

**Piotr Radosław Sawicki**  
**Sem. 9**  
**Grupa: T1IN-IN**

**Warszawa 14-01-2004r**

**Projekt z Kompresji Danych**  
**Temat:**  
**Koder Falkowy z Interfejsem**

## **I) Cel projektu:**

Tematem projektu jest opracowanie programu z graficznym interfejsem użytkownika (ang. GUI), umożliwiającego przeprowadzanie kompresji/dekompresji obrazów w standardzie JPEG2000. Interfejs ma umożliwiać wygodne wprowadzanie parametrów kodeka, jak również musi umożliwiać prezentację efektów jego działania.

## **II) Wstęp:**

JPEG2000 jest nowym standardem kompresji obrazów, opartym o transformatę falkową. W odróżnieniu od dotychczasowych metod zapisu obrazów (np.: jpeg), nowy standard jest bardziej nastawiony na zastosowania multimedialne, a więc tam gdzie stosuje się strumieniowy przepływ informacji (np.: Internet). O ile w wypadku dotychczasowych metod kompresji/dekompresji obrazów, nie mieliśmy możliwości sterowania procesem dekodowania informacji, o tyle nowy standard umożliwia nam sterowanie procesem rekonstrukcji obrazu, a przez to, znaczne zmniejszenie przepływu danych (co jest bardzo ważne w zastosowaniach sieciowych, gdzie każdy bajt przesłanej informacji jest cenny, ze względu na ograniczoną przepustowość łącz). Zorientowanie na zastosowania multimedialne przejawia się także w możliwości zastosowania regionów zainteresowania – ROI. Dzięki temu, odbiorca zdekodowanej informacji, może otrzymać to co go naprawdę interesuje (zamiast całego obrazu wysokiej jakości, może pobrać obraz o niskiej jakości z wyszczególnionym obszarem ROI o wysokiej jakości). Inną rzeczą godną uwagi jest możliwość skalowania rekonstruowanego obrazu, jak również określanie jego jakości. Dodatkową zaletą jest mała wrażliwość na błędy, co jest atutem w przypadku rozwoju urządzeń mobilnych, ale również ogólnie przy przesyłaniu strumienia kodowego kanałem gdzie występują duże zakłócenia. Mała wrażliwość na błędy ma taką przewagę nad dotychczasowymi metodami kompresji obrazów, że zakłócenie nieznacznie wpływa na odtwarzany obraz, i nie jest konieczne powtórzenie procesu transmisji (w przypadku np.: standardu JPEG, małe przekłamanie powodują zazwyczaj to, że odtworzony obraz jest „nieczytelny”).

### III) Schemat kodera falkowego:

#### a) Kompresja

Na kompresję w standardzie JPEG200 składają się następujące kroki:

- 1) Formowanie danych  
Rozmieszczenie komponentów (np.: dla składowych R,G,B mamy trzy oddzielne komponenty) na siatce odniesienia i podzielenie ich na części (tile). Każda z części jest w następnych krokach przetwarzana niezależnie.
- 2) Transformacja kolorów  
Są stosowane gdy obraz zawiera co najmniej trzy komponenty. Dokonuje się transformacji YCbCr w wersji odwracalnej i nieodwracalnej.
- 3) Zastosowanie transformaty falkowej  
Stosuje się tu dwa rodzaje filtrów:
  - Bank filtrów 9/7 (dolnoprzepustowy filtr analizy rzędu 9/górnoprzepustowy rzędu 7) – realizuje transformację rzeczywistoliczbową (real)
  - Bank filtrów 5/3 – realizuje transformację całkowitoliczbową (int)
- 4) Kwantyzacja  
Stosuje się tu kwantyzator równomierny DUTQ
- 5) Organizacja kodu , kodowanie arytmetyczne  
Bloki zostają podzielone na dzielnice (precinct), natomiast te na bloki kodowe (code block).  
Bloki kodowe zawierają płaszczyzny bitowe (bit plane), które są niezależnie kodowane koderem arytmetycznym.  
W ten sposób otrzymujemy pakiety.  
Pakiety są pogrupowane pod względem jakości. Pakiety o jednakowej jakości są grupowane w warstwy (layer).  
Zbiór wszystkich pakietów, uzupełnia się nagłówkiem zawierającym informacje o obrazie i parametrach kompresji tworząc w ten sposób strumień kodowy (code stream).

