



Kompresja JPG obrazu sonarowego z uwzględnieniem założonego poziomu błędu

MARIUSZ BORAWSKI

Zachodniopomorski Uniwersytet Technologiczny w Szczecinie,
Wydział Informatyki, 70-210 Szczecin, ul. Żołnierska 49
mborawski@wi.zut.edu.pl

Streszczenie. W artykule zaprezentowano dwie metody kompresji danych z sonaru wykorzystujące algorytm JPG. Umożliwiają one uzyskanie określonego odchylenia standardowego dla bloków obrazu będącego różnicą obrazu skompresowanego i nieskompresowanego oraz zapewniają utrzymanie się w granicach założonego błędu kompresji. Zaprezentowane metody zmieniają tylko sposób kompresji, dekompresja pozostaje bez zmian. Dzięki temu obraz poddany działaniu tych metod może być dekompresowany przez każdy program służący do tego celu. Dla obrazów sonarowych jest to szczególnie ważne ze względu na możliwość wykorzystania kompresji JPG w standardzie GeoTiff pozwalającym na zapis informacji o współrzędnych geograficznych i orientacji względem północy obrazu sonarowego.

Słowa kluczowe: rozpoznawanie obrazów, sonar, obraz sonarowy, kompresja JPG, GeoTiff

1. Wstęp

Ze względu na długi czas pracy oraz wysokie rozdzielczości systemy sonarowe generują duże ilości danych. Przykładowo, sonar boczny EdgeTech 272-TD może rejestrować ponad 43 tys. próbek na sekundę, co daje ponad dwa i pół miliona próbek na minutę. Powstają w ten sposób gigabajty danych, które muszą być archiwizowane.

Zastosowanie kompresji może znacząco zmniejszyć rozmiar danych, jednak w przypadku zastosowania metod stratnych zysk jest niewielki (dla przykładowego obrazu sonarowego otrzymano zmniejszenie o 24% dla kompresji LZW i 33% dla kompresji ZIP). Ograniczeniem do stosowanych metod kompresji jest konieczność wykorzystania standardu zapisu pliku graficznego z możliwością zawarcia

w nim informacji georeferencyjnych (takim formatem jest GeoTiff, kompatybilny z formatem TIFF 6.0). Zawęźa to listę możliwych do zastosowania metod kompresji do metod zawartych w specyfikacji formatu. Wśród tych metod można wymienić [17]: kompresję LZW, JPG, CCITT RLE, CCITT T.4, CCITT T.6, oraz PackBits. Implementowana jest także kompresja ZIP oraz Zlib, a kompresja JPG jest zastępowana unowocześnioną wersją [8].

W przypadku surowych danych sonarowych format zapisu determinowany jest przez producenta sonaru. Zwykle producenci zapisują dane w plikach opartych na własnej specyfikacji. Część z tych specyfikacji stała się już nieformalnymi standardami. Przykładowymi typami plików sonarowych mogą być Q-MIPS (rozszerzenie *.dat), Coda (*.cda), XTF (*.xtf), MUSE (*.seg), Mesotech (*.smb) itp.

Dane sonarowe zwykle przetwarza się do postaci mozaiki, która zapewnia prawidłowe odtworzenie kształtów obiektów [15, 16]. Mozaika charakteryzuje się znacznym przyrostem objętości pliku w stosunku do pliku z danymi niepoddanymi obróbce. Jest to związane z koniecznością zastosowania interpolacji dla miejsc, w których brakuje danych. Mozaika zapisywana jest jako zwykły plik graficzny albo plik w formacie GeoTiffa.

Ze wszystkich formatów kompresji, jakie dostarcza TIFF, największe „upakowanie” zapewnia kompresja JPG. Niestety jest to kompresja stratna powodująca zmiany w zapisywanym obrazie. Nie zawsze jednak te zmiany są istotne. Każdy obraz zawiera szum, na powstanie którego składa się wiele przyczyn. Zmiana obrazu nieprzekraczająca poziomu szumu nie ma znaczenia dla jakości obrazu. Jeżeli nie analizuje się samego szumu, to zastosowanie kompresji stratnej zmieniającej obraz na poziomie nieprzekraczającym szumu zapewnia taką samą jakość obrazu jak kompresja bezstratna.

Poziom szumu w przypadku zwykłych zdjęć jest bardzo mały. Dla obrazu sonarowego może przyjmować bardzo duże wartości. Odchylenie standardowe może wynosić do nawet trzech procent dopuszczalnego zakresu wartości. Daje to spore możliwości zmian poziomu wartości opisujących piksele, a zatem pozwala na uzyskanie wysokich wartości współczynników kompresji dla kompresji stratnej. W kompresji obrazu sonarowego wykorzystuje się głównie metody kompresji bezstratnej: LZW, RLE, kompresje oparte o transformaty falkowe, metody słownikowe, deflate/inflate [5-7]. Podejmowane są również próby zastosowania kompresji stratnej: falkowej [9] i JPG [14], jednak bez uwzględniania poziomu szumu jako granicy możliwej zmiany wartości pikseli.

W kompresji zwykłych zdjęć, ze względu na niski poziom szumu, nie odnosi się stopnia kompresji do poziomu szumu, ale do innych parametrów. Przykład takiego podejścia można znaleźć w pracach Rosenholta i Watsona [13, 21]. Stopień kompresji jest tutaj regulowany na podstawie wartości jasności pikseli i kontrastu. Regulacja stopnia kompresji odbywa się poprzez odpowiednią modyfikację macierzy kwantyzacji. Metoda ta wymaga dołączenia do pliku macierzy kwantyzacji, co

nie jest ujęte w standardzie TIFF. Czyni to ją nieprzydatną dla plików zapisanych w GeoTiffie. Podobne rozwiązanie przyjęto w pracy Ramosa i Hemami [12], gdzie tablica kwantyzacji jest budowana na podstawie wykrytych konturów. W niektórych pracach podejmuje się próbę określenia związków pomiędzy rozmiarem pliku a jego jakością, aby umożliwić sterowanie jakością za pomocą określenia oczekiwanej wielkości pliku [4]. Ma to szczególne znaczenie przy przesyłaniu skompresowanej informacji multimedialnej.

Większość modyfikacji metod kompresji w ramach standardu JPG bazuje na zmianie sposobu kwantyzacji, jednak możliwa jest również zmiana sposobu kompresji bezstratnej poprzez zastąpienie algorytmu Huffmana bardziej wydajnym rozwiązaniem. Podejście takie zaprezentowano w pracy Brailovskiy i Plotkina [1].

