danych w ujęciu statystycznym, koncentracja energii sygnału w możliwie małej liczbie współczynników lub inaczej - uwypuklenie zależności danych w hierarchicznej strukturze przestrzenno-częstotliwościowej, może być osiągnięte w koderze falkowym poprzez odpowiedni dobór schematu pasmowej dekompozycji (tj. diadyczny Mallata, równomierny, adaptacyjny z pakietami falek itp.) oraz banku filtrów, realizujących ten schemat poprzez dolno- i górnoprzepustową filtrację w obu kierunkach przestrzeni.

Istotnym zadaniem w konstrukcji kodera falkowego jest projektowanie efektywnych filtrów do analizy i syntezy obrazu. Wydajność kompresji całego algorytmu kompresji można znacząco poprawić poprzez: budowanie bazy falkowej z funkcji o odpowiednim poziomie gładkości czy regularności przy założonym kryterium ortogonalności lub biortogonalności (nadmiarowe bazy są raczej nieużywane w zastosowaniach kompresji), zapewnienie symetryczności czy antysymetryczności tych funkcji, ustalenie odpowiedniej długości skojarzonych z bazą falkową filtrów oraz wykorzystanie innych stopni swobody kształtowania optymalnej postaci przekształcenia falkowego (np. możliwości adaptacyjnej zmiany bazy transformacji, wiedzy *a priori* o klasie funkcji (sygnałów) kompresowanych itp.).

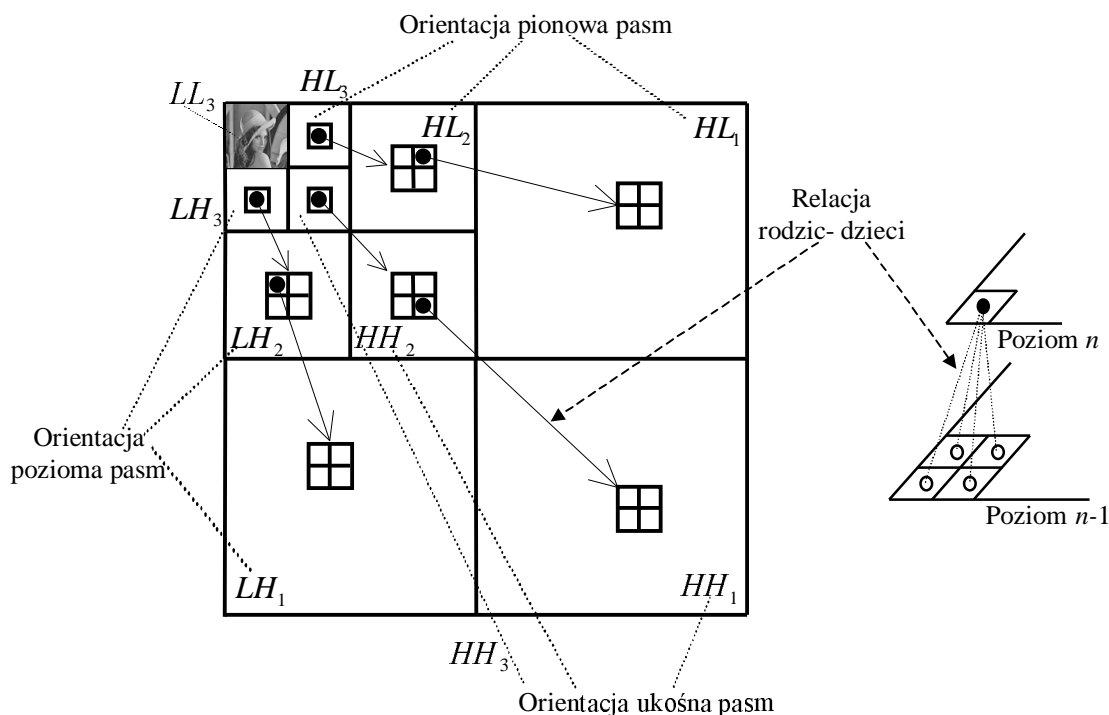
Ze względu na niestacjonarny charakter modelu źródła informacji - dużą różnorodność cech obrazu istotnych z punktu widzenia konkretnych aplikacji, a także różny poziom jakości kompresowanych obrazów (stosunek sygnału do szumu, przestrzenna rozdzielczość, częstotliwościowe widmo sygnału itp.), bardzo trudnym zadaniem okazuje się opracowanie metody konstrukcji filtrów optymalnych z punktu widzenia efektywności kompresji. Na obecnym poziomie wiedzy wiele zagadnień, odnoszących się do wyboru najbardziej skutecznych w kompresji filtrów, jest nadal nie rozwiązanych. Podobne problemy występują przy konstruowaniu praktycznych metod wyznaczania schematu optymalnej dekompozycji pasmowej, a wspomniane w rozdz. 3 pakiety falek nie zawsze pozwalają na uzyskanie najkorzystniejszych rozwiązań (duża złożoność obliczeniowa, konieczność dopisania informacji dodatkowej).



Rys.1.7. Schemat falkowej analizy oraz syntezy obrazu. \tilde{h}, \tilde{g} - dolno- i górnoprzepustowy filtr analizy, h, g - dolno- i górnoprzepustowy filtr analizy.

Hierarchiczne drzewo dekompozycji Mallata zostało przedstawione na rys. 1.8. Cztery podpasma składowych o najniższych częstotliwościach stanowią najwyższy poziom tego drzewa. Dane należące do tego poziomu nie mają rodzica i są rodzicami pierwszej generacji dla wszystkich skojarzonych przestrzennie współczynników. Każdy współczynnik, na poziomie różnym od podstawy drzewa,

rozrasta się w grupę czterech współczynników kolejnego poziomu dokładniejszej skali, będąc z nimi w bezpośredniej relacji rodzic-dzieci.



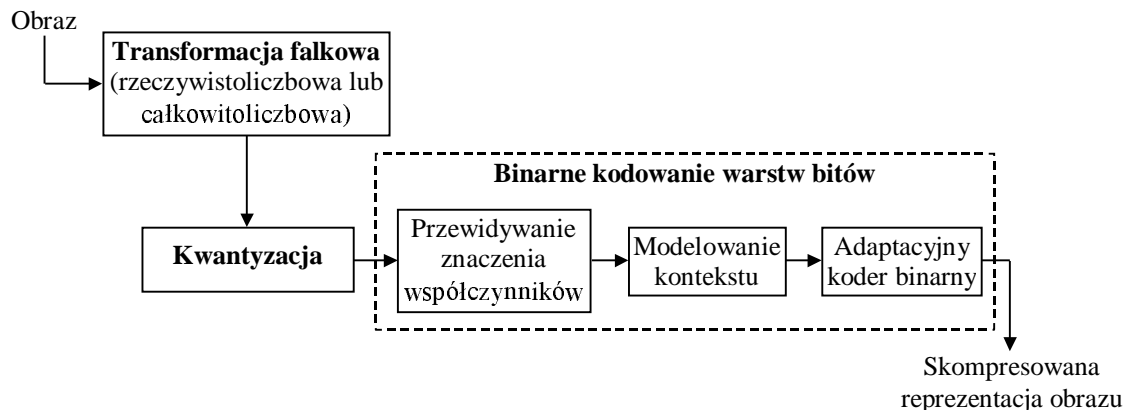
Rys.1.8. Podstawowy schemat falkowej dekompozycji obrazu.

Pierwsze w hierarchii podpasmo najniższych częstotliwości LL_3 zawiera zwykle najwięcej informacji o obrazie (średnio na pojedynczy współczynnik). Potem występują kolejne podpasma największej skali: HL_3, LH_3 i na końcu HH_3 , gdzie L oznacza podpasmo po filtracji dolnoprzepustowej, a H – górnoprzepustowej (najpierw po wierszach, potem po kolumnach - zgodnie z rys. 1.8). Zależności pomiędzy współczynnikami tych podpasm określa horyzontalna relacja drzewa dekompozycji, wyrażająca podobieństwa treści częstotliwościowych kolejnych podpasm tej samej skali w danym miejscu przestrzeni. Następnie w hierarchii są trzy podpasma drugiego poziomu drzewa: HL_2, LH_2, HH_2 , których współczynniki pozostają w relacji rodzic-dzieci w stosunku do współczynników z podpasm zarówno bardziej zgrubnej skali, jak i dokładniejszej. Na najniższym poziomie drzewa znajdują się podpasma najdokładniejszej skali: HL_1, LH_1, HH_1 , które nie mają węzłów potomnych.

