

Marcin Raniszewski\*

## Stratna kompresja obrazu z wykorzystaniem aproksymacji liniowej

### 1. Wstęp

Wzrost zapotrzebowania na komunikację, mobilność, wygodę i dobrą jakość usług pociąga za sobą rozwój wielu dziedzin w tym kompresji danych. Pojęcie to pojawia się w wielu rozważaniach na temat nowoczesnych metod komunikacji. Kompresja zmniejsza przesyłane dane, daje szybszy dostęp do informacji, owocuje zaoszczędzeniem miejsca na nośnikach. Wymyślne algorytmy kompresji stratnej korzystają z wad zmysłów ludzkich, zapisując dane w sposób, który umożliwi usunięcie z nich niezbyt ważnych i często trudno zauważalnych szczegółów. Dzięki temu skompresowane dane zajmują o wiele mniej miejsca i mogą być o wiele szybciej przesyłane, zapewniając wysoki komfort pracy i wygodę [3, 5].

Szukanie zależności pomiędzy danymi, które opisują pewien obiekt, może być niezwykle pasjonujące i interesujące. Różnorodność takich obiektów daje szerokie pole do badań skuteczności zaproponowanego algorytmu.

Znane standardy kompresji stratnej obejmują:

- JPEG (*Joint Photographic Expert Group*) do kompresji obrazów,
- MPEG (*Moving Picture Expert Group*) do kompresji sekwencji wideo,
- MP3 (*MPEG Audio Laser 3 encoding*) do kompresji muzyki.

W artykule tym przedstawiony został nowy algorytm kompresji. Wykorzystuje się w niej technikę przybliżania barw pojedynczych punktów obrazu za pomocą aproksymacji liniowej, techniki zaczerpniętej z metod numerycznych [1]. Taka technika kompresji stratnej została zastosowana głównie ze względu na dużą liczbę obrazów, w których ważne jest zachowanie kształtów przedstawionych obiektów, przy jednoczesnej możliwości utraty szczegółów ich struktury.

### 2. Kompresja danych

Kompresja danych jest nauką reprezentowania informacji w zwartej postaci. Aby móc dobrze kompresować dane, musimy poznać ich własności strukturalne i na ich podstawie oprzeć algorytm kompresji.

---

\* Katedra Informatyki Stosowanej, Politechnika Łódzka

Gdy mówimy o technice lub algorytmie kompresji, mamy na myśli dwa algorytmy. Pierwszy to algorytm kompresji, który dla pewnych danych wejściowych  $X$  generuje reprezentację  $X_c$ . Drugi z tych algorytmów, to tak zwany algorytm dekompresji, który na podstawie reprezentacji  $X_c$  tworzy rekonstrukcję  $Y$ . Zatem algorytmem kompresji będziemy nazywać parę: algorytm kompresji i algorytm dekompresji.

Algorytmy kompresji dzielą się na dwie klasy:

- 1) algorytmy kompresji bezstratnej,
- 2) algorytmy kompresji stratnej.

W schemacie kompresji bezstratnej wymagamy, aby rekonstrukcja  $Y$  i dane oryginalne  $X$  były identyczne, natomiast w schemacie kompresji stratnej, która zazwyczaj pozwala na większą kompresję niż kompresja bezstratna, dopuszcza się, aby  $Y$  i  $X$  były różne [3, 5].

### 2.1. Kompresja stratna

W technikach kompresji stratnej dopuszczamy pewną utratę informacji. Dane zrekonstruowane nie pokrywają się z oryginalnymi, pojawiają się zniekształcenia. Jest to cena, jaką musimy zapłacić za (często) lepszy stopień kompresji niż w przypadku kompresji bezstratnej.

Dzięki temu, że w wielu zastosowaniach nie wymagamy dokładnej rekonstrukcji kompresowanych danych, kompresja stratna ma szerokie zastosowanie.

W zależności od tego, jaka ma być jakość rekonstrukcji, możemy dopuszczać różne wielkości, o które może się różnić oryginalna wartość próbki od zrekonstruowanej [3, 5].