W artykule zostanie przedstawiona metoda pozwalająca na uzyskanie zadanej wielkości błędu kompresji. Dzięki temu wynik kompresji można uzależnić od oszacowanego poziomu szumu obecnego w obrazie sonarowym. Przedstawiona metoda kompresji nie jest zgodna ze standardem, w przeciwieństwie do dekompresji. Zgodność dekompresji ze standardem umożliwia odczytanie skompresowanych plików przez dowolny program odczytujący pliki TIFF i GeoTiff.

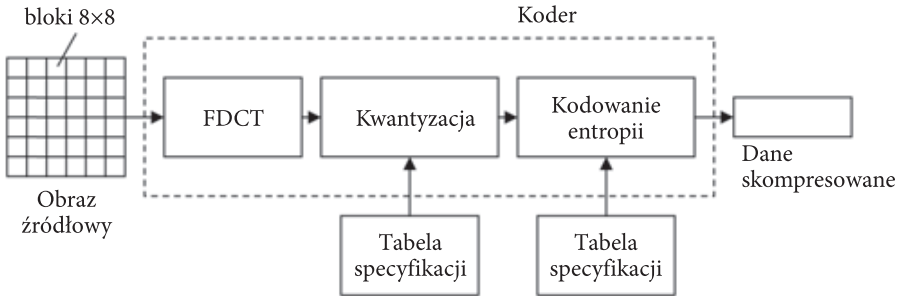
2. Kompresja JPG

Proponowana metoda kompresji składa się z podobnych etapów jak algorytm kompresji zgodny ze standardem JPG. Pierwszym krokiem jest konwersja do modelu YCbCr [10]:

$$\begin{cases} Y &= 0,299R &+& 0,587G &+& 0,114B \\ Cb &= -0,1687R &-& 0,3313G &+& 0,5B &+ 128 \\ Cr &= 0,5R &-& 0,4187G &-& 0,0813B &+ 128. \end{cases} \quad (1)$$

Dla obrazu konwersję wykonuje się ze względu na to, że składowe Cb i Cr mają mniejszą amplitudę niż składowa Y , w związku z tym są silniej kompresowane, ale przy tym strata informacji jest słabiej wyczuwana przez oko ludzkie. Własność tę posiadają zwykłe zdjęcia, dla obrazu sonarowego jej wystąpienie zależy od wybranej palety.

Kolejne etapy kompresji zostały przedstawione na rysunku 1. Obraz dzielony jest na bloki 8×8 , a następnie wartości są przesuwane z zakresu $\langle 0; 2^p - 1 \rangle$ do zakresu $\langle -2^{p-1}; 2^{p-1} - 1 \rangle$, gdzie jest ilością bitów, na których zapisane są składowe koloru obrazu.



Rys. 1. Proces kodowania obrazu [18]

Po przesunięciu wartości wykonywana jest FDCT (Forward DCT) [18]:

$$F(u, v) = \frac{1}{4} C(u) C(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right], \quad (2)$$

gdzie: $f(x, y)$ — elementy bloku 8×8 ,
 $F(u, v)$ — współczynnik FDCT,
 przy czym [18]:

$$C(w) = \begin{cases} \frac{1}{\sqrt{2}} & \text{dla } w = 0 \\ 1 & \text{dla } w \neq 0, \end{cases} \quad (3)$$

Wyliczone współczynniki $F(u, v)$ są poddawane procesowi kwantyzacji z wykorzystaniem macierzy kwantyzacji [19]:

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}, \quad (4)$$

Macierz ta jest wykorzystywana dla obrazów czarno-białych i dla luminancji obrazów kolorowych. Na jej podstawie wyznacza się macierz kwantyzacji Q_{QL} (dla konkretnej „jakości” obrazu):

$$Q_{QL} = wQ, \quad (5)$$

przy czym [3]:

$$w = \begin{cases} \frac{50}{QL} & \text{dla } QL < 50 \\ 1 & \text{dla } QL = 50 \\ \frac{100 - QL}{50} & \text{dla } QL > 50. \end{cases} \quad (6)$$

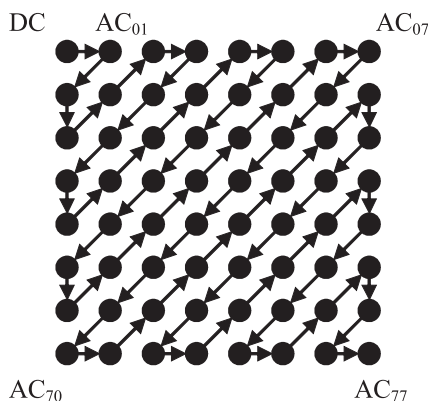
Współczynnik QL określa jakość obrazu po kompresji zadaną przez użytkownika (podawaną w zakresie od 1 do 100).

Macierz (4) uznawana jest za macierz stanowiącą optimum pomiędzy jakością i rozmiarem obrazu. Oznacza się ją jako Q_{50} . Na podstawie macierzy kwantyzacji Q_{QL} wyznacza się skwantowane wartości współczynników $F(u, v)$ [18]:

$$F^Q(u, v) = \left\lceil \frac{F(u, v)}{Q_{QL}(u, v)} \right\rceil, \quad (7)$$

gdzie $\lceil x \rceil$ to zaokrąglenie do wartości całkowitej.

Uzyskane wartości są zaokrąglane do najbliższej wartości całkowitej. Kolejnym etapem w kompresji JPG jest przejście od dwuwymiarowej tablicy $F^Q(x, y)$ do tablicy jednowymiarowej $F^Q(k)$. Wykonuje się to przejście tak, aby składowe wysokoczęstotliwościowe, które po kwantyzacji często przybierają wartości zerowe, znajdowały się obok siebie. Gwarantuje to uzyskanie wyższych współczynników kompresji. Procedurę przejścia z tablicy jednowymiarowej do tablicy dwuwymiarowej zilustrowano na rysunku 2.



Rys. 2. Zamiana tablicy dwuwymiarowej na jednowymiarową [18]

W końcowej części tablicy jednowymiarowej zwykle znajdują się zera lub bardzo małe wartości. Nie zapamiętuje się całej tablicy, ale wszystkie jej elementy do wystąpienia ostatniego niezerowego elementu w tablicy. Ze względu na dużą liczbę zer w zapamiętywanej części tablicy, bardzo dobrze poddaje się ona kompresji bezstratnej. W związku z tym ostatnim etapem kompresji JPG jest kodowanie Huffmanna. Polega ono na utworzeniu słów kodowych, których długość jest tym mniejsza, im większa jest częstotliwość występowania danej wartości reprezentowanej przez słowo kodowe.