1.3. SCHEMAT OGÓLNY KODERA FALKOWEGO

Schemat blokowy kodera falkowego, który znajduje zastosowanie w wielu aplikacjach multimedialnych czy też medycznych systemach informacyjnych, pokazano na rys. 1.9. Większość efektywnych koderów falkowych wykorzystuje rzeczywistoliczbowe transformacje falkowe, czyli przekształcenia całkowitych (z dziedziny liczb całkowitych) wartości pikseli w zbiór współczynników o wartościach rzeczywistych (z dziedziny liczb rzeczywistych). Uniemożliwia to realizację w prosty sposób koderów bezstratnych ze względu na konieczność przybliżenia tych wartości liczbami całkowitymi (kwantyzacja), które dają się efektywnie zakodować. Warunkiem koniecznym uzyskania kompresji odwracalnej jest więc wykorzystanie całkowitoliczbowych transformacji falkowych, które dają współczynniki całkowite bez konieczności kwantyzacji. Są więc odwracalne w arytmetyce o ograniczonej precyzji. Transformacje te pozwalają skonstruować strumień kodowy w koncepcji kompresji stratnej-bezstratnej, a blok kwantyzacji definiuje jedynie postać pośrednią (wpływa na kolejność ustawienia informacji w strumieniu danych). Skuteczność transformacji całkowitoliczbowych na etapie kompresji stratnej jest nieco mniejsza, jednak odpowiednia aranżacja procesu kodowania pozwala w wielu wypadkach otrzymać efektywność zbliżoną do transformacji

rzeczywistoliczbowych. Ponadto, wyniki bezstratnej kompresji obrazów (osiągane średnie bitowe) są nierzadko podobne do rezultatów najskuteczniejszych koderów odwracalnych, takich jak CALIC. Kodowanie warstw bitowych, zawierających kolejne partie bitów poszczególnych współczynników, opiera się na przewidywaniu wystąpienia współczynnika znaczącego (względem wartości aktualnego progu selekcji), modelowaniu kontekstu, czasem jego kwantyzacji w celu budowy optymalnego modelu statystycznego, sterującego adaptacyjnym koderem entropijnym, najczęściej a rytmicznym.



Rys. 1.9. Schemat blokowy efektywnego koderza falkowego w wersji stratnej (z transformacją rzeczywistoliczbową) oraz stratnej-bezstratnej (z transformacją całkowitoliczbową). Wykorzystanie elementu kwantyzacji pozwala często na prostsze rozwiązania koderza binarnego.

Na podstawie analiz teoretycznych i eksperymentalnych ustalono, że najbardziej użyteczne w rozwiązaniach praktycznych schematy kwantyzacji oparte są na klasycznym, skalarnym kwantyzatorze równomiernym ze zmodyfikowaną szerokością przedziału zerowego. Istotne z punktu widzenia globalnej efektywności kompresji okazuje się także wprowadzenie do tego schematu czynnika adaptacyjnego, dopasowanego do lokalnych własności danych, a przez to w pełni wykorzystującego przestrzenno-częstotliwościową charakterystykę domeny falkowej. Innym, sprawdzonym, jednak nie zawsze efektywnym rozwiązaniem jest kwantyzacja z kodowaniem kraty TCQ (ang. *Trellis Coded Quantization*). Brakuje w tej koncepcji jednoznacznie skutecznych metod optymalizacji podstawowego schematu TCQ, mającego znaczące ograniczenia przy kompresji w zakresie małych średnich bitowych.

Wielorozdzielcza dekompozycja danych obrazowych w koncepcji Mallata daje naturalną reprezentację danych w hierarchii skali i częstotliwości kolejnych pasm, zgodnej w pierwszym przybliżeniu z hierarchią znaczeń poszczególnych współczynników (tj. modulem ich wartości). Największa energia współczynników (skalujących, po wielokrotnej filtracji dolnoprzepustowej w obu kierunkach przestrzeni obrazu) występuje bowiem w podpaśmie najniższych częstotliwości. Przesuwając się w kierunku pasm o wyższej częstotliwości, średni poziom energii współczynników maleje. Przy kompresji zachowywana jest często taka właśnie kolejność, ustalona przez częstotliwościowe uporządkowanie przesyłanej informacji kolejnych skal (od dużej do małej), uzupełniona modelowaniem korelacji i zależności danych w celu skutecznej ich kwantyzacji i kodowania. Chcąc uzyskać monotonicznie malejącą energię przesyłanych współczynników, trzeba zastosować algorytmy przeglądania i porządkowania współczynników z różnych pasm. Można wreszcie optymalizować strumień w sensie R-D (stopnia zniekształceń źródeł informacji), biorąc pod uwagę nie tylko ilość informacji zawartej w wartości danego współczynnika czy grupie współczynników, ale także możliwość skutecznego jej zakodowania w konkretnym miejscu tworzonego strumienia danych.

Szczególnej wagi nabiera kodowanie współczynników falkowych z wykorzystaniem wielowymiarowych modeli kontekstu oraz binarnych koderów arytmetycznych. Dokładne opisanie statystyki danych, za pomocą odpowiednio dobranych kontekstów, pozwala sterować koderem arytmetycznym z wysoką skutecznością. Prosty schemat sukcesywnej aproksymacji wartości współczynników okazuje się najwygodniejszym i często wystarczająco efektywnym schematem kwantyzacji. Jednak wymaga on stosowania złożonych metod modelowania kontekstu, często obejmującego swym zasięgiem sąsiednie pasma i skale, co ogranicza użytkowe walory

skompresowanego strumienia danych. Trudniejszy jest bowiem dostęp do wybranych fragmentów obrazu, wzrasta także propagacja hipotetycznych błędów transmisji strumienia.

Optymalizację schematu kodowania binarnego przeprowadza się często razem z kwantyzacją, uwzględniając przy tym realizowany sposób falkowej dekompozycji obrazu. Jeszcze trudniej jest wskazać tu rozwiązania optymalne. Część algorytmów wykorzystuje strukturę drzew zer, dobrze modelującą relacje rodzic-dzieci w hierarchii falkowej dekompozycji, inne z kolei rozbudowują modele statystyczne koderów entropijnych obejmując nimi lokalne zależności danych domeny falkowej. Coraz większą rolę odgrywa także sposób formowania strumienia, kolejność ustawienia informacji, zabezpieczenie przed błędami transmisji itp. wymuszając rozwiązania nietypowe.

2. STANDARD JPEG 2000

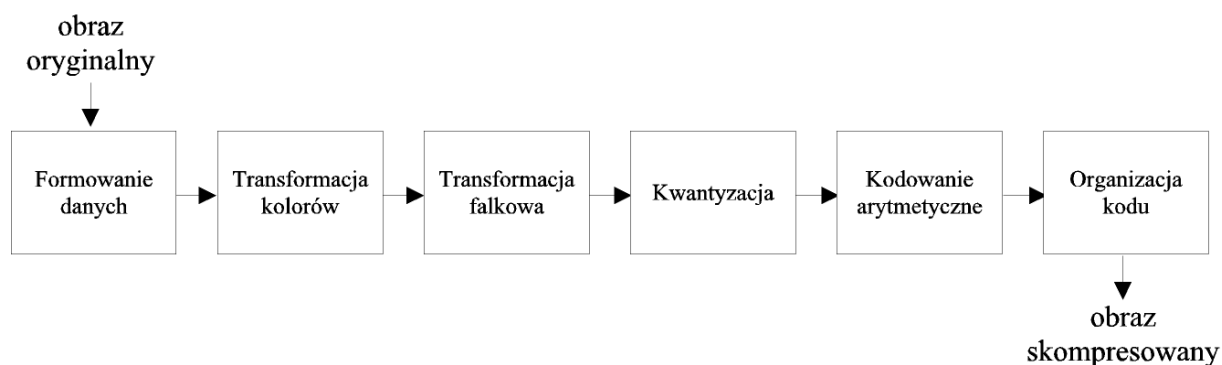
W marcu 1997 rozpoczęto prace nad nowym standardem kompresji obrazów JPEG2000. Obecnie prowadzone są prace (niektóre już zakończono) nad następującymi częściami JPEG2000:

- część 1: podstawowy system kompresji obrazów pojedynczych;
- część 2: rozszerzenia (dodatkowe opcje systemu, m.in. możliwość definiowania własnych transformacji falkowych i kolorów, inny sposób kodowania ROI, nowe algorytmy kwantyzacji itp.);
- część 3: algorytm kompresji sekwencji obrazów (Motion JPEG 2000);
- część 4: testy zgodności (conformance);
- część 5: oprogramowanie referencyjne, tj. biblioteki w Javie (JJ2000) oraz C (JasPer);
- część 6: format pliku dla danych złożonych (obrazowanie dokumentów, artykułów prasowych itp.);
- część 7: (prace nad tą częścią zostały zaniechane);
- część 8: JPSEC, bezpieczeństwo danych;
- część 9: JPIP, interaktywne protokoły transmisji i API;
- część 10: JP3D, obrazowanie 3D;
- część 11: JPWL, aplikacje bezprzewodowe.

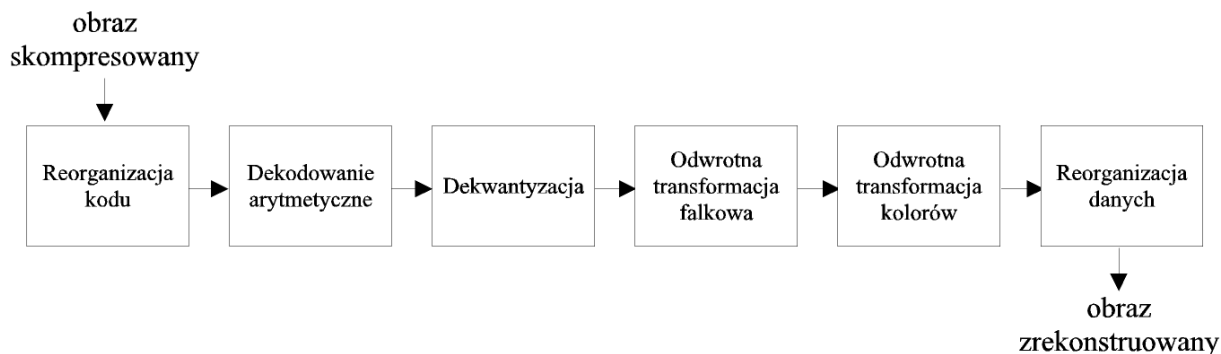
Standard JPEG2000 umożliwia kompresję obrazów o niemal dowolnej reprezentacji oryginalnej, tzn. możliwa jest duża liczba komponentów oraz bitów na próbkę w niemal nieograniczonej przestrzeni obrazu, a także różna orientacja dziedziny obrazu względem układu odniesienia. Liczba komponentów jest ograniczona do 2^{14} (16384); natomiast ich maksymalna szerokość i wysokość to $2^{32}-1$ próbki (przekracza 4 miliardy próbek). Liczba bitów na próbkę jest ograniczona do 38 bitów (ponad 4 bajty).