### 2.2. Miary jakości kompresji

Istnieje wiele różnych kryteriów, według których można oceniać algorytmy kompresji. Możemy mierzyć złożoność algorytmu, pamięć potrzebną do jego implementacji, szybkość działania na konkretnym komputerze, rozmiar kompresji oraz podobieństwo danych po dekompresji do danych oryginalnych.

Jednym z najważniejszych kryteriów opisującym, jak efektywnie algorytm kompresuje określony zbiór danych, jest stosunek liczby bitów potrzebnych do reprezentacji danych przed kompresją do liczby bitów potrzebnych do reprezentacji danych po kompresji. Proporcja ta nazywana jest *stopniem kompresji*.

W kompresji stratnej dane uzyskane po dekompresji różnią się od danych oryginalnych. Aby zatem móc określić efektywność algorytmu kompresji, potrzebujemy sposobu mierzenia tych różnic. Różnicę pomiędzy danymi oryginalnymi i danymi po dekompresji nazywamy często *zniekształceniem* [3, 5].

### 2.3. Kryteria oceny zniekształceń

Naturalnym sposobem oceny wierności dekompresji jest rozważenie różnic pomiędzy wartościami oryginalnymi i zrekonstruowanymi, czyli zniekształceń powstałych w trakcie kompresji. *Kwadratowa miara błędu* i *bezwzględna miara błędu* to popularne miary zniekształceń. Nazywa się je także *różnicowymi miarami zniekształceń*.

Trudno jest powiedzieć cokolwiek na temat różnic pomiędzy całymi dokumentami (oryginałem i rekonstrukcją) na podstawie ciągu różnic poszczególnych elementów (np. pikseli). Dlatego też wprowadza się sumy różnic pomiędzy poszczególnymi elementami ciągów. Najpopularniejszą miarą tego typu jest średnia arytmetyczna z kwadratów poszczególnych elementów ciągu, nazywana *błędem średniokwadratowym*. Inną często wykorzystywaną miarą zniekształceń jest również *wartość błędu bezwzględnego*. Miara ta jest szczególnie użyteczna przy ocenie algorytmów kompresji obrazów.

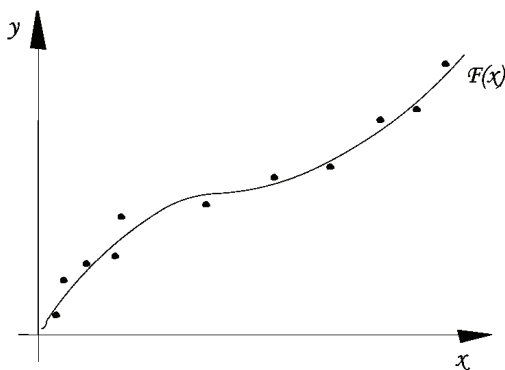
W niektórych sytuacjach, takich jak wymóg wielkości zniekształceń mniejszej od pewnej wartości progowej, istotnym parametrem jest *maksymalna wielkość błędu* [3, 5].

### 3. Aproksymacja

Celem aproksymacji jest znalezienie funkcji  $F(x)$  (rys. 1), która przybliży pewną funkcją  $f(x)$ , określoną w postaci ciągu punktów:

$$(x_0, f(x_0) = y_0), (x_1, f(x_1) = y_1), \dots, (x_n, f(x_n) = y_n) \quad (1)$$

Punkty  $x_0, x_1, \dots, x_n$  nazywamy *węzłami aproksymacji*. Mogą one pochodzić z pomiarów, albo być wynikami innych obliczeń. W przypadku kompresji, zaproponowanej w niniejszej pracy, punkty te będą określały parę: piksel i jego składowa barwa (czerwona, zielona lub niebieska).



Rys. 1.  $F(x)$  aproksymuje przykładowe punkty pomiarowe

Funkcję aproksymującą buduje się najczęściej w postaci kombinacji liniowej ortogonalnych funkcji bazowych  $U_0(x), U_1(x), \dots, U_m(x)$

$$F(x) = a_0U_0(x) + a_1U_1(x) + a_2U_2(x) + \dots + a_mU_m(x) \quad (2)$$

Celem naszych obliczeń jest znalezienie współczynników  $a_0, a_1, \dots, a_m$ .