3. Kompresja z maksymalną wartością odchylenia standardowego błędu

W kompresji JPG współczynnik QL jest zadawany z góry i niezależny od charakteru obrazu, a jedynie od uznania użytkownika programu. W przypadku obrazu z sonaru sektorowego należy wprowadzić parametr zależny od poziomu szumu obecnego w obrazie sonarowym. Może on być łatwy do oszacowania ze względu na możliwość wielokrotnego sondowania tego samego obszaru w tych samych warunkach. Dla obrazu sonarowego, zamiast parametru QL , lepszym parametrem określającym jakość obrazu byłaby wielkość pozwalająca odnieść się do szumu znajdującego się w obrazie. Takim parametrem może być odchylenie standardowe różnicy obrazu skompresowanego i nieskompresowanego.

Ze względu na to, że mnożenie przez macierz kwantyzacji jest konieczne w celu podbicia wartości składowych wysokoczęstotliwościowych, które bez tej operacji przyjęłyby po kwantyzacji wartości zerowe, nie można wyeliminować całkowicie współczynnika QL . W określaniu jakości kompresji stosowane są dwa współczynniki QL i wartość odchylenia standardowego. Wartość QL powinna być dobrana możliwie wysoko, aby nie spowodowała przekroczenia założonego poziomu odchylenia standardowego. W celu upewnienia się co do właściwej decyzji użytkownika, po kwantyzacji powinno nastąpić sprawdzenie, czy zadana wartość odchylenia standardowego nie została przekroczona. Wykonuje się to poprzez odtworzenie współczynników $F(u, v)$ na podstawie macierzy kwantyzacji:

$$F'(u, v) = F^o(u, v)Q_{QL}(u, v) \quad (8)$$

oraz wyliczenie odwrotnej transformaty kosinusowej [18]:

$$f'(x, y) = \frac{1}{4} \left[\sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F'(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (9)$$

i zaokrąglenie otrzymanych wartości do liczb całkowitych. Otrzymuje się w ten sposób ostateczne wartości współczynników $f''(x, y)$.

Maksymalna wartość odchylenia standardowego σ_{Δ} różnicy obrazów $f(x, y)$ i $f''(x, y)$ powinna być mniejsza od założonego odchylenia standardowego σ_{ϵ} :

$$\sigma_{\Delta} \leq \sigma_{\epsilon}. \quad (10)$$

Jeżeli warunek nie jest spełniony, to powinno się wybrać współczynnik kompresji zapewniający wyższą jakość.

Stopień kompresji można zwiększyć, zerując składowe częstotliwościowe, aż do uzyskania założonego poziomu błędu. Operację tę należy wykonać po przejściu z tablicy dwuwymiarowej do jednowymiarowej. Przypisywanie wartości zero należy rozpocząć od końcowej części tablicy. Znajdują się tam składowe wysokoczęstotliwościowe związane z szumem obecnym w obrazie sonarowym. Pojawia się jednak problem określenia poziomu błędu przy zmianie ich wartości. W tym celu konieczne jest, po każdej zmianie współczynnika, wyliczenie $f'(x, y)$ oraz wyznaczenie na jego podstawie błędu, co wymaga znacznego nakładu obliczeniowego.

Alternatywnym podejściem jest wykorzystanie do tego celu własności niezmienniczości długości wektora podczas transformacji współrzędnych wektora w układzie współrzędnych ortonormalnych. W tym celu trzeba wyliczyć $f'(x, y)$, aby określić, jakie błędy wprowadziła kwantyzacja. Po wyliczeniu $f'(x, y)$ zaokrągliła się wszystkie wartości do liczb całkowitych, a potem znów wylicza się DCT, ale tym razem w wersji ortonormalnej. Transformaty mogą być traktowane jako metoda wyznaczania współrzędnych w nowym układzie współrzędnych, czyli jako transformacje między układami współrzędnych. W przypadku transformaty DCT jest to przeliczenie współrzędnych bloku 8×8 do układu współrzędnych, w których osiami są dwuwymiarowe funkcje kosinus.

W przypadku obliczeń z udziałem odchylenia standardowego konieczne jest przeliczenie założonego błędu do nowego układu współrzędnych wyznaczonego przez dyskretną transformatę cosinusów. Niestety, jeżeli błąd jest podany w postaci odchylenia standardowego, to nie jest możliwe wykorzystanie go w operacjach w przestrzeni wektorowej. Nie może on być częścią wektora ani nie może być skalarem, gdyż nie może przyjmować wartości ujemnych. Zamiast odchylenia standardowego konieczne jest zatem posługiwanie się przyrostem odchylenia standardowego, który spełnia formalne warunki narzucone przez aksjomaty przestrzeni wektorowej. Przyrost odchylenia standardowego od odchylenia standardowego różni się tylko tym, że może przyjmować wartości ujemne.

Przyrost odchylenia standardowego musi być w określony sposób odwzorowany w obliczeniach w przestrzeni wektorowej. W kompresji JPG blok 8×8 stanowi wektor w przestrzeni 64-wymiarowej. Do tego wektora, jako dodatkowy element, można dodać przyrost odchylenia standardowego. Powstanie wówczas wektor:

$$\vec{f} = [f(0,1), f(0,2), \dots, f(7,6), f(7,7), \Delta\sigma_{\epsilon}]. \quad (11)$$

Ostatnim elementem tego wektora jest przyrost odchylenia standardowego $\Delta\sigma_\varepsilon$ mówiący o maksymalnym dopuszczalnym dla danego bloku poziomie odchylenia standardowego. Wyliczenie DCT dla tego wektora jest wyznaczeniem jego nowych współrzędnych w układzie współrzędnych zdefiniowanych przez ortogonalną bazę funkcji cosinus. Układ ten jest ortogonalny, ale nie ortonormalny. Oznacza to, że wektory osi nie są wektorami jednostkowymi. W układach współrzędnych ortonormalnych wartość $\Delta\sigma_\varepsilon$ jest zawsze taka sama, gdyż można ją interpretować jako promień hiperkuli, określającej miejsce w przestrzeni, gdzie mieszczą się końce pewnej liczby wektorów określonych wzorem (11). Liczba ta zależy od przyjętego rozkładu prawdopodobieństwa. Hiperkula jest figurą, której rzuty na dowolną oś w każdym ortonormalnym układzie współrzędnych są zawsze równe podwojonej długości promienia. W układach ortogonalnych nieortonormalnych, wartości jej rzutów mogą ulec zmianie, a jeżeli skala nie zostanie zachowana na osiach, to nie będą one takie same dla wszystkich osi.