2.1. SCHEMAT KOMPRESJI

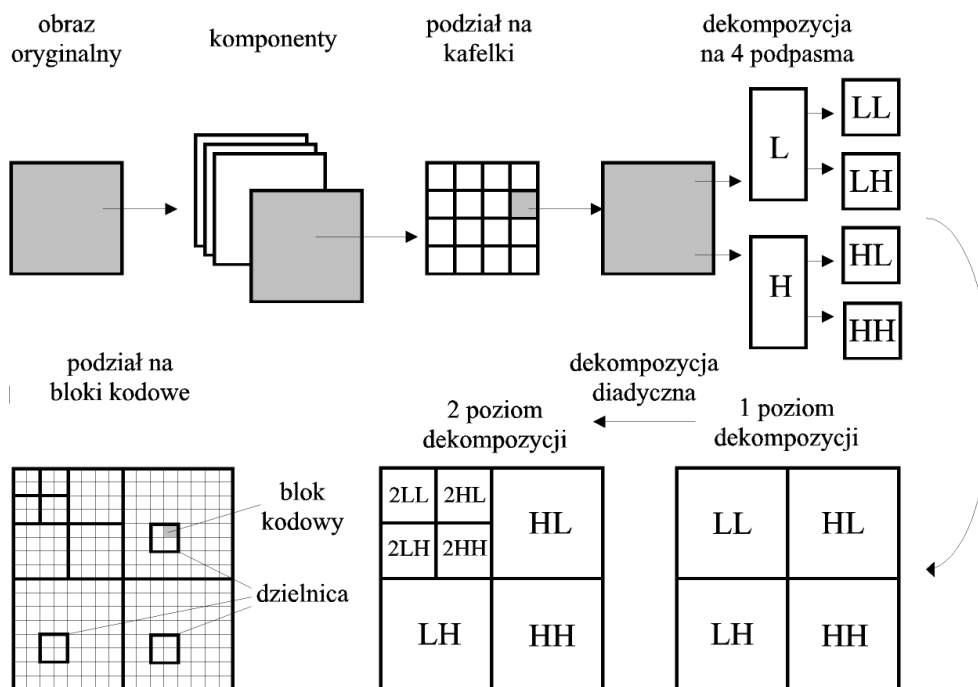
Na rysunkach 2.1 i 2.2. przedstawiono schemat blokowy odpowiednio algorytmu kompresji oraz dekompresji; natomiast na rysunku 2.3 postać danych w kolejnych krokach procesu kompresji.



Rys. 2.1. Schemat kompresji w JPEG2000.



Rys. 2.2. Schemat dekompresji w JPEG2000.



Rys. 2.3. Przeobrażenia na kolejnych etapach kompresji według JPEG2000.

Pierwszym etapem kompresji jest formatowanie danych polegające na umiejscowieniu poszczególnych komponentów obrazu na *siatce odniesienia* (ang. *reference grid*) oraz podziale ich na części (ang. *tile*). Zastosowanie siatki umożliwia dokonywanie prostych operacji na obrazie skompresowanym (m.in. obrót o wielokrotność 90 stopni) oraz kompresję obrazów o komponentach o różnej wielkości. Podział obrazu na kafelki ma na celu zmniejszenie zapotrzebowania na pamięć, umożliwia przyspieszenie kodowania (dzięki możliwości równoczesnego kodowania wielu części obrazu) oraz daje dostęp do fragmentów obrazu bez konieczności dekompresji całości (należy zaznaczyć, że taka lokalna dekompresja jest możliwa także w inny sposób). Po podziale obrazu na części każda z nich jest dalej przetwarzana niezależnie.

Po podziale każda część jest poddawana transformacji kolorów. W cz. I standardu zastosowano transformację YCbCr (YCC) w wersji odwracalnej (YCC_R) oraz nieodwracalnej (YCC_I). Transformacja może być zastosowana tylko w przypadku, kiedy obraz zawiera co najmniej 3 komponenty, przy czym 3 pierwsze muszą mieć taką samą wielkość. Po transformacji kolorów wszystkie komponenty są dalej przetwarzane niezależnie. Cz. II standardu umożliwia dodanie innych (własnych) transformacji przestrzeni kolorów. Poniżej zdefiniowano kilka takich transformacji:

- nieodwracalne przekształcenie RGB na YCbCr:

$$\begin{aligned}
 Y &= 0.2989 R + 0.5866 G + 0.1145 B \\
 Cb &= -0.1687 R - 0.3312 G + 0.5000 B \\
 Cr &= 0.5000 R - 0.4183 G - 0.0816 B
 \end{aligned}$$

z przekształceniem odwrotnym YCbCr na RGB:

$$\begin{aligned} R &= 1.000 Y + 1.4022 Cr \\ G &= 1.000 Y - 0.3456 Cb - 0.7145 Cr \\ B &= 1.000 Y + 1.7710 Cb \end{aligned}$$

- odwracalne RGB→YCbCr:

$$Y = \left[\frac{1}{4} R + \frac{1}{2} G + \frac{1}{4} B \right]; Cb = B - G; Cr = R - G$$

a YCbCr→RGB:

$$G = Y - \left[\frac{1}{4} Cb + \frac{1}{4} Cr \right]; R = Cr + G; B = Cb + G$$

- nieodwracalna RGB→YIQ:

$$\begin{aligned} Y &= 0.299 R + 0.587 G + 0.114 B \\ I &= 0.569 R - 0.274 G - 0.322 B \\ Q &= 0.212 R - 0.523 G + 0.311 B \end{aligned}$$

oraz YIQ→RGB:

$$\begin{aligned} R &= 1.000 Y + 0.956 I + 0.621 Q \\ G &= 1.000 Y - 0.272 I - 0.647 Q \\ B &= 1.000 Y - 1.105 I + 0.702 Q \end{aligned}$$

- nieodwracalna RGB→XYZ:

$$\begin{aligned} X &= 0.431 R + 0.342 G + 0.178 B \\ Y &= 0.222 R + 0.707 G + 0.071 B \\ Z &= 0.020 R + 0.130 G + 0.939 B \end{aligned}$$

oraz XYZ→RGB:

$$\begin{aligned} R &= 3.063 X - 1.393 Y - 0.476 Z \\ G &= -0.969 X + 1.876 Y + 0.042 Z \\ B &= 0.068 X - 0.229 Y + 0.069 Z \end{aligned}$$

- odwracalna RGB→CMY:

$$C = 1 - R; M = 1 - G; Y = 1 - B$$

oraz CMY→RGB:

$$R = 1 - C; G = 1 - M; B = 1 - Y$$

Ponadto, w cz. I standardu zastosowano dwie rodziny filtrów. Do kompresji stratnej stosuje się bank filtrów 9/7 (dolnoprzepustowy filtr analizy jest rzędu 9, natomiast górnoprzepustowy – rzędu 7) realizujący transformację rzeczywistoliczbową. W tabeli 2.1. przedstawiono współczynniki tego banku filtrów.

Tabela 2.1. Współczynniki banku filtrów 9/7 ze standardu JPEG2000 (cz. I).

Nr współczynnika	Współczynniki filtrów analizy		Współczynniki filtrów syntezy	
	\tilde{h}	\tilde{g}	h	g
0	0.6029490182363579	1.115087052456994	1.115087052456994	0.6029490182363579
±1	0.2668641184428723	-0.5912717631142470	0.5912717631142470	-0.2668641184428723
±2	-0.07822326652898785	-0.05754352622849957	-0.05754352622849957	-0.07822326652898785
±3	-0.01686411844287495	0.09127176311424948	-0.09127176311424948	0.01686411844287495
±4	0.02674875741080976	-	-	0.02674875741080976

W procesie kompresji bezstratnej stosowany jest bank filtrów 5/3 realizujący transformację całkowitoliczbową. W tabeli 2.2 zamieszczono współczynniki tych filtrów.

Tabela 2.2. Współczynniki banku filtrów 5/3 ze standardu JPEG2000 (cz. I).

Nr współczynnika	Współczynniki filtrów analizy		Współczynniki filtrów syntezy	
	\tilde{h}	\tilde{g}	h	g
0	6/8	1/2	1/2	6/8
±1	2/8	-1/4	1/4	-2/8
±2	-1/8	-	-	-1/8

W standardzie JPEG 2000 zastosowano kwantyzator równomierny według schematu DUTQ (ang. *deadzone uniform threshold quantization*) - jest on modyfikacją prostego schematu UTQ.