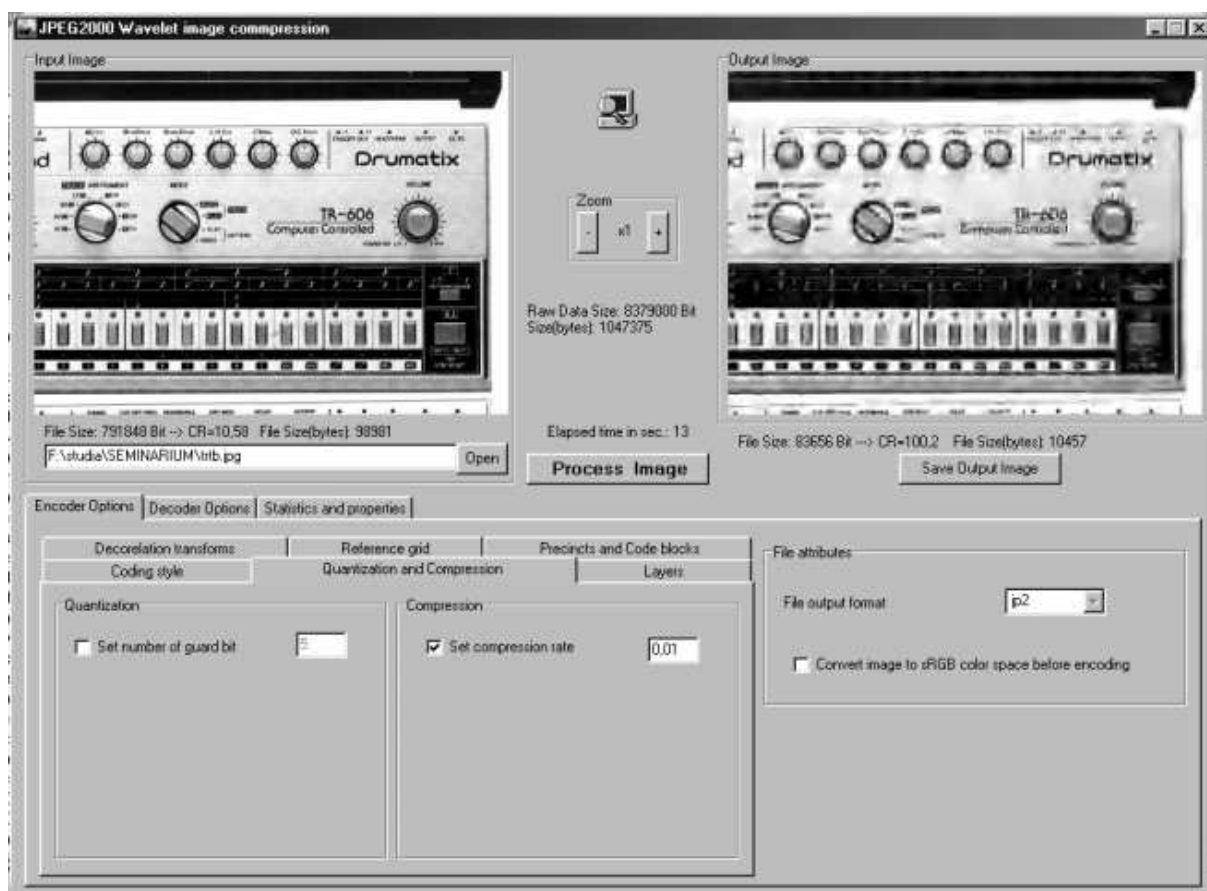
#### b) Dekompresja

- 1) Reorganizacja kodu
- 2) Dekodowanie arytmetyczne
- 3) Dekwantyzacja
- 4) Zastosowanie odwrotnej transformaty falkowej
- 5) Zastosowanie odwrotnej transformaty kolorów
- 6) Odtworzenie obrazu na podstawie zdekodowanych danych

#### IV) Opis implementacji:

Aplikacja została napisana w języku C++ w środowisku Borland C++ Builder.