Transformata DCT dana wzorem (2) definiuje przekształcenie do układu ortogonalnego, ale nie ortonormalnego. Można jednak ten wzór odpowiednio zmodyfikować, aby uzyskać przekształcenie do układu ortonormalnego:

$$F(u, v) = \frac{1}{8} C'(u) C'(v) \left[\sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right], \quad (12)$$

przy czym:

$$C'(w) = \begin{cases} \sqrt{2} & \text{dla } w = 0 \\ 1 & \text{dla } w \neq 0. \end{cases} \quad (13)$$

Transformując wektor dany wzorem (11) za pomocą wzoru (12), otrzyma się wektor:

$$\vec{F}_{\text{ort}} = [F_{\text{ort}}(0,1), F_{\text{ort}}(0,2), \dots, F_{\text{ort}}(7,6), F_{\text{ort}}(7,7), \Delta\sigma_\varepsilon]. \quad (14)$$

Ze względu na ortonormalność układu współrzędnych wartość $\Delta\sigma_\varepsilon$ nie ulegnie zmianie. Współrzędne powstałego wektora muszą zostać „poprzestawiane” zgodnie z zasadą przedstawioną na rysunku 2. Otrzyma się w ten sposób wektor:

$$\vec{F}_{\text{ort}} = [F_{\text{ort}}(0), F_{\text{ort}}(1), \dots, F_{\text{ort}}(63), F_{\text{ort}}(64), \Delta\sigma_\varepsilon]. \quad (15)$$

Każdy wektor można podzielić na dwa wektory: wektor A_{sr} reprezentujący poziom wartości średniej obrazu i wektor A_{zm} reprezentujący zmienność obrazu, czyli wahania jasności pikseli wokół wartości średniej. W dalszych obliczeniach należy skorzystać ze związku pomiędzy długością wektora A_{zm} a odchyleniem standardowym jego współrzędnych:

$$\sigma_{\vec{A}_{zm}} = \sqrt{\frac{\sum_{k=1}^N (A_{zm}(k) - 0)^2}{N-1}} = \sqrt{\frac{\sum_{k=1}^N A_{zm}^2(k)}{N-1}}, \quad (16)$$

gdzie: $A_{zm}(k)$ — współrzędne wektora A_{zm} ;

N — wymiar przestrzeni,

przy czym długość wektora A_{zm} wynosi:

$$\|\vec{A}_{zm}\| = \sqrt{\sum_{k=1}^N A_{zm}^2(k)}, \quad (17)$$

zatem:

$$\sigma_{\vec{A}_{zm}} = \frac{\|\vec{A}_{zm}\|}{\sqrt{N-1}}. \quad (18)$$

Dla wektora A_{zm} niezmiennikiem transformacji współrzędnych jest odchylenie standardowe jego współrzędnych. W związku z tym można je wyznaczyć ze wzoru:

$$\sigma_{\vec{F}_{ort}} = \sqrt{\frac{\sum_{k=2}^N F_{ort}(k)}{N-1}}. \quad (19)$$

Aby obliczyć wartość odchylenia standardowego dla współczynników wektora \vec{F}_{ort} , nie trzeba zatem wyliczać odwrotnej transformaty cosinusów.

Mając wektor \vec{F}_{ort} niepoddany procesowi kwantyzacji oraz wektor \vec{F}'_{ort} poddany procesowi kwantyzacji i przeskalowany do poziomu początkowych wartości, a także znając wartość $\Delta\sigma_\varepsilon$, można zwiększyć stopień kompresji, zerując współczynniki wektorów \vec{F}'_{ort} i \vec{F}^Q związane z wysokimi wartościami. Oba wektory są ze sobą związane. Wektor \vec{F}^Q można otrzymać, mnożąc współrzędne wektora \vec{F}'_{ort} przez odpowiednie współczynniki. Zależność ta nie ulegnie zmianie, jeżeli przypisze się wartość zero odpowiadającym sobie współrzędnym obu wektorów. Dzięki temu sprawdzenie warunków przypisywania zer współrzędnym można wykonać dla wektora \vec{F}'_{ort} , a po wykonaniu przypisania zer wektora \vec{F}^Q użyć w dalszym procesie kompresji. Ze względu na zaokrąglenia podczas obliczeń nie gwarantuje to, że wektor \vec{F}^Q nie wprowadzi błędu większego niż założony, jednak przekroczenie założonego progu może być tylko nieznaczące.

Proces przypisywania zer współrzędnym można opisać następującym algorytmem:

1. Przyjęcie jako początkowego numeru współrzędnej k wartości 64.
2. Wyznaczenie wartości odchylenia standardowego $\sigma_{\Delta\vec{F}_{ort}}$ wektora $\Delta\vec{F}_{ort}$ będącego różnicą pomiędzy wektorem \vec{F}_{ort} a \vec{F}'_{ort} .
3. Jeżeli wartość $\sigma_{\Delta\vec{F}_{ort}}$ jest większa od $\Delta\sigma_\varepsilon$, należy zakończyć algorytm.

4. Przypisanie zera współrzędnym wektorów \vec{F}'_{ort} i \vec{F}^{O} o numerze k .
5. Zmniejszenie k o jeden.
6. Skok do punktu 2.

4. Kompresja z maksymalnym założonym błędem

Oprócz odchylenia standardowego kolejnym parametrem, którym można sterować kompresją, jest maksymalna różnica pomiędzy obrazem skompresowanym $f''(x, y)$ i nieskompresowanym $f(x, y)$:

$$\varepsilon_{\Delta} = \max |f(x, y) - f''(x, y)|. \quad (20)$$

Różnica powinna być mniejsza od zadanej wartości błędu ε :

$$\varepsilon_{\Delta} \leq \varepsilon. \quad (21)$$

Różnica ta jest zawsze dodatnia, ale na potrzeby rachunku w przestrzeni wektorowej należy dopuścić jej ujemne wartości. Analogicznie jak dla odchylenia standardowego, można zdefiniować wektor:

$$\vec{f} = [f(0,1), f(0,2), \dots, f(7,6), f(7,7), \Delta\varepsilon]. \quad (22)$$

Parametr $\Delta\varepsilon$ jest przyrostem maksymalnej dopuszczalnej różnicy pomiędzy aktualną wartością piksela a wartością w obrazie skompresowanym. Posiada inną interpretację niż $\Delta\sigma_{\varepsilon}$. Nie definiuje w przestrzeni hiperkuli, ale hipersześcian, którego bok ma wysokość $2\Delta\varepsilon$. Rzuty hipersześcianu na osie ortonormalnego układu współrzędnych, w odróżnieniu od hiperkuli, zależą od „ułożenia” układu względem hipersześcianu. Utrudnia to znacznie obliczenia. Dla uproszczenia można zastąpić hipersześcian hiperkulą, na której jest on opisany, przyjmując promień równy $\Delta\varepsilon$. Uproszczenie to powoduje, że współczynnik kompresji będzie mniejszy niż możliwy do osiągnięcia przy wykorzystaniu hipersześcianu, jednak uprości to znacznie proces obliczeniowy.