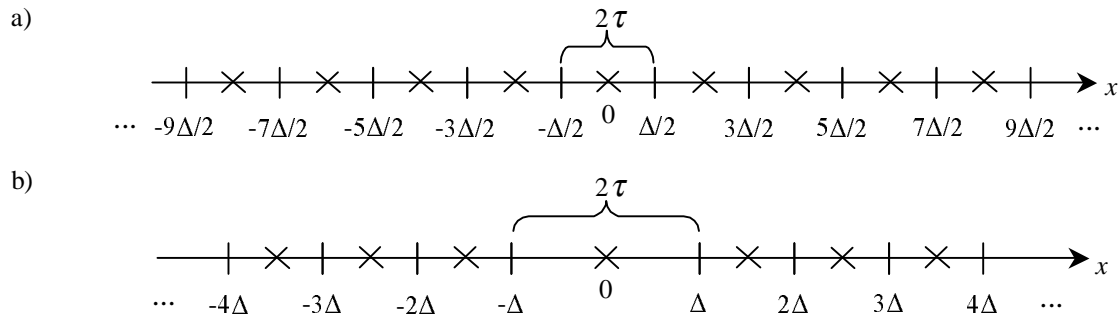
Schemat UTQ W przypadku schematu UTQ funkcje kwantyzacji kodera i dekodera dla danego przedziału kwantyzacji Δ są następujące:

$$d = Q_K(x) = \text{znak}(x) \left\lfloor \frac{|x| + 0.5\Delta}{\Delta} \right\rfloor$$

oraz

$$\tilde{x} = \overline{Q_K^{-1}}(d) = \begin{cases} 0, & d = 0 \\ \text{znak}(d)(|d| + \delta)\Delta, & d \neq 0 \end{cases}$$

gdzie δ jest dobieranym przez użytkownika parametrem z zakresu $-0.5 \leq \delta < 0.5$, d – indeks kwantyzacji, a \tilde{x} - wartość rekonstruowana. Typowa wartość $\delta = 0$, choć modyfikacja tej wielkości może nieznacznie poprawić jakość obrazów rekonstruowanych. Sposób działania kwantyzatora UTQ został zobrazowany na rys. 2.4a).



Rys. 2.4. Schematy charakteryzujące pracę kwantyzatora: a) UTQ, b) DUTQ z zaznaczoną szerokością przedziału zerowego 2τ .

Schemat DUTQ W przypadku kwantyzatora DUTQ funkcje kodera i dekodera są tak zmodyfikowane, aby zwiększyć przedział zerowy. Standardowy (cz.I) sposób zwiększenia przedziału zerowego realizowany jest według zależności:

$$d = \text{znak}(x) \left\lfloor \frac{|x|}{\Delta} \right\rfloor.$$

Współczynniki falkowe o wartościach z zakresu $(-\Delta, \Delta)$ (o wartości modułu mniejszej niż próg $\tau = \Delta$) są zerowane, co oznacza zwiększenie przedziału zerowego do 2Δ ($B_{zero} = 2\Delta$), podczas gdy szerokość pozostałych przedziałów wynosi Δ ($B_{podst} = \Delta$). Rozszerzenie przedziału zerowego

$\eta = \frac{\tau}{\Delta} = \frac{B_{zero}}{2B_{podst}} = 1$, co widać na rys. 2.4b). Funkcja dekodera określona jest jak niżej:

$$\tilde{x} = \overline{Q_K^{-1}}(d) = \begin{cases} 0, & d = 0 \\ \text{znak}(d)(|d| + \delta)\Delta, & d \neq 0 \end{cases}$$

przy czym wartość δ dobierana jest z zakresu $0 \leq \delta < 1$ (zwykle $\delta = 0.5$).

Schemat ten można zmodyfikować (cz. II) dobierając statycznie lub dynamicznie szerokość zerowego przedziału kwantyzacji przy zachowaniu szerokości pozostałych przedziałów równej Δ . Uogólniona postać kwantyzatora DUTQ definiowana jest według zależności:

$$d = \begin{cases} 0, & |x| < -k\Delta \\ \text{znak}(x) \left\lfloor \frac{|x| + k\Delta}{\Delta} \right\rfloor, & |x| \geq -k\Delta \end{cases}$$

oraz

$$\tilde{x} = \begin{cases} 0, & d = 0 \\ \text{znak}(d)(|d| - k + \delta)\Delta, & d \neq 0 \end{cases}$$

Szerokość przedziału zerowego wynosi teraz $2(1-k)\Delta$. Wartość k zmieniana w zakresie $k \in (-1,1)$ pozwala regulować szerokość przedziału zerowego w zakresie $B_{zero} \in (0,4\Delta)$, przy czym: dla $k = 0$ uzyskujemy $B_{zero} = 2\Delta$, ujemne wartości k powodują zwiększenie szerokości przedziału zerowego, a dodatnie – jego zmniejszenie.

Po kwantyzacji dane zostają podzielone na dzielnice (ang. *precinct*), a te z kolei na bloki kodowe (ang. *code block*). Każdy blok kodowy to zbiór płaszczyzn bitowych (ang. *bit plane*), które są niezależnie kodowane w koderze arytmetycznym. W wyniku zakodowania jednego bloku kodowego powstaje zbiór pakietów (ang. *packet*). Każdy pakiet należy do danego poziomu jakości – *warstwy* (ang. *layer*). Warstwa jest zbiorem wszystkich pakietów kolejnych bloków kodowych z danego poziomu jakości. Zbiór wszystkich pakietów (uzupełniony nagłówkiem z informacjami o obrazie oraz parametrach kompresji) tworzy strumień kodowy (ang. *code stream*) – obraz w postaci skompresowanej (plik zawierający taki strumień ma zazwyczaj rozszerzenie JPC).

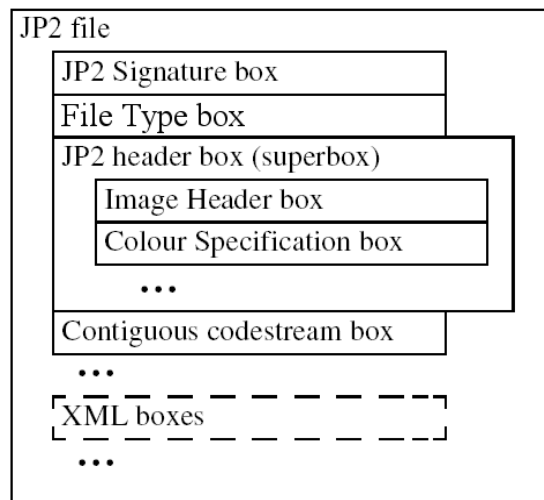
Ze względu na dowolny wzajemny porządek poszczególnych pakietów w strumieniu kodowym (organizacja kodu), możliwe jest kodowanie progresywne. Informacja może się zwiększać poprzez stopniowy przyrost jakości rekonstrukcji (co oznacza grupowanie pakietów należących do tej samej warstwy - progresja LRCP), przyrost rozdzielczości (grupowanie pakietów poszczególnych poziomów dekompozycji – progresja RLCP lub RPCL), powiększenie zdekodowanego obszaru (grupowanie pakietów poszczególnych dzielnic lub części - PCRL) bądź kodowanie kolejnych komponentów (grupowanie pakietów w obrębie komponentów - CPLR). Jeśli obraz został skompresowany z progresją RPCL, to w strumieniu kodowym pakiety zostały zgrupowane według poziomów dekompozycji (rozdzielczości, R_{PCL} – *resolution*) – od najwyższego poziomu dekompozycji (najniższa rozdzielczość) do najniższego (rozdzielczość oryginalna). Każda taka grupa składa się z podgrup pakietów ułożonych według położenia (R_{PCL} – *position*), czyli należących do kolejnych (w porządku rastrowym) dzielnic. Analogicznie, proces przyporządkowania pakietów powtarza się względem komponentów (R_{PCL} – *component*) oraz, na końcu, warstw (R_{PCL} – *layer*).

2.2. STRUKTURA PLIKU JP2

Jak już wspomniano skompresowany obraz w standardzie JPEG2000 jest ciągiem pakietów, które tworzą strumień kodowy. Strumień ten może być zapisany do pliku z rozszerzeniem JPC i odczytany przez dekompresor. Plik taki nie może jednak przechowywać żadnych dodatkowych informacji (np. opisu obrazu). W standardzie JPEG2000 zdefiniowano format JP2 (część I), który grupuje dane różnego typu w jednym pliku.

Struktura pliku JP2 to ciąg pudełek (ang. *box*) – rysunek 2.4. Niektóre z nich mogą zawierać inne pudełka i są nazywane super pudełkami (ang. *superbox*). Początek pierwszego pudełka stanowi pierwszy bajt pliku. Pudełko jest zbiorem danych składającym się z trzech części: długości (4 bajtowej liczby określającej łączną długość całego pudełka), typu (4 bajtowego pola określającego zawartość pudełka) oraz danych.

Część pudełek jest wymagana i musi się znaleźć w pliku JP2 (m.in. ‘sygnatura’, ‘nagłówek’, ‘strumień kodowy’); część jest opcjonalna (m.in. ‘dane XML’). Generalnie porządek pudełek w pliku jest dowolny, jest jednak kilka wyjątków (‘sygnatura’ musi być pierwszym pudełkiem w pliku, ‘typ pliku’ musi się znajdować zaraz za pudełkiem ‘sygnatura’, natomiast ‘nagłówek’ nie powinien się znajdować za ‘strumieniem kodowym’). Każdy plik JP2 może zawierać dowolną liczbę pudełek ‘strumień kodowy’, ‘dane XML’ lub innych zdefiniowanych na potrzeby konkretnej aplikacji.