W projekcie wykorzystano gotową bibliotekę JasPer, a dokładniej uruchamialny z wiersza poleceń program, który eksploatuje wszystkie możliwości tejże biblioteki. Aplikacja została stworzona jako nakładka na istniejący program uruchamiany z linii poleceń. Dzięki temu użytkownik może korzystać w dwojaki sposób z programu jasper.exe. Dodatkowo, w celu poszerzenia gamy formatów wejściowych aplikacji, zastosowano program do konwersji grafiki (topng).



Ryc1. Widok ekranu roboczej aplikacji.

Na obrazku Ryc1 przedstawiony został ekran roboczy aplikacji. Jak widzimy na panelu roboczym, można wyszczególnić dwa główne obszary, górny i dolny. Górny służy do wizualizacji (możliwość subiektywnego określenia jakości kompresji), natomiast dolny do konfiguracji, oraz ilościowego określenia jakości kompresji przy zadanych parametrach kodeka. Górną część można podzielić również na dwie części. Lewą związaną z obrazem wejściowym, oraz prawą związaną z obrazem zakodowanym.

Przycisk „Open” służy do wczytania obrazka wejściowego. Format obrazka możemy wybrać z następującej listy:

*"bmp", "jpg", "jif", "jpeg", "png", "ico", "jng", "koa", "iff", "lbm", "mng", "pbm", "pcd", "pcx", "pgm", "ppm", "ras", "tga", "targa", "tif", "tiff", "wbmp", "wap", "psd", "cut", "jp2", "jpc"*

Po wybraniu i zatwierdzeniu obrazka, zostanie on wyświetlony w lewym oknie podglądu. Przytrzymując lewy przycisk myszy na obrazie w tym okienku i przesuwając myszkę, możemy obejrzeć poszczególne fragmenty obrazu, bez konieczności kłopotliwego wyświetlania obrazu w całości. Przyciski + i - służą do powiększania i pomniejszania zawartości obu okien podglądu. Dzięki tym dwóm rozwiązaniom, możemy dokładnie przyjrzeć się szczegółom obrazka wejściowego jak i wyjściowego. Jeśli nie wystarcza nam taki tryb pracy, możemy obejrzeć obraz o rzeczywistych rozmiarach. Aby to uczynić, należy szybko, dwukrotnie kliknąć lewym przyciskiem myszy na obrazku. Otworzy się wówczas oddzielne okienko zawierające obraz. W zależności od tego, które okno podglądu klikniemy, ukazać się nam okna z obrazem wejściowym bądź wyjściowym.

Dolny panel jest podzielony na trzy główne zakładki:

- a) **Encoder Options** – opcje kodera
- b) **Decoder Options** – opcje dekodera
- c) **Statistics and properties** – Statystyki i parametry

Zakładka **“Encoder Options”** zawiera inne podpaneje, które grupują opcje **kodera**. Są to odpowiednio:

- a) Coding style – Określa styl kodowania, oraz dodatkowe opcje znaczące przy dekodowaniu a które mają zapewnić poprawne zdekodowanie w wypadku błędów.
  - Enable lazy coding mode – Dodanie bitów niosących informację bez żadnego kodowania.
  - Use vertical stripe causal context – Podczas kodowania użyty będzie kontekst sąsiednich pionowych linii w kodowanych blokach. Opcja jest przydatna podczas równoległego dekodowania i może przyczynić się do mniejszego zużycia pamięci przez dekodera.
  - Reset the probability models after each pass – Chodzi tu o to, by każdy blok kodować nowym modelem po każdej iteracji kodera.
  - Generate SOP marker segment – Określa czy dodawać marker startu pakietu .
  - Generate EPH marker segment – Określa czy dodawać marker końca pakietu.
  - Terminall all coding passes – Chodzi tu o to, by zatrzymywać i inicjować od początku koder arytmetyczny, bez niszczenia modelu probabilistycznego kontekstu.
  - Use segmentation symbols – Dodaje dodatkowy znacznik po każdej iteracji kodera.
  - Use predictable termination - Dzięki użyciu tej opcji dekodera potrafi rozpoznać na podstawie zaproponowanej metody przewidywania kodera, czy nie nastąpił błąd.
- b) Quantization and Compression
  - Set number of guard bits – określa liczbę tzw. Bitów bezpieczeństwa.
  - Set compresion rate – określamy stopień kompresji