Korzystając ze wzoru (12) należy wyznaczyć wartości wektora \vec{F} , a następnie uporządkować go zgodnie z rysunkiem 2. Uzyskuje się w ten sposób wektor:

$$\vec{F} = [F(0), F(1), \dots, F(63), F(64), \Delta\varepsilon]. \quad (23)$$

Następnie należy przypisywać wartość zero współrzędnym wektora \vec{F} , počynając od ostatniej współrzędnej. Po każdym przypisaniu wartości zero należy

sprawdzić odległość \vec{F} początkowego od \vec{F} aktualnie zmienianego oznaczonego dalej jako \vec{F}' zgodnie ze wzorem:

$$d(\vec{F}, \vec{F}') = \sqrt{\sum_k^N [F(k) - F'(k)]^2}. \quad (24)$$

Odległość tę należy odnieść do $\Delta\epsilon$. Jeżeli jest ona większa od $\Delta\epsilon$, to znaczy, że błąd mógł przekroczyć dopuszczalną wartość. Jak wspomniano, przyjęte uproszczenie, polegające na zastąpieniu hipersześcianu przez hiperkulę, powoduje, że nie osiąga się możliwego poziomu kompresji. Aby go zwiększyć, po przypisaniu zera składowym wysokoczęstotliwościowych powyższą metodą, można po przekroczeniu przez odległość $d(\vec{F}, \vec{F}')$ wartości $\Delta\epsilon$ wykonać dodatkowe sprawdzenie. Sprawdzenie wykonuje się poprzez wyliczenie odwrotnej transformaty cosinusów i wyznaczenie błędu względem obrazu nieskompresowanego. Jest to operacja dużo bardziej czasochłonna, ale z reguły pozwalająca na przypisanie zera jeszcze paru dodatkowym składowym, co zwiększy poziom kompresji. Połączenie obu metod daje znaczne przyspieszenie obliczeń w porównaniu z użyciem tylko drugiej metody, gdyż spełnienie warunku:

$$d(\vec{F}, \vec{F}') \leq \Delta\epsilon - \epsilon_\Delta \quad (25)$$

automatycznie gwarantuje nieprzekroczenie założonego poziomu błędu, co pozwala uniknąć wyliczania odwrotnej transformaty cosinusów.

5. Przykład kompresji obrazu sonarowego

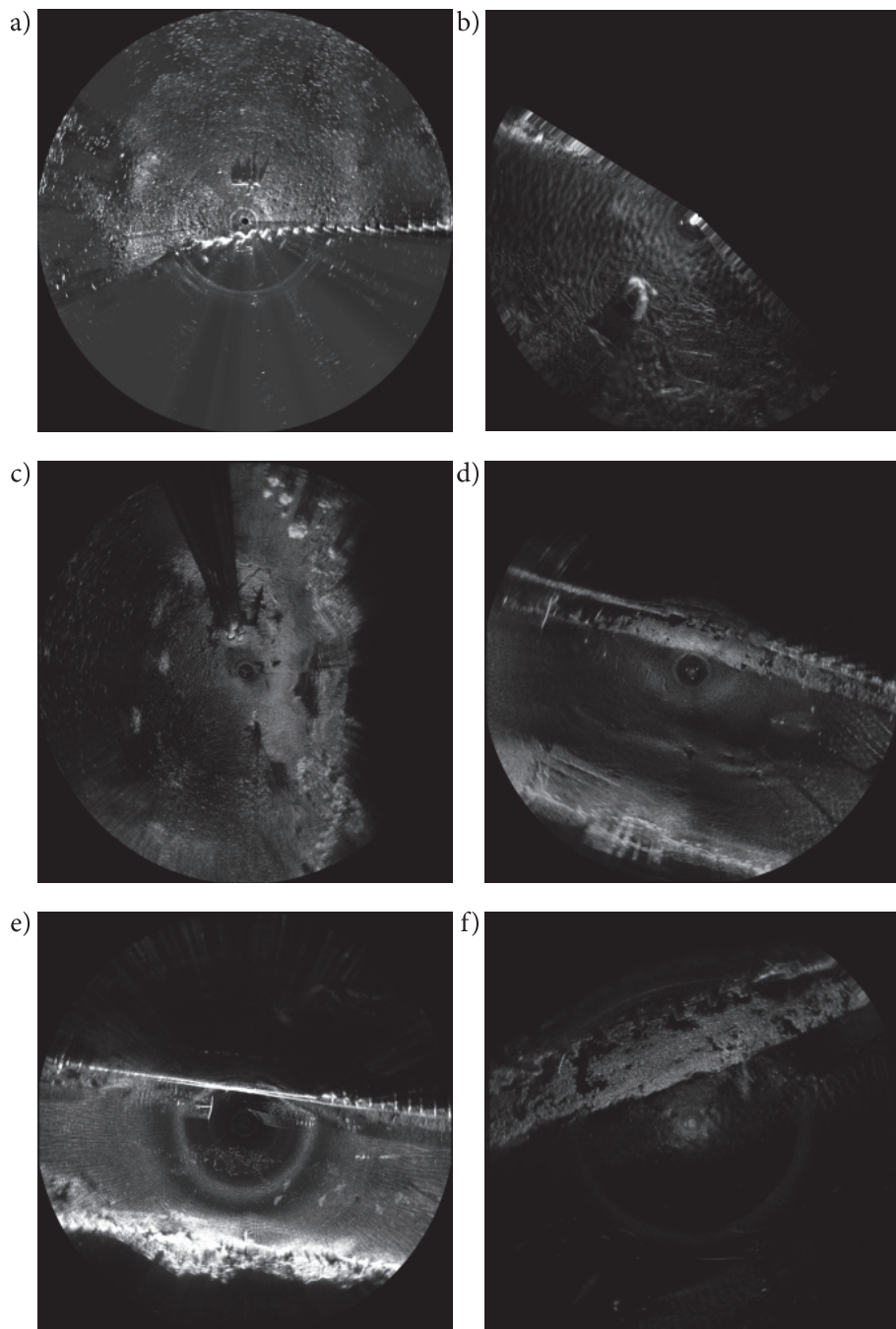
Prezentowane metody kompresji przetestowano na sześciu obrazach sonarowych. Obrazy te pochodzą z sonaru MS1000 pracującego w trybie sektorowym. Dane z sonaru zostały przetworzone do postaci mozaiki (rys. 3).