Rys. 2.4. Struktura pliku JP2.

3. PROGRAM JPER 2000 (autor Dominik Rives)

Program JPER 2000 jest praktyczną realizacją kodera JPEG2000 z wykorzystaniem biblioteki JasPer (w wersji 1.500.4). Jest aplikacją z graficznym interfejsem użytkownika (GUI, ang. *graphic user interface*) typu MDI (ang. *multiple document interface*). Umożliwia więc wygodną pracę z wieloma obrazami równocześnie. Aplikacja może pracować w każdym systemie obsługującym programy Win32. Ponieważ JPER 2000 ma obsługiwać także obrazy medyczne, możliwa jest kompresja i dekompresja obrazów o różnej liczbie bitów na próbkę oraz dowolnej liczbie komponentów, a także dodanie opisu obrazu do pliku (w języku XML) oraz określenia zniekształceń w obrazie zrekonstruowanym. Do operacji kompresji oraz dekompresji obrazów w standardzie JPEG zastosowano bibliotekę IJG JPEG (wersja 6b).

3.1. BIBLIOTEKA JASPER

Pierwotnie biblioteka JasPer (<http://www.ece.uvic.ca/~mdadams/jasper/>) została napisana przez Michaela Davida Adamsa przy współudziale firmy Image Power oraz The University of British Columbia. Obecnie jest rozwijana przez szereg osób w The Department of Electrical and Computer Engineering na The University of Victoria. Pracami nad projektem nadal koordynuje Michael David Adams. JasPer jest oficjalną implementacją standardu JPEG 2000 i został włączony do jego 5 części.

Biblioteka jest udostępniana w postaci źródłowej w języku C i może być kompilowana na platformie Windows, a także praktycznie każdej platformie Unix (także Linux). Te dwie cechy oraz bezpłatna i liberalna licencja spowodowały o wyborze biblioteki JasPer do realizacji programu.

Korzystanie z biblioteki JasPer polega na wywoływaniu jej poszczególnych funkcji. Wszystkie użyteczne z poziomu aplikacji funkcje (tworzące programowy interfejs biblioteki) można podzielić na kilka grup:

- *Funkcje inicjująca i kończąca pracę biblioteki.*
Funkcja inicjująca (**jas_init()**) jest odpowiedzialna za początkowe ustawienie zmiennych globalnych (m.in. inicjowanie tabeli obsługiwanych formatów obrazów). Funkcja kończąca pracę biblioteki (**jas_image_clearfmts()**) ma za zadanie zwolnić pamięć zaalokowaną w momencie inicjacji biblioteki.
- *Funkcje odpowiedzialne za operacje plikowe.*
Służą m.in. do otwierania, zamykania i tworzenia plików. Są odpowiednikami funkcji standardowych języka C.
- *Funkcje dekodująca oraz kodująca wraz z funkcjami pomocniczymi.*
Zestaw najważniejszych funkcji. Pierwsza z nich (**jas_image_decode()**) dekoduje otwarty plik i wypełnia strukturę obrazu (**jas_image_t**, struktura ta zawiera informacje o geometrii obrazu oraz sam obraz w postaci bitmapy). Funkcja kodująca (**jas_image_encode()**) zapisuje obraz zawarty w strukturze do otwartego pliku. W ten sposób może być zdekodowany/zakodowany obraz w każdym formacie obsługiwanym przez bibliotekę (format obrazu jest określony przez jeden z parametrów wywołania obu funkcji). Każdy format jest obsługiwany przez osobny zestaw funkcji (oczywiście można

dodawać nowe zestawy obsługujące inne formaty), z które wywołuje funkcja dekodująca lub kodująca. Pozostałe parametry funkcji to opcje dekodowania i kodowania, w tym parametry kompresji. Uzupełnieniem do obu wymienionych są funkcje zarządzające strukturą obrazu (służące do utworzenia obrazu „pustego”, dodania lub skasowania komponentu, kopiowania komponentów między obrazami itp.).

- *Funkcje operacji na komponentach.*

Ta grupa funkcji umożliwia dokonanie operacji na komponentach (zerowania wszystkich próbek, przesuwania bitowego itp.), które są reprezentowane przez tablicę próbek. Tablica taka (struktura **jas_matrix_t**) musi być wcześniej utworzona (przy pomocy funkcji **jas_matrix_create()**) oraz wypełniona wartościami wszystkich próbek (**jas_image_readcmpt()**) zawartymi w strukturze **jas_image_t**. Możliwy jest też dostęp do pojedynczej próbki (**jas_matrix_get()** oraz **jas_matrix_set()**).

Taka budowa biblioteki pozwala na w miarę swobodne budowanie programu a także łatwą modyfikację samej biblioteki. Poniżej przedstawiono wszystkie zmiany (wraz z krótkim opisem), które zostały wprowadzone do biblioteki w celu rozszerzenia jej możliwości:

- *Modyfikacja zestawu funkcji odczytującej pliki obrazów w formacie PNM.*

Funkcja odczytująca pliki PNM (**pnm_getdata()**) została rozszerzona w celu umożliwienia odczytu plików o dowolnej ilości bitów na próbkę (taki format pliku PNM jest wykorzystywany przez JLS Encoder kompresujący pliki do standardu JPEG-LS).

- *Dodanie możliwości odczytu plików obrazów w formacie JPEG.*

Biblioteka JasPer została uzupełniona o zestaw funkcji obsługujących format JPEG. Do odczytu plików w standardzie JPEG, zastosowano bibliotekę IJG JPEG w wersji 6b (<http://www.iijg.org/>) autorstwa Thomasa G. Lane.

- *Rozszerzenie listy obsługiwanych transform przestrzeni kolorów.*

Dodano transformaty przestrzeni kolorów na YIQ (nieodwracalna), XYZ (nieodwracalna) i CMY (odwracalna). Modyfikacja ta wymagała zmian w funkcjach kodujących i dekodujących kafelki (**jpc_dec_tileencode()** i **jpc_dec_tileddecode()**) oraz dodania funkcji przeprowadzającej transformację przestrzeni kolorów (z (**koda_transform()**) i na (**koda_transform_back()**) przestrzeń RGB).

- *Dodanie pełnej obsługi (zapis i odczyt danych) pudełka 'dane XML'.*

Wymagało to drobnej modyfikacji funkcji zapisującej dane do pliku JP2 (**jp2_encode()**) oraz całkowitego przeprojektowania funkcji odczytującej plik formatu JP2 (**jp2_decode()**). Ponadto konieczne było oczywiście napisanie funkcji zapisującej pudełko 'dane XML' do strumienia (tzn. wyliczającej długość pudełka, wpisującej jego nagłówki oraz właściwe dane XML; **write_xml_box()**). Odczyt pudełka 'dane XML' jest realizowany przez zmodyfikowaną funkcję odczytującą kolejne pudełka z pliku (**jp2_box_get()**). Modyfikacja wymienionej funkcji polegała na dodaniu funkcji alokującej niezbędną pamięć na dane XML oraz kodu przepisującego dane do zaalokowanej pamięci. Konieczne okazało się także uzupełnienie tabeli definicji typów pudełek o definicję pudełka 'dane XML'.

Podczas prac nad programem JPER 2000 został wykryty błąd w bibliotece JasPer, który uniemożliwiał otwieranie plików znajdujących się na nośniku zabezpieczonym przez zapisem. Jak się okazało, błąd powodowała funkcja **tmpfile()**, która tworzy plik tymczasowy. W kompilatorach środowiska Unix funkcja ta zawsze tworzy plik w katalogu dla plików tymczasowych, niezależnie od ścieżki katalogu bieżącego. Funkcja **tmpfile()** dostarczona w pakiecie MS Visual Studio C++ próbuje założyć plik nie w katalogu tymczasowym, lecz bieżącym. W przypadku, kiedy katalog bieżący jest na nośniku 'tylko do odczytu', próba uwożenia pliku kończy się niepowodzeniem. Powyższy problem został rozwiązany poprzez tymczasową zmianę katalogu bieżącego na katalog dla plików tymczasowych.

JasPer nie obsługuje niestety regionów zainteresowania. Ze względu na konstrukcję biblioteka nie rozpocznie dekompresji, jeśli nie otrzyma całego skompresowanego pliku. Nie udostępnia także obrazu przed całkowitą rekonstrukcją (tzn. nie udostępni obrazu przez zdekodowaniem wszystkich żądanych pakietów lub warstw). Ostatnią wadą biblioteki jest jej stosunkowo wolne działanie.