c) Layers

- Use extra layers – użycie dodatkowych płaszczyzn. Możemy określić liczbę płaszczyzn, oraz wartość stopnia kompresji warstwy pierwszej. Stopień kompresji warstwy ostatniej jest określony w zakładce Quantization and Compression. Należy pamiętać, by wartości stopnia kompresji były szeregiem monotonicznym. W przypadku prezentowanej aplikacji, jest to postęp arytmetyczny.
- Set progresion order – możemy wybrać kolejność progresji.

d) Decorelation transforms

- Coding mode – możemy wybrać rodzaj filtra stosowanego przy transformacie falkowej. W wypadku wybrania int (kernel 5/3) mamy do czynienia z transformatą bezstratną, natomiast real (kernel 9/7) jest już transformatą stratną.
- Disallow the use of any multicomponent transform – Wymusza nieużywanie transformaty dla wielu komponentów.
- Set resolution level – Ustawiamy ilość poziomów.

e) Reference grid

- Set coordinates – ustawiamy współrzędne górnego-lewego rogu obrazu na siatce odniesienia.
- Use tiles – Użycie części. Możemy ustawić współrzędne lewego-górnego rogu takiej części(tile), oraz nominalną wysokość i szerokość.

f) Precincts and Code blocks

- Use precincts - Ustawianie parametrów dzielnic.
- Set code blocks - Ustawianie parametrów bloków kodowych.

Dodatkowo występują opcje związane z plikiem, takie jak format pliku wyjściowego (.jpc, jp2), oraz opcja, czy przestrzeń barw ma być konwertowana na sRGB przed kompresją.

W opcjach **dekodera** możemy wybrać ilość odczytywanej z pliku informacji. Może być ona podana w ilości płaszczyzn (number of layers) lub ilości pakietów (number of packets).

Gdy mamy ustawione opcje kodera, i chcemy obejrzeć rezultat kompresji, wciskamy przycisk „**Process Image**”. W wyniku w prawym oknie podglądu powinien pojawić się obraz będący wynikiem kompresji.

Przycisk „**Save Output Image**” umożliwia nam zapis obrazka w dwóch formatach, bitmapy .bmp, oraz formatach: .jp2 lub .jpc w zależności od tego, jaką opcję wybraliśmy w „**Encoder Options**”.

U dołu każdego z obrazków możemy odczytać następujące informacje:

- Rozmiar obrazka w bitach
- Rozmiar obrazka w bajtach
- CR (definiowany jako stosunek rozmiaru obrazu w postaci mapy bitowej do rozmiaru pliku związanego z danym obrazem)

W zakładce „**Statistics and properties**” możemy odczytać jakie są rozmiary obrazka, ile przypada bitów na zakodowanie pojedynczego piksela, oraz rozmiar obrazu w postaci mapy bitowej, czyli de facto dane dotyczące obrazu w oryginalnej postaci. Przycisk „**Compute**” służy do obliczania statystyk na podstawie obrazka wejściowego i wyjściowego. Informacje jakie uzyskujemy to:

- Średnia bitowa dla obrazka wejściowego i wyjściowego (bpp bit per pixel)
- CR dla obrazków wejściowego i wyjściowego
- CR(input/output) – stopień kompresji liczony jako stosunek rozmiarów plików obrazka wyjściowego do wejściowego.
- Maksymalny błąd bezwzględny (MaxAE)
- Błąd średnio-kwadratowy (MSE)
- Pierwiastek błędu średnio-kwadratowego (RMSE)
- Średni błąd bezwzględny (MAE)

Po przeprowadzeniu kompresji możemy zapisać skompresowany obraz w formacie .jpc bądź .jp2), wciskając przycisk „**Save Output Image**”. Dodatkowo możemy zapisać obraz w postaci bitmapy (.bmp), która jest produktem dekompresji przetwarzanego obrazu.