Miarą zgodności funkcji aproksymującej  $F(x)$  i funkcji aproksymowanej  $f(x)$  jest norma:

$$\|E(f)\| = \|F(x) - f(x)\| \quad (3)$$

Do najczęściej stosowanych norm należą:

- norma średniokwadratowa:

$$\|E(f)\|_2 = \sqrt{\sum_{i=0}^n (F(x_i) - f(x_i))^2} \quad (4)$$

- norma maksymalna:

$$\|E(f)\|_{\max} = \max_i |F(x_i) - f(x_i)| \quad (5)$$

Ze względu na normę możemy wyróżnić dwa rodzaje aproksymacji. Mamy wtedy do czynienia z *aproksymacją metodą najmniejszych kwadratów* (norma średniokwadratowa) lub z *aproksymacją jednostajną* (norma maksymalna). Zależy nam by norma było najbliższa zero, czyli aby funkcja  $F(x)$  możliwie najwierniej odzwierciedlała przebieg funkcji  $f(x)$ . W naszym przypadku będziemy korzystać z aproksymacji metodą najmniejszych kwadratów i nią dalej będziemy się zajmować.

Drugim rodzajem podziału aproksymacji jest podział ze względu na funkcje bazowe. Mamy tutaj: *aproksymację wielomianową* i *aproksymację trygonometryczną*. Zajmować się będziemy aproksymacją wielomianową, czyli aproksymacją, w której  $F(x)$  będzie skonstruowana za pomocą funkcji bazowych, będących wielomianami [1].

### 3.1. Aproksymacja liniowa metodą najmniejszych kwadratów

Aproksymacja liniowa jest szczególnym przypadkiem aproksymacji wielomianowej (we wzorze (2)  $m = 1$ ). Funkcja aproksymująca jest funkcją liniową postaci

$$F(x) = a_0 + a_1x \quad (6)$$

Posługując się metodą szukania współczynników aproksymacji wielomianowej metodą najmniejszych kwadratów, otrzymujemy:

$$a_0 = \frac{t_0S_2 - t_1S_1}{S_0S_2 - S_1^2} \quad (7)$$

$$a_1 = \frac{t_1S_0 - t_0S_1}{S_0S_2 - S_1^2} \quad (8)$$

gdzie:

$$S_k = \sum_{i=0}^n x_i^k, \quad k = 0, 1, 2 \quad (9)$$

$$t_0 = \sum_{i=0}^n y_i \quad (10)$$

$$t_1 = \sum_{i=0}^n x_i y_i \quad (11)$$

zaś współrzędne  $x_i, y_i$  to współrzędne aproksymowanych punktów (patrz wzór (1)).

Łatwo pokazać, że czynnik w mianownikach wzorów (7) i (8) nigdy nie przyjmie wartości zero.

#### 4. Algorytm stratnej kompresji obrazu z wykorzystaniem aproksymacji liniowej

Jeśli kilka sąsiednich pikseli obrazu oznaczymy kolejnymi liczbami naturalnymi, a ich barwy przeliczymy w pewien sposób na liczby, to będziemy mogli utworzyć pary postaci: (numer piksela, jego barwa). Ich rozmieszczenie w kartezjańskim układzie współrzędnych może być różne w zależności od tego, jakie barwy reprezentują nasze piksele. Mówiąc o liniowym przejściu pomiędzy barwami sąsiednich pikseli, będziemy mieć na myśli ich reprezentację w kartezjańskim układzie współrzędnych, gdzie układają się one w kształt linii prostej. Należy zauważyć, że obszary jednobarwnych lub prawie jednobarwnych pikseli będą układać się na wykresie także jako linia prosta, równoległa do osi  $OX$ .

Analizując obrazy, można zauważyć, że niektóre z nich mają dużo liniowych przejść pomiędzy barwami zarówno w poziomie, jak i pionie. Często obraz zawiera płaszczyzny jednokolorowe lub nieznacznie, prawie niezauważalnie, zmieniające swoją barwę, co jest równoznaczne z linią równoległą do osi  $OX$  w kartezjańskim układzie współrzędnych (przy powyżej opisanej reprezentacji pikseli obrazu).

Rysunek 2 zawiera duże jednokolorowe płaszczyzny oraz liniowe przejścia koloru. Widzimy to wyraźniej na rysunku 3, który jest powiększeniem fragmentu policzka dziecka z rysunku 2.