3.2. PRACA Z PROGRAMEM JPER 2000

Praca z programem JPER 2000 nie różni się od pracy z innymi programami MDI, aby jednak sprawnie się nim posługiwać należy poznać kilka szczegółów filozofii jego funkcjonowania. Aplikacja traktuje obrazy w różny sposób w zależności od ich formatu: pliki w standardzie JPEG2000 są otwierane jako obrazy zrekonstruowane (okno z takim obrazem ma ikonę z literą 'R'); natomiast

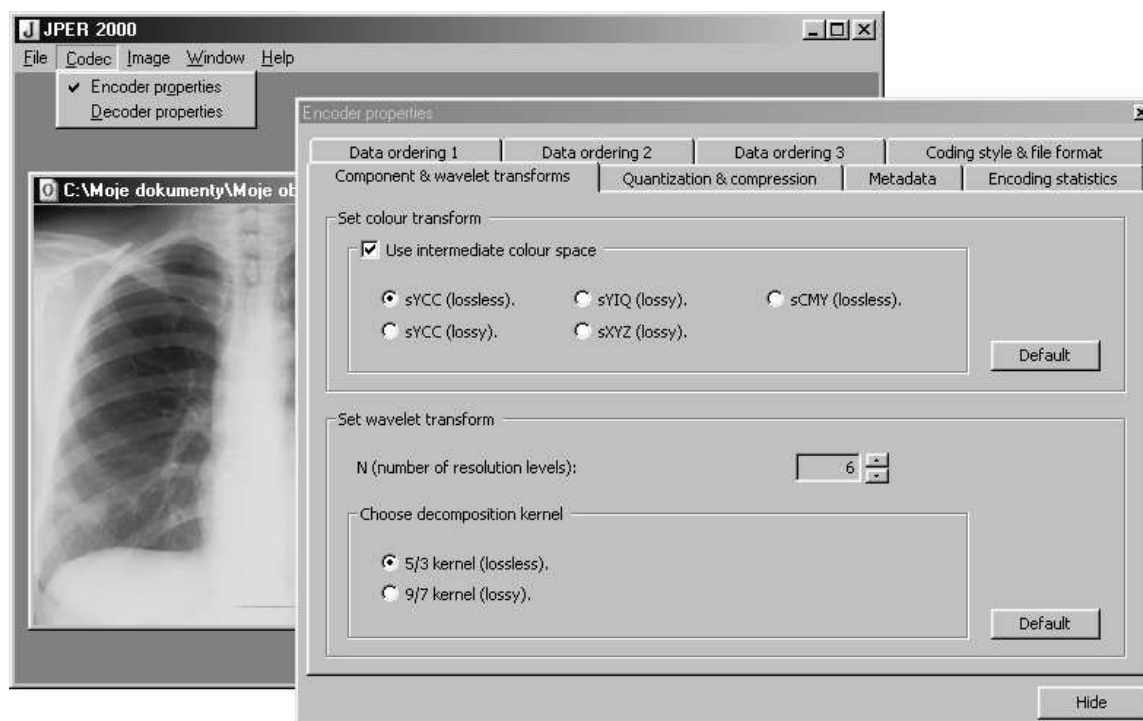
pozostałe obrazy – jako oryginalne (ikona z literą ‘O’); rysunek 3.1. Rozróżnienie wynika z tego, że obrazy oryginalne mogą być tylko kompresowane, natomiast zrekonstruowane – tylko dekompresowane.



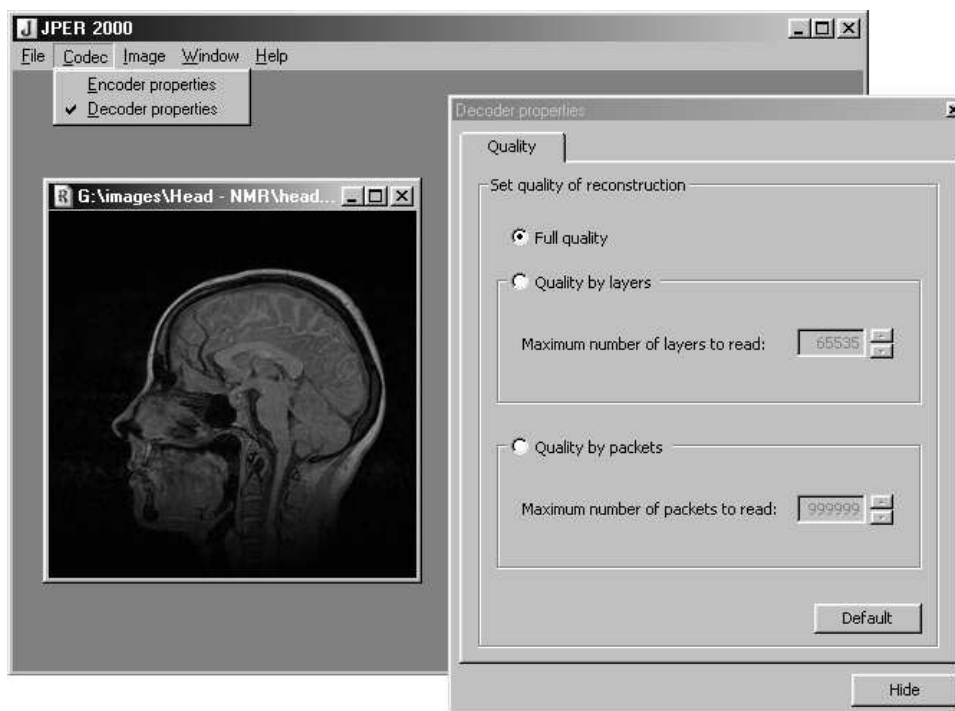
Rys. 3.1. Aplikacja z dwoma otwartymi obrazami i zaznaczonymi ikonami.

Od momentu uruchomienia programu, użytkownik ma stały dostęp do parametrów kodera oraz dekodera (niezależnie od otwartych okien obrazów). Parametry te są dostępne odpowiednio w okienkach ‘Encoder properties’ oraz ‘Decoder properties’. Ponieważ są to okna typu ‘modeless’ nie muszą być zamykane, aby użytkownik miał dostęp do poleceń aplikacji. Na rysunku 3.2 przedstawiono okienko kodera, natomiast na rysunku 3.3 – dekodera.

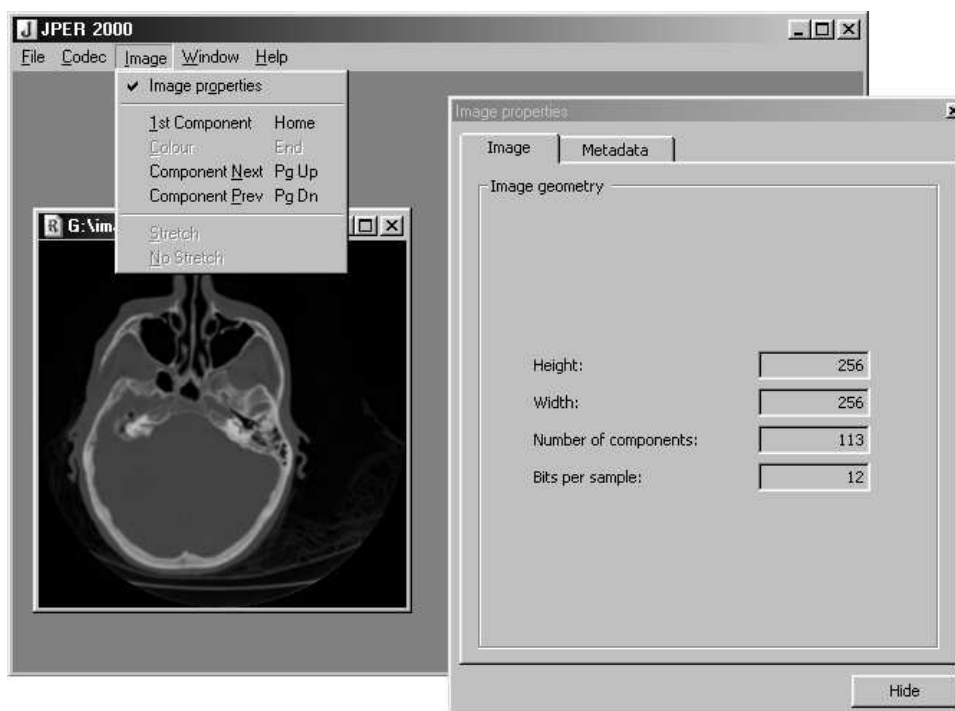
Właściwości obrazów są podawane w okienku ‘Image properties’. To okienko także jest typu ‘modeless’ i zawsze są w nim właściwości aktywnego okna obrazu; zmiana aktywnego okna wymusza zmianę podawanych w oknie ‘Image properties’ wartości. Okno właściwości obrazu zostało pokazane na rysunku 3.4.



Rys. 3.2. Okienko wyboru parametrów kompresji.



Rys. 3.3. Okienko wyboru parametrów dekompresji.



Rys. 3.4. Okienko właściwości obrazu.

Poniżej przedstawiono listę właściwości oraz możliwości programu.

- *Otwieranie obrazów.*

Obraz otwiera się przy pomocy polecenia '**File | Open...**'. Aplikacja obsługuje obrazy w następujących formatach:

- Bitmapa-Windows lub OS/2 (pliki BMP),
- Portable Greymap/Pixelmap/Anymap (pliki PGM/PPM/PNM, popularny format nieskompresowanych obrazów rastrowych),

- Portable Greymap/Pixelmap (pliki PGM/PPM w wersji niestandardowej (o dwubajtowych próbkach), używanej przez JLS Encoder),
- Portable X-map (pliki PGX to modyfikacja formatu Portable Greymap przystosowana do obrazów od dowolnej liczbie bitów na próbkę i jednym komponentie),
- JPEG (baseline, extended-sequential, progressive mode);
- JPEG 2000 (w postaci czystego strumienia kodowego lub pliku JP2).