„Elapsed time” określa nam łączny czas kompresji, oraz dekompresji obrazu.

## V) Testy:

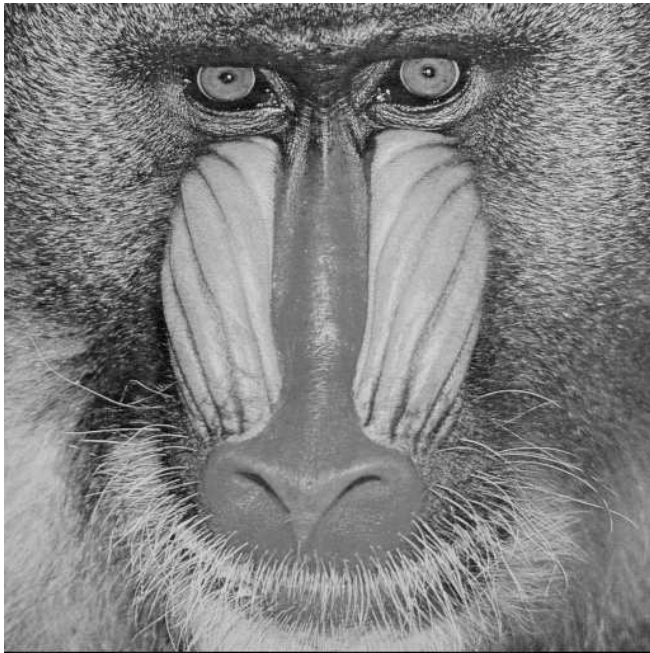
Głównym wyznacznikiem testów było sprawdzenie wygody korzystania z interfejsu, a przez to na ile on był pomocny przy dokonywaniu eksperymentów i wyciąganiu z nich poprawnych wniosków.

Do testów wybrano dwa obrazy:

a) Obraz Leny (lena.bmp)



b) Obraz baboon (baboon.bmp)



Parametry obu obrazów (odczytane w zakładce „Statistics and properties”):

- wysokość 515 pikseli
- szerokość 512 pikseli
- ilość bitów na piksel 24
- rozmiar w bajtach 786432
- rozmiar w bitach 6291456

Jak widzimy na drugim obrazie jest dużo drobnych szczegółów (składowe wysokoczęstotliwościowe), dzięki opcji przybliżania, będziemy mieć możliwość wygodnego oglądania różnic.



**Test1:** Badanie jakości w zależności od stopnia kompresji dla obydwu rodzajów filtra.

a) Obraz Leny

- format wyjściowy: jpc

Ratio	kernel	Opis zniekształceń	Czas operacji[s]	PAE R	PAE G	PAE B	MSE R	MSE G	MSE B	BPP [bit/pixel]
0,1	5/3	brak	15	27	16	25	13,2793	8,71931	18,1716	2,397
	7/9	brak	19	24	17	28	9,07875	7,88669	15,6633	2,398
0,05	5/3	Nieznaczone zniekształcenia Widoczne na tle pod powiększeniem	13	32	33	46	18,1385	16,0757	29,5516	1,199
	7/9	Jak punkt wyżej	15	27	32	45	12,935	14,6108	27,6199	1,199
0,01	5/3	Widoczny efekt kwantyzacji, lekko rozmyte krawędzie	12	74	89	116	58,3629	62,5506	76,7959	0,2395
	7/9	Jak punkt wyżej	16	54	72	124	44,0411	54,9939	73,6647	0,2393
0,001	5/3	Obraz zniekształcony, lecz da się odczytać zarys osoby	13	122	147	136	342,166	389,015	314,566	0,02399
	7/9	Większe zniekształcenia niż przy zastosowaniu filtra 5/3 – fałszywe kolory	15	120	142	143	336,318	396,626	329,656	0,02347