Do oszacowania błędów kompresji użyto współczynników:

1. PSNR (ang. *Peak Signal-to-Noise Ratio*) [1]:

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{x=1}^N \sum_{y=1}^M [g(x, y) - \hat{g}(x, y)]^2}, \quad (26)$$

gdzie: N, M — rozmiary obrazu;
 $g(x, y)$ — obraz oryginalny;
 $\hat{g}(x, y)$ — obraz po kompresji.



Rys. 3. Obrazy z sonaru sektorowego a, b) manekin; c) dalby; d, e, f) kanał (źródło: Hydrograf XXI, Akademia Morska w Szczecinie)

2. SSIM (ang. *Structural Similarity Index*) [20]:

$$\text{SSIM} = \frac{(2\mu_g \mu_{\hat{g}} + C_2)(2\sigma_{g\hat{g}} + C_2)}{(\mu_g^2 + \mu_{\hat{g}}^2 + C_1)(\sigma_g^2 + \sigma_{\hat{g}}^2 + C_2)}, \quad (27)$$

gdzie: $\mu_g, \mu_{\hat{g}}$ — wartości średnie obrazów przed kompresją i po kompresji;
 $\sigma_g, \sigma_{\hat{g}}$ — odchylenia standardowe obrazów przed kompresją i po kompresji;
 $\sigma_{g\hat{g}}$ — kowariancja pomiędzy obrazem przed kompresją i po kompresji.

4. NLSE (ang. *Normalized Least-Squares Error*) [11]:

$$\text{NLSE} = \frac{\sum_{x=1}^N \sum_{y=1}^M [g(x, y) - \hat{g}(x, y)]^2}{\sum_{x=1}^N \sum_{y=1}^M [g(x, y)]^2}, \quad (28)$$

przy czym będzie on użyty w formie logarytmicznej [11]:

$$\text{NLSE}_{\log} = -10 \log_{10} \text{NLSE}. \quad (29)$$

W pierwszym eksperymencie kompresowano obrazy z rysunku 3 z założoną maksymalną wartością odchylenia standardowego pomiędzy obrazem nieskompresowanym a skompresowanym równą 5. Tabela 1 przedstawia uzyskane wyniki dla proponowanej metody kompresji (oznaczonej przez Jpg*), kompresji Jpg, Jpg2000 oraz fraktalnej. Do kompresji Jpg* i Jpg użyto oprogramowania wykonanego w języku Matlab. Do kompresji Jpg2000 wykorzystano przeglądarkę zdjęć FastStone Image Viewer 4.4, natomiast do kompresji fraktalnej programu FracTalCompressions. Z trzech dostępnych w programie metod kompresji wykorzystano metodę Jacquins. Jest to jedyna metoda pozwalająca skompresować największy obrazek w czasie poniżej siedmiu dni. Współczynniki kompresji dla wszystkich metod zostały tak dobrane, aby uzyskać rozmiary obrazów zbliżone do kompresji Jpg*. Z czterech porównywanych metod kompresji najwyższymi wartościami współczynników PSNR, SSIM i NLSE_{\log} charakteryzuje się kompresja Jpg2000. Oznacza to, że błąd rekonstrukcji danych dla kompresji Jpg2000 jest najmniejszy. Dodatkowo, oprócz trzech wymienionych współczynników, użyto również współczynnika σ_{\max} . Wartość σ_{\max} jest maksymalną wartością odchylen standardowych dla bloków 8×8 różnicy pomiędzy obrazem skompresowanym a nieskompresowanym. Uzyskane wartości σ_{\max} dla kompresji Jpg2000 wskazują, że nie może ona zagwarantować określonej wartości tego współczynnika. Waha się on od wartości 1,3 aż do 14,3, a dla obrazu c przekracza prawie trzykrotnie założony poziom.

TABELA 1

Wartości maksymalnego odchylenia standardowego odniesione do współczynników szacujących błąd kompresji dla obrazów z rysunku 3

Obraz	Rozmiar obrazu	Kompresja	Rozmiar obrazu po kompresji	PSNR	SSIM	NLSE _{log}	σ_{\max}
a	9000000	Jpg*	1217087	39,04	0,9953	22,81	5,1754
		Jpg	1211215	38,70	0,9949	22,47	27,9011
		Jpg2000	1212522	51,97	0,9998	35,88	1,7348
		Fraktalna	1342740	19,75	0,7945	3,66	69,7464
b	2250000	Jpg*	381914	39,65	0,9870	16,90	5,1922
		Jpg	384695	38,37	0,9830	15,63	14,5704
		Jpg2000	380540	53,50	0,9995	30,91	1,3144
		Fraktalna	397423	29,70	0,7814	7,11	79,7129
c	36000000	Jpg*	4085238	39,98	0,9926	20,08	5,1777
		Jpg	3997215	38,85	0,9908	18,96	28,6907
		Jpg2000	4072451	53,52	0,9997	33,80	14,3537
		Fraktalna	4184536	25,60	0,8072	5,88	59,19
d	9000000	Jpg*	1612903	38,87	0,9924	19,64	5,1653
		Jpg	1623790	37,37	0,9895	18,14	16,6468
		Jpg2000	1610826	51,02	0,9996	31,93	1,6545
		Fraktalna	1718196	25,33	0,7987	6,25	53,28
e	16000000	Jpg*	2763227	38,71	0,9983	26,37	5,2992
		Jpg	2775595	38,32	0,9982	25,97	22,0370
		Jpg2000	2767362	51,72	0,9999	39,45	7,9214
		Fraktalna	3135065	18,70	0,8045	6,42	77,49
f	2250000	Jpg*	204621	41,61	0,9851	16,83	5,1824
		Jpg	205608	41,91	0,9863	17,14	13,3463
		Jpg2000	204516	52,76	0,9991	28,21	1,7567
		Fraktalna	192811	32,44	0,8136	7,88	36,8945

Wartości PSNR, SSIM i $NLSE_{\log}$ dla Jpg* i Jpg są bardzo zbliżone. Wynika to z faktu, że obie metody kompresji są do siebie bardzo zbliżone. Różnica pojawia się jedynie w wartości współczynnika σ_{\max} , która dla Jpg* jest bliska wartości 5, a dla Jpg jest od dwóch do prawie trzech razy wyższa. Nieznaczne przekroczenie przez kompresję Jpg* wartości założonej wynika z błędów zaokrągleń. Najniższe wartości współczynników PSNR, SSIM i $NLSE_{\log}$ oraz najwyższe σ_{\max} są obserwowane dla kompresji fraktalnej. Czas potrzebny na kompresję pojedynczych obrazów jest bardzo długi i sięga 5-6 dni dla obrazu o największych rozmiarach.