Aplikacja oferuje także możliwość jednoczesnego wczytania dowolnej liczby obrazów (np. serii obrazów tomograficznych) i kompilacji ich do jednego obrazu składającego się z wielu komponentów przy pomocy polecenia: **'File | Open Group...'**. Program jest przystosowany do odczytu i wyświetlania obrazów o maksymalnej liczbie komponentów i bitów na próbkę przewidzianej przez standard JPEG 2000. Ponadto, obraz może być wczytany poprzez podanie go jako argumentu programu; dzięki temu obsługiwane przez aplikację formaty mogą być z nią skojarzone.

- *Wybór trybu wyświetlania obrazu.*

Istnieje możliwość wyboru wyświetlanego komponentu (własność szczególnie przydatna w przypadku obrazów o niestandardowej liczbie komponentów). Można wyświetlać dowolny pojedynczy komponent obrazu lub jego trzy pierwsze składowe (polecenia **'Image | 1st Component/Colour/Component Next /Component Prev'**).

- *Wybór parametrów kompresji.*

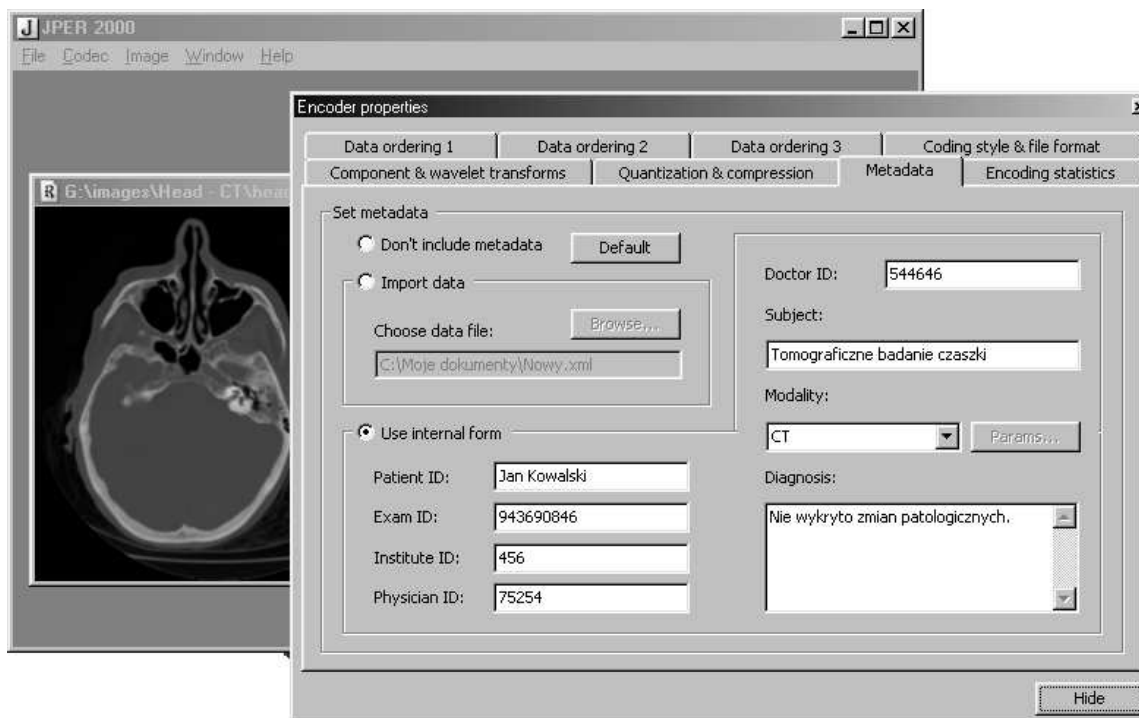
Parametry kodera są dostępne po otwarciu okienka 'Encoder properties' (polecenie **'Codec | Encoder properties'**, rysunek 2). Dostępne parametry:

- rodzaj transformaty kolorów (brak, YCC_R , YCC_I , YIQ_I , XYZ_I , CMY_R) (**'Component & wavelet transforms | Set colour transform'**);
- rodzaj transformaty falkowej (5/3 lub 9/7), liczba poziomów dekompozycji (1...14) (**'Component & wavelet transforms | Set wavelet transform'**);
- liczba bitów bezpieczeństwa (0...7) (**'Quantization & compression | Set quantization'**);
- stopień kompresji (1...999) (**'Quantization & compression | Set compression'**);
- położenie obrazu na siatce odniesienia (odległość od początku układu odniesienia w zakresie 0...99999, określana niezależnie dla obu współrzędnych) (**'Data ordering 1 | Set image origin on reference grid'**);
- podział obrazu na kafelki (brak; kafelki o wielkości 1...99999 próbek, określonej niezależnie dla obu współrzędnych) (**'Data ordering 1 | Set tile partitioning on reference grid'**);
- podział podpasma na dzielnice (brak; dzielnice o wielkości $2^{0...15}$, określonej niezależnie dla obu współrzędnych) (**'Data ordering 2 | Set precinct partition dimensions'**);
- podział podpasma na bloki kodowe (bloki o wielkości $2^{2+0...8}$, określonej niezależnie dla obu współrzędnych) (**'Data ordering 2 | Set code-block dimensions'**);
- liczba dodatkowych warstw (brak, 1...99) oraz trend przyrostu informacji w obrazie (logarytmiczny – pierwsze warstwy niosą więcej informacji niż końcowe, liniowy – każda warstwa niesie podobną ilość informacji lub eksponentalny – najwięcej informacji przynoszą końcowe warstwy) (**'Data ordering 3 | Extra Layers'**);
- kolejność przyrostu informacji w obrazie (wg jakości rekonstrukcji, rozdzielczości (dwie wersje), obszaru lub komponentu) (**'Data ordering 3 | Choose progression order'**);
- styl kodowania; zawiera szereg opcji określających sposób kodowania arytmetycznego oraz tworzenia strumienia kodowego, dostępne są m.in. opcje pozwalające na uodpornienie strumienia na błędy transmisji (**'Coding style | Set coding style & file format'**);
- format pliku (czysty strumień kodowy lub JP2) (**'Coding style | Set coding style & file format'**);

- *Opis zawartości kompresowanego obrazu.*

Istnieje możliwość dodania do pliku informacji o kompresowanym obrazie (w języku XML). Można skorzystać albo z wewnętrznego formularza (dostępnego w okienku 'Encoder properties' (**'Codec | Encoder properties'**) na zakładce **'Metadata'** w ramce **'Use internal form'**) lub wykorzystać dowolny plik w formacie XML (ramka **'Import data'**). Wewnętrzny formularz ma szereg pól, w które można wprowadzić informacje administracyjne (dotyczące pacjenta, badania,

instytucji, osoby bezpośrednio przeprowadzającej badanie i lekarza stawiającego rozpoznanie) oraz temat badania, modalność i diagnozę (rysunek 3.5). Informację można dodać tylko w przypadku kompresji do pliku w formacie JP2.

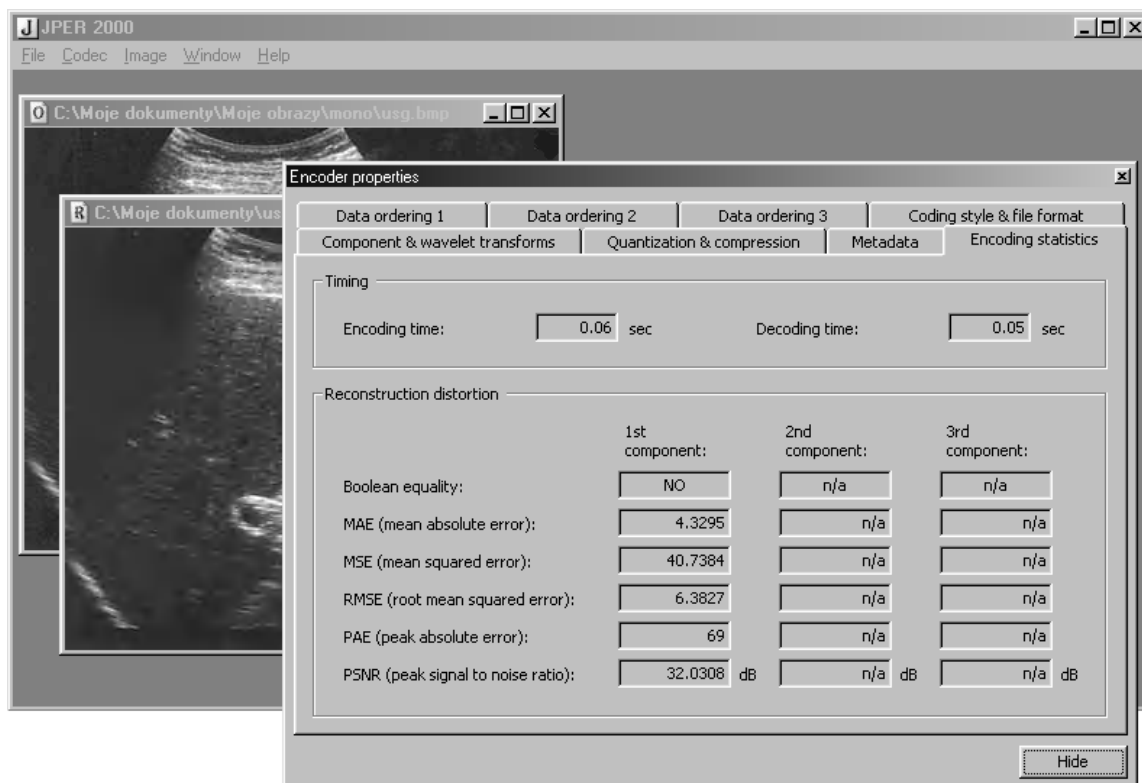


Rys. 3.5. Okienko parametrów kompresji, zakładka 'Metadata'.