b) Obraz baboon

ratio	kernel	Opis zniekształceń	Czas operacji[s]	PAE R	PAE G	PAE B	MSE R	MSE G	MSE B	BPP [bit/pixel]
0,1	5/3	Brak	15	63	46	64	111,998	62,9355	127,897	2,398
	7/9	Widoczne w powiększeniu rozmycia białych odbłysek na nosie zwierzęcia	18	55	43	68	96,6721	61,5651	133,471	2,392
0,05	5/3	Wyraźne rozmycie na niebieskiego obszaru	15	86	88	95	186,729	134,164	236,811	1,197
	7/9	Jak punkt wyżej	18	69	81	104	162,348	125,644	252,638	1,195
0,01	5/3	Wyraźny efekt rozmycia niebieskiego obszaru i poniżej brody	14	161	150	131	439,27	482,323	574,478	0,2366
	7/9	Jak punkt wyżej	16	157	138	133	386,768	458,657	577,697	0,2344
0,001	5/3	Obraz zniekształcony, lecz da się odczytać zarys osoby	14	156	176	175	790,97	938,775	1097	0,02399
	7/9	Większe zniekształcenia niż przy zastosowaniu filtra 5/3 – fałszywe kolory	17	163	175	178	809,21	970,813	1127,42	0,02313

Oprócz oczywistego wniosku, zmniejszania jakości obrazu wraz ze wzrostem CR (ratio maleje), można wyciągnąć jeszcze kilka, które zaobserwowaliśmy dzięki możliwościom programu:

- Dla obrazu o większej ilości składowych wysokoczęstotliwościowych (obraz baboon), wraz ze wzrostem CR, jakość malała szybciej niż dla obrazu o małej ilości składowych wysokoczęstotliwościowych (obraz Leny) [jest to też związane ze stonowaną paletą użytych barw – ale to i tak implikuje to, że składowe wysokoczęstotliwościowe będą małe]
- Przy zastosowaniu filtra 7/9 czas operacji trwał zazwyczaj dłużej niż w wypadku stosowania filtra 5/3 – wynika to nie tylko z zasady działania kodeka, ale również zależy od rodzaju obrazka (dla obrazu o dużej zawartości wysokich częstotliwości, czas ten był nieznacznie dłuższy)
- Dla filtra 7/9 zaobserwowano znaczne perturbacje kolorów przy większym CR

Dzięki możliwości powiększenia, można było zaobserwować efekt rozmycia, już nawet przy niskich stopniach kompresji.

**Test2:** Symulacja efektu odtwarzania obrazu na podstawie pewniej ilości otrzymanych pakietów.

W zakładce „Decoder options” wybieramy „Data to read” i po wybraniu „Number of packets” wpisujemy zadaną wartość odczytanych pakietów. Jednocześnie zbadamy opcję „Progression order” w zakładce „Layers”. W tym celu najpierw będziemy kompresować obraz z zadaną progresją, z zapisany w wyniku operacji obraz w kolejnych krokach odczytywać dla różnej ilości pakietów.

Kolejność progresji	1 pakiet	2 pakiety	3 pakiety	4 pakiety	5 pakietów	6 pakietów	7 pakietów
LRCP	Obraz monochromatyczny, składowe niskoczęstotliwościowe	Obraz dwubarwny, Składowe niskoczęstotliwościowe	Kolory jak w oryginale, Składowe niskoczęstotliwościowe	Jak poprzednio, Więcej szczegółów	Nie widać różnicy	Jak poprzednio	Więcej szczegółów niż w poprzednich krokach
RLCP	Obraz monochromatyczny, Niskie częstotliwości	Obraz dwubarwny, Niskie częstotliwości	Kolory jak w oryginale, Niskie częstotliwości	Więcej szczegółów, Zmiany barwy	Brak zmian, Zmiany barwy	Jak poprzednio	Więcej szczegółów
PCRL	Monochromatyczny, Niskie częstotliwości	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Jakość bardzo dobra	Monochromatyczny, Jakość b.dobra	Pojawiają się kolejne składowe kolorów
RPCL	Monochromatyczny, Niskie częstotliwości	Obraz dwubarwny, Niskie częstotliwości	Kolory jak w oryginale, Niskie częstotliwości	Więcej szczegółów, Zmiany barwy	Brak zmian, Zmiany barwy	Jak poprzednio	Więcej szczegółów
CPRL	Monochromatyczny, Niskie częstotliwości	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Więcej szczegółów	Monochromatyczny, Jakość bardzo dobra	Monochromatyczny, Jakość b.dobra	Pojawiają się kolejne składowe kolorów

**Test3:** Wpływ ustawienia „resolution levels”

Wnioski: Ustawienie ilości poziomów na które ma być kompresowany, nabiera znaczenia w wypadku stosowania stratnej kompresji.