W drugim eksperymencie kompresowano obrazy z rysunku 3 z założoną maksymalną różnicą pomiędzy obrazem nieskompresowanym a skompresowanym równą 10. W tabeli 2 przedstawiono uzyskane wyniki. Wartość ε_{\max} jest maksymalną różnicą pomiędzy wartościami obrazu skompresowanego i nieskompresowanego. Uzyskane wartości różnic mogą przekraczać założoną wartość na skutek błędów zaokrągleń. Dla kompresji Jpg2000 nie udało się uzyskać podobnych rozmiarów plików jak dla Jpg*, ze względu na ograniczenia wynikające z przyjętego w przeglądarce zdjęć FastStone Image Viewer 4.4 zakresu współczynnika kompresji. Wyniki uzyskano podobne jak w poprzednim eksperymencie. Najlepsze rezultaty uzyskano dla kompresji jpg2000 przy jeszcze większej rozpiętości współczynnika ε_{\max} (od 1 do 63). Kompresja fraktalna uzyskała najgorsze wyniki. Żadna z metod poza Jpg* nie zagwarantowała utrzymania wartości ε_{\max} na poziomie mniejszym albo zbliżonym do zadanego.

TABELA 2

Wartości ε_{\max} odniesione do współczynników szacujących błędy kompresji dla obrazów z rysunku 3

Obraz	Rozmiar obrazu	Kompresja	Rozmiar obrazu po kompresji	PSNR	SSIM	$NLSE_{\log}$	ε_{\max}
a	9000000	Jpg*	2956496	45,69	0,9989	29,45	10
		Jpg	2957320	44,67	0,9986	28,44	19
		Jpg2000	2025379	112,90	1,0000	96,81	1
		Fraktalna	2430853	19,81	0,8010	3,72	244
b	2250000	Jpg*	813688	46,46	0,9961	23,72	10
		Jpg	822501	46,06	0,9959	23,32	11
		Jpg2000	484642	104,66	1,0000	82,08	1
		Fraktalna	795534	30,08	0,7860	7,49	231
c	36000000	Jpg*	8841130	46,18	0,9977	26,29	11
		Jpg	8827356	44,26	0,9968	24,37	27
		Jpg2000	6798478	81,40	1,0000	61,68	63
		Fraktalna	9372028	25,70	0,8109	5,98	145

cd. tabeli 2

d	9000000	Jpg*	3711716	45,21	0,9978	25,97	11
		Jpg	3692777	44,12	0,9973	24,89	14
		Jpg2000	2464810	109,22	1,0000	90,13	1
		Fraktalna	3438230	25,48	0,8029	6,38	179
e	16000000	Jpg*	6346068	45,31	0,9995	32,96	11
		Jpg	6368275	43,92	0,9993	31,58	18
		Jpg2000	4295917	83,06	1,0000	70,78	22
		Fraktalna	6198780	18,70	0,8049	6,43	245
f	2250000	Jpg*	453864	47,14	0,9935	22,37	9
		Jpg	455329	45,63	0,9920	20,86	22
		Jpg2000	402802	108,64	1,0000	84,09	1
		Fraktalna	393488	33,19	0,8279	8,63	98

W trzecim eksperymencie kompresowano dane z założonym poziomem współczynnika PSNR (tab. 3). Kompresując obrazy starano się tak dobrać współczynniki kompresji, aby uzyskać możliwie podobne wartości PSNR. W badaniu pominięto kompresję fraktalną ze względu na zbyt długi czas kompresji (dla największego pliku prawie tydzień). Najmniejsze rozmiary plików uzyskano dla kompresji Jpg2000. Wartości σ_{\max} dla Jpg2000 i Jpg są mniej więcej dwa razy większe niż dla Jpg*.

TABELA 3

Wartości maksymalnego odchylenia standardowego odniesione do współczynników szacujących błędy kompresji dla obrazów z rysunku 3 przy stałym PSNR

Obraz	Rozmiar obrazu	Kompresja	Stosunek wielkości obrazu przed kompresją i po kompresji [%]	PSNR	SSIM	NLSE _{log}	σ_{\max}
a	9000000	Jpg*	15,66	39,99	0,9962	23,76	4,4468
		Jpg	16,73	40,06	0,9963	23,83	19,6214
		Jpg2000	2,40	39,98	0,9963	23,93	12,4315
b	2250000	Jpg*	18,19	39,96	0,9878	17,22	4,9230
		Jpg	20,58	40,03	0,9879	17,28	9,5191
		Jpg2000	3,30	40,03	0,9895	17,45	9,8355
c	36000000	Jpg*	11,34	39,98	0,9926	20,08	5,1777
		Jpg	12,85	40,04	0,9921	19,73	23,9111
		Jpg2000	1,40	39,84	0,9932	20,16	14,8866

cd. tabeli 3

d	9000000	Jpg*	21,94	40,01	0,9940	20,77	4,2725
		Jpg	26,14	39,94	0,9939	20,71	8,4825
		Jpg2000	4,00	40,02	0,9946	20,96	9,3383
e	16000000	Jpg*	21,05	40,01	0,9987	27,67	4,2686
		Jpg	23,09	39,94	0,9987	27,59	13,0259
		Jpg2000	3,20	39,90	0,9988	27,63	12,1688
f	2250000	Jpg*	7,53	40,02	0,9798	15,25	6,7506
		Jpg	7,20	39,97	0,9803	15,21	18,5725
		Jpg2000	0,70	39,85	0,9811	15,33	15,5894

6. Podsumowanie

W artykule zaprezentowano dwie metody kompresji obrazu sonarowego będące modyfikacją kompresji JPG. Pierwsza z nich to metoda z założoną wartością odchylenia standardowego różnicy obrazów skompresowanego i nieskompresowanego. Jej wadą jest nieznaczne, ale stałe przekraczanie założonej wartości ze względu na błędy zaokrągleń. Druga metoda zapewnia eliminację tego błędu. Najważniejszą zaletą obu metod jest jednak niezmienną formą dekompresji obrazu. Wyniki dla zaproponowanej metody kompresji są zbliżone do klasycznej kompresji Jpg. Jedyną różnicą jest możliwość utrzymania zadanego poziomu σ_{\max} lub ε_{\max} , co w przypadku kompresji Jpg nie jest możliwe do osiągnięcia. Kompresja fraktalna nie może być aktualnie wykorzystywana do kompresji obrazu sonarowego ze względu na bardzo długi czas obliczeń (dla największego obrazu sięgający prawie tydzień). Badana metoda kompresji fraktalnej charakteryzowała się też najmniejszymi wartościami współczynników PSNR, SSIM i $NLSE_{\log}$. Być może inne metody kompresji fraktalnej osiągnęłyby lepsze rezultaty, jednak długi czas obliczeń uniemożliwił wzięcie ich pod uwagę podczas badań. Najlepsze wyniki osiągnięto dla kompresji Jpg2000, jednak podobnie jak kompresja Jpg nie gwarantuje ona zachowania założonych wartości σ_{\max} lub ε_{\max} .