- *Kompresja.*
Obraz można skompresować do standardu JPEG 2000 korzystając z polecenia **'File | Encode & Save...'**. Parametry kompresji są oczywiście określone w okienku 'Encoder properties'. Bezpośrednio przed kompresją program prosi o podanie pliku docelowego. Po zakończonej kompresji pojawia się okno z obrazem zrekonstruowanym natomiast w okienku 'Encoder properties' (**'Codec | Encoder properties'**) na zakładce **'Encoding statistics'** w ramce **'Timing'** są podane czasy kompresji oraz dekompresji (ponieważ program pracuje w środowisku wielozadaniowym, a także ze względu na interwał podawania czasu przez zegar systemowy wynoszący prawie 55 milisekund, dokładność podawanych czasów należy traktować wyłącznie orientacyjnie), natomiast w ramce **'Reconstruction distortion'** pojawia się szereg wartości błędów rekonstrukcji dla poszczególnych komponentów. Wyliczane są następujące miary: boolean equality (informacja o bitowej identyczności obrazów), MAE, MSE, RMSE, PAE oraz PSNR. Miara PSNR jest podawana w decybelach (rysunek 3.6).
- *Wybór parametrów dekompresji.*
Parametry dekodera są dostępne po otwarciu okienka 'Decoder properties' (polecenie **'Codec | Decoder properties'**, rysunek 3.3). Parametry dekompresji sprowadzają się do wyboru żądanej jakości, w jakiej zostanie zdekodowany obraz. Są trzy możliwości:
 - pełna jakość dekompresji (**'Quality | Full quality'**);
 - jakość ograniczona liczbą zdekodowanych warstw (1...65535) (**'Quality | Quality by layers'**);
 - jakość ograniczona liczbą zdekodowanych pakietów (1...999999) (**'Quality | Quality by packets'**);
- *Dekompresja.*
Dekompresja obrazu formatu JPEG 2000 odbywa się po wybraniu polecenia **'Processing | Decode & Save...'**. Parametry dekompresji są określone we wcześniej omówionym okienku 'Decoder properties'. Bezpośrednio przed dekompresją program prosi o wyznaczenie pliku docelowego i jego formatu. Obraz może być zapisany w następujących formatach:

- Windows Bitmap (pliki BMP)
- Portable Anymap (pliki PNM)
- Portable X-map (pliki PGX)

Nie każdy jednak obraz będzie mógł być zapisany w każdym wyżej przedstawionym formacie (i wobec tego nie wszystkie formaty będą w okienku dialogowym dostępne). Do pliku w formacie BMP może być zapisany wyłącznie obraz zawierający 1 lub 3 komponenty o 8 bitach na próbkę. W formacie PNM mogą być zapisane tylko obrazy o 1 lub 3 komponentach o maksymalnie 8 bitach na próbkę. Format PGX jest najbardziej uniwersalny (akceptuje dowolną liczbę bitów na próbkę) i można w nim zapisać każdy obraz. Ponieważ w pliku formatu PGX mogą być zapisywane wyłącznie obrazy o jednym komponentcie; w przypadku dekompresji obrazu o więcej niż jednym komponentcie generowana jest seria plików PGX. Każdy wygenerowany w ten sposób plik stanowi jeden komponent obrazu oryginalnego (numer komponentu jest dodany do nazwy).



Rys. 3.6. Okienko parametrów kompresji, zakładka 'Encoding statistics'.

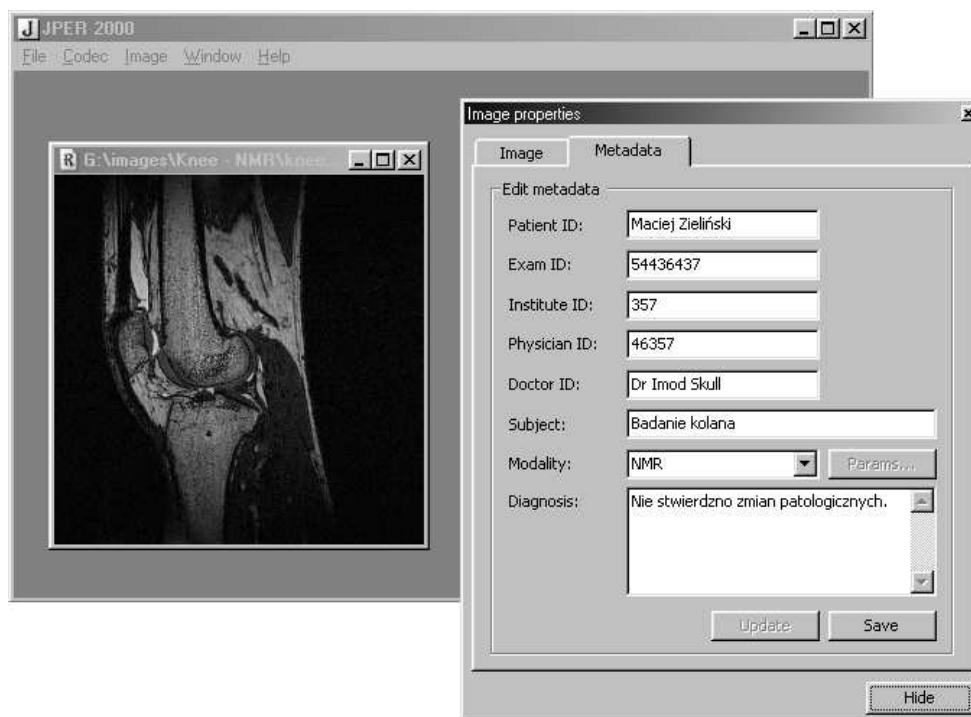
- *Informacje o obrazie.*

O każdym otwartym obrazie można uzyskać informacje na temat jego geometrii (okienko 'Image properties' ('Image | Image properties'), zakładka 'Image'). Jeśli dany plik jest w formacie JP2 i zawiera dane XML mogą być one zapisane do pliku w formacie XML (zakładka 'Metadata', przycisk 'Save'). Jeśli dane te zostały wprowadzone przy użyciu formularza dostępnego w okienku 'Encoder properties', zostaną one przedstawione – rysunek 3.7. Jeśli plik nie jest chroniony przed zapisem, dane mogą być modyfikowane (należy pamiętać, aby po modyfikacji informacji kliknąć na przycisk 'Update').

3.3. MOŻLIWOŚCI ROZWOJU

Choć przedstawiony program jest w pełni funkcjonalny, jego możliwości mogą być rozszerzone. Program powinien odczytywać więcej formatów obrazów (włączając formaty tomografów oraz innych obrazowych aparatów diagnostycznych). Szczególnie istotne wydaje się dodanie obsługi formatu RAW ('dane surowe'), która umożliwi odczyt dowolnych plików niezależnie od ich formatu (oczywiście chodzi o formaty bez kompresji). Kolejnym problemem do rozwiązania jest prezentacja

obrazu. W obecnej postaci użytkownik nie może swobodnie kontrolować skali obrazu. W przypadku obrazów o większej od 8 liczbie bitów na próbkę doskwiera także brak obsługi okienkowania.



Rys. 3.7. Okienko właściwości obrazu, zakładka 'Metadata'.

Pewną niekonsekwencją w interfejsie programu jest to, że wartości zniekształcenia są podawane w okienku kodera ('Encoder properties') – powinny być one umieszczone w okienku właściwości obrazu ('Decoder properties') lub innym, oddzielnym okienku oraz pozwalać na porównanie dowolnych obrazów (aplikacja w obecnej postaci umożliwia jedynie porównanie obrazu skompresowanego i zrekonstruowanego z oryginalnym).

Program należy także wyposażyć w większą możliwość ingerencji w kompresję oraz dekompresję. Dotyczy to szczególnie procesu kwantyzacji (wybór dowolnego kroku kwantyzacji, zastosowanie kwantyzatora nierównomiernego) oraz transformacji falkowej (większy wybór filtrów). Program powinien także obsługiwać region zainteresowania. Przystosowanie programu do wyświetlania kolejnych wersji dekompresowanego obrazu jest cechą mniej ważną, aczkolwiek przydatną w przypadku dużych obrazów, których dekodowanie trwa stosunkowo długo.

Na zakończenie należy wspomnieć o obrazach składających się z wielu komponentów. Norma rozróżnia plik zawierający obraz o wielu komponentach (JP2, opisany w I części standardu) oraz plik zawierający serię obrazów (JPM, przedstawiony w III części). Program JPER 2000 umożliwia tworzenie jedynie plików typu JP2, a więc kompresja obrazu zawierającego serię obrazów tomograficznych nie do końca jest zgodna z filozofią normy. Nie znaczy to oczywiście, że tworzony plik jest niezgodny z normą, a jedynie wskazuje na fakt wykorzystania standardu niezgodnie z przeznaczeniem. Ze względu na podobieństwo struktury pliku JP2 i JPM, dodanie obsługi standardu JPM nie powinno być problemem.