Jakość obrazu gwałtownie maleje ( jest to dostrzegalne ) wraz ze wzrostem CR i maleniem ilości poziomów. Poniżej obraz Leny, przy ratio = 0,05 (CR = 20), oraz resolution levels = 1



#### **Test 4:** Użycie opcji wyłączenia wielokomponentowej transformaty

Opcja ta nabiera znaczenia również w wypadku kompresji stratnej. Przykładowe wyniki, dla obrazu Leny z włączoną opcją.

Output Image: CR=200,7

Peak absolute error: R=96    G=148    B=122  
Mean squared error: R=127,074    G=178,444    B=160,43  
Root mean squared error: R=11,2727    G=13,358    B=12,6661  
Mean absolute error: R=7,69476    G=9,00766    B=8,7566

Czas obróbki 13s

Wyniki dla wyłączonej opcji:

Output Image: CR=200,5

Peak absolute error: R=81    G=118    B=122  
Mean squared error: R=101,391    G=110,022    B=121,591  
Root mean squared error: R=10,0693    G=10,489    B=11,0268  
Mean absolute error: R=7,21278    G=7,14905    B=7,89338

Czas obróbki 11s

W przypadku wyłączonej opcji otrzymujemy mniejsze wartości błędów. Objawia się to również wizualnie. W wypadku włączenia tej opcji, można dostrzec przekłamanie kolorów (efekt rozmycia barwy na obszary w ogóle z tą barwą nie związane). Krótszy też jest ~~sa~~ czas przetwarzania.

#### **Test 5:** Użycie części i opcji ustawienia obrazu na siatce odniesienia

Tu również ma znaczenie zadany stopień kompresji obrazu. Im jest on większy, tym większy jest wpływ oferowanych tu opcji.

W wypadku ustawień współrzędnych obrazu na siatce odniesienia, można przy dużym stopniu kompresji, tak je dobrać, by uzyskać jak najmniejsze błędy.

#### **Test 6:** Ustawienia stylu kodowania

Opcje tu podane nie miały bardzo znikomy wpływ na jakość kompresji. Opcje te są stosowane jak widzimy do kodowania, tak więc ich dodanie lub nie powoduje wydłużenie bądź skrócenie długości pliku wyjściowego, a ponieważ zadane ratio dla enkodera jest stałe, przez to dodatkowe bity nie niosące informacji powodują, że obraz może być nieznacznie gorszy. Subtelne różnice da się odkryć jedynie przy pomocy obliczeń, ponieważ wizualne porównanie obrazów nie wykazało różnic.

W wypadku bardzo dużego CR, ustawienia te mogą bardzo wpłynąć na jakość obrazu zdekompresowanego, ale jest to związane właśnie z nadmiarowością informacji (w małym pliku (2-3kB) stosunek informacji potrzeba na zapisanie markerów i innych znaczników w stosunku do użytecznej informacji jest znacznie większy niż w przypadku pliku o dużych rozmiarach(50-..kB)).

### **Test 7: Ustawienia dzielnic i wielkości bloków kodowych**

Ustawienia wielkości dzielnic jak i bloków kodowych nie wpływają na jakość kompresji.

### **Wnioski:**

Prezentowana aplikacja okazała się bardzo przydatnym i wygodnym narzędziem, ponieważ zwalnia użytkownika z uciążliwej obsługi konsolowego programu jasper.exe, pamiętania poszczególnych skrótów opcji, oraz ręcznego przeglądania wyników pracy.

W takim rozwiązaniu, możemy w pełni wykorzystać opcje jakie niesie program „jasper”, szybko zbadać dobór różnych opcji i trymować tak parametry, by uzyskać oczekiwany rezultat.