Przewagą proponowanej metody nad innymi metodami kompresji stratnej jest możliwość przyjęcia wartości σ_{\max} lub ε_{\max} poniżej poziomu szumu występującego na obrazie sonarowym. Dzięki temu kompresja będzie wprowadzać zmiany, które nie spowodują utraty informacji w obrazie. Umożliwia to zatem jej zamienne stosowanie z metodami kompresji bezstratnej. Dla obrazu sonarowego ma to istotne znaczenie ze względu na jego duże rozmiary i w wielu przypadkach bardzo wysokie koszty pozyskania danych, które wykluczają wszelkie metody kompresji powodujące utratę informacji.

Artykuł wpłynął do redakcji 5.07.2010 r. Zweryfikowaną wersję po recenzji otrzymano w czerwcu 2010 r.

LITERATURA

- [1] I. BRAILOVSKIY, D. PLOTKIN, *Modified JPEG algorithm with Binary Interval Transform coding with improved compression ratio*, Proceedings GraphiCon, Moscow State University, June 23-27, 2008.
- [2] A. C. BOVIK (red.), *Handbook of Image and Video Processing*, Elsevier, San Diego, 2000.
- [3] K. CABEEN, P. GENT, *Image Compression and the Discrete Cosine Transform*, <http://online.redwoods.cc.ca.us/instruct/darnold/laproj/fall98/pken/dct.pdf> [20.05.2010].
- [4] S. CHANDRA, C. EBLIS, *JPEG compression metric as a quality aware image transcoding*, Proceedings of USITS' 99: The 2nd USENIX Symposium on Internet Technologies & Systems, Boulder, Colorado, USA, October 11-14, 1999.
- [5] A. CHYBICKI, M. KULAWIAK, Ł. KAMIŃSKI, Z. ŁUBNIEWSKI, J. DĄBROWSKI, *Algorytmy analizy, przetwarzania i wizualizacji danych z sonaru wielowiązkowego w rozproszonych systemach GIS*, Polish Hyperbaric Research, 1, 26, 2009, 19.
- [6] A. CHYBICKI, M. MOSZYŃSKI, P. POCWIARDOWSKI, *Applications of compression techniques for reducing the size of multibeam sonar records*, 1st International Conference on Information Technology, Gdańsk, May 18-21, 2008.
- [7] A. CHYBICKI, M. MOSZYŃSKI, *Metody przetwarzania danych pochodzących z sonarów wielowiązkowych dla potrzeb kompresji*, Zeszyty Naukowe Wydziału ETI Politechniki Gdańskiej, *Technologie Informacyjne*, T. 15, 2008, 207-212.
- [8] *Class TIFFImageWriteParam*, http://java.sun.com/products/java-media/jai/forDevelopers/jai-imageio-1_0_01-fcs-docs/com/sun/media/imageio/plugins/tiff/TIFFImageWriteParam.html [20.05.2010].
- [9] J. IMPAGLIAZZO, W. GREENE, Q. HUYNH, *Wavelet image compression algorithm for side-scan sonar and teleradiology*, Proceedings Spie, 2491, 162, 1995.
- [10] JPEG File Interchange Format, Version 1.02, <http://www.jpeg.org/public/jfif.pdf> [20.05.2010].
- [11] K. W. PRATT, *Digital image processing*, A Wiley-Interscience Publication, New York, 2001.
- [12] M. RAMOS, S. HEMAMI, *Edge-adaptive JPEG image compression*, Proceedings Spie, vol. 2727, Symposium on Visual Communications & Image Processing, 1996.
- [13] R. ROSENHOLTZ, A. WATSON, *Perceptual adaptive JPEG coding*, Proceedings, IEEE International Conference on Image Processing, Lausanne, Switzerland, IEEE, 11996, 901-904.
- [14] M. SMAYRA, *A sonar imaging system*, Dysertacja doktorska, University of British Columbia, 1992.
- [15] A. STĄCZNY (red.), *Metody nawigacji porównawczej*, Gdańskie Towarzystwo Naukowe, Gdańsk, 2004.
- [16] A. STĄCZNY, *Problems of comparative plotting of the ship's position*, Reports on Geodesy, 1, 61, 2002.
- [17] *TIFF Revision 6.0 Final* <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf> [20.05.2010].
- [18] G. WALLACE, *The JPEG still picture compression standard*, *Communications of the ACM archive*, 34, 4, 1991, 30-44.
- [19] A. B. WATSON, *Image Compression Using the Discrete Cosine Transform*, *Mathematica Journal*, 4, 1, 1994, 81-88.
- [20] Z. WANG, A. C. BOVIK, H. R. SHEIKH, E. P. SIMONCELLI, *Image Quality Assessment: From Error Visibility to Structural Similarity*, *IEEE Transactions on Image Processing*, 13, 4, 2004, 600-612.

- [21] A. B. WATSON, *Visual optimization of DCT quantization matrices for individual images*, Proceedings AIAA Computing in Aerospace 9, San Diego, CA, American Institute of Aeronautics and Astronautics, 1993, 286-291.

M. BORAWSKI

JPG compression of sonar image with taking into account an assumed level of error

Abstract. In the article, two types of sonar data compression were presented. They both use the JPG algorithm. Thanks to them, it is possible to obtain the standard deviation for blocks of an image, which is a result of compressed and uncompressed image subtraction. They let to stay in the range of assumed compression error level as well. Instead of DCT transform, used in JPG compression, its orthonormal version was used. It allows us to use the properties of invariance of vector length in orthogonal coordinate system. Thanks to that, to determine the error of compression, calculation of reverse DCT transform is not needed.

Presented methods change only the way of compression and not the decompression itself. It allows us to use any decompressing software with images compressed by these methods. For sonar images it is very important, because of the possibility of using the JPG compression in GeoTiff standard, allowing us to memorize the information about geographic coordinates and orientation in relation to the north direction of sonar image.

Presented methods were compared with Jpg, Jpg2000 and fractal compression for chosen sonar images. The coefficients PSNR and SSIM were used. By the same size of files after the compression, the presented methods give lower quality of image, however, only they allow us to obtain the maximum value of error which is close to predicted one. The excess of this error is a result of rounding errors. For Jpg, Jpg2000 and fractal compression, the maximum error was exceeded several times.

Keywords: pattern recognition sonar, sonar image, JPG compression, GeoTiff