Zaproponowana metoda kompresji wykorzystuje aproksymację liniową do liniowego przybliżania zmienności barw pikseli [4]. Zamiast informacji o kolorze każdego piksela (tak jak ma to miejsce w bitmapie [2]), w ramach jednej prostej aproksymującej przechowywane są informacje o ilości aproksymowanych pikseli i o barwach pierwszego i ostatniego piksela. Należy zauważyć, że ostatni piksel pewnej prostej aproksymującej jest jednocześnie pierwszym pikselem następnej prostej.

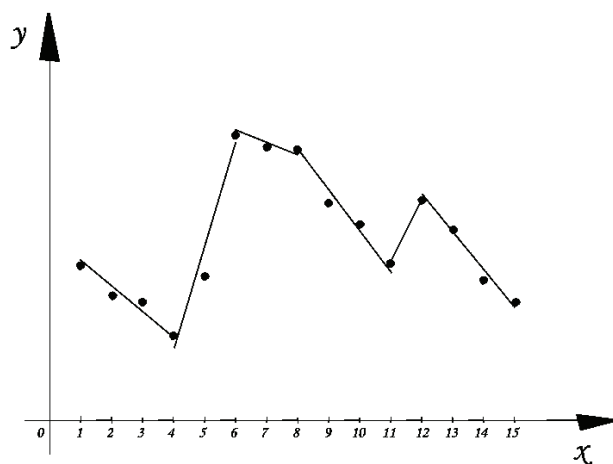


**Rys. 2.** Widoczne duże jednokolorowe obszary oraz liniowe zmiany koloru szarego na policzku dziecka



**Rys. 3.** Powiększenie policzka dziecka z rysunku 2

Układ przedstawiony na rysunku 4 jako całość jest chaotyczny i trudno by było dopatrzeć się w nim jakiś prawidłowości. Istnieją różne obrazy, które jako całość tworzą układy odmienne i trudne jest, a może nawet niemożliwe, znalezienie uniwersalnej metody, która opierałaby się na ogólnych właściwościach dowolnego obrazu. Dzielic jednak ten układ na pewne podukłady, uzyskujemy możliwość aproksymacji liniowej par pikseli i ich barw w obrębie wymienionych podukładów.



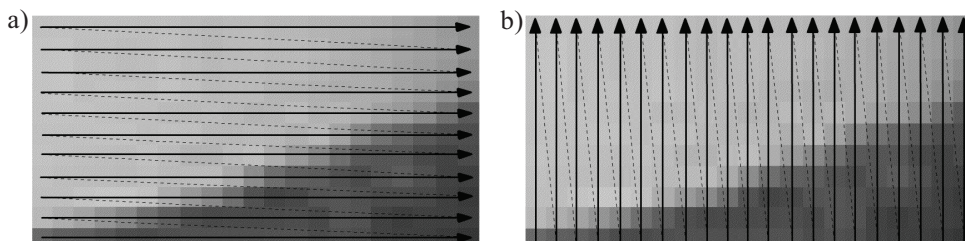
Rys. 4. Aproksymacja liniowa przykładowego rozmieszczenia par pikseli i ich barw na wykresie

Można zatem przewidzieć, że za pomocą proponowanej kompresji o wiele lepiej kompresowały się będą obrazy o dużej ilości liniowych zmian kolorów, zachodzących w liniach poziomych lub pionowych. Gorzej będą się kompresowały obrazy o dużej ilości „ostrych” przejść pomiędzy kolorami, w których mało jest jednobarwnych obszarów i liniowych przejść pomiędzy barwami.

Proponowana kompresja, jak już zostało to powiedziane, jest kompresją stratną. Podczas aproksymacji liniowej tracona jest informacja o dokładnej barwie pikseli. Wartości te są przybliżane przez aproksymującą je prostą. Ponieważ oko ludzkie jest niedoskonałe, więc w większości przypadków przy dobrych parametrach kompresji, straty będą niezauważalne.

#### 4.1. Wczytanie bitmapy i aproksymacja liniowa

Bitmapa może zostać wczytana w liniach poziomych lub pionowych (rys. 5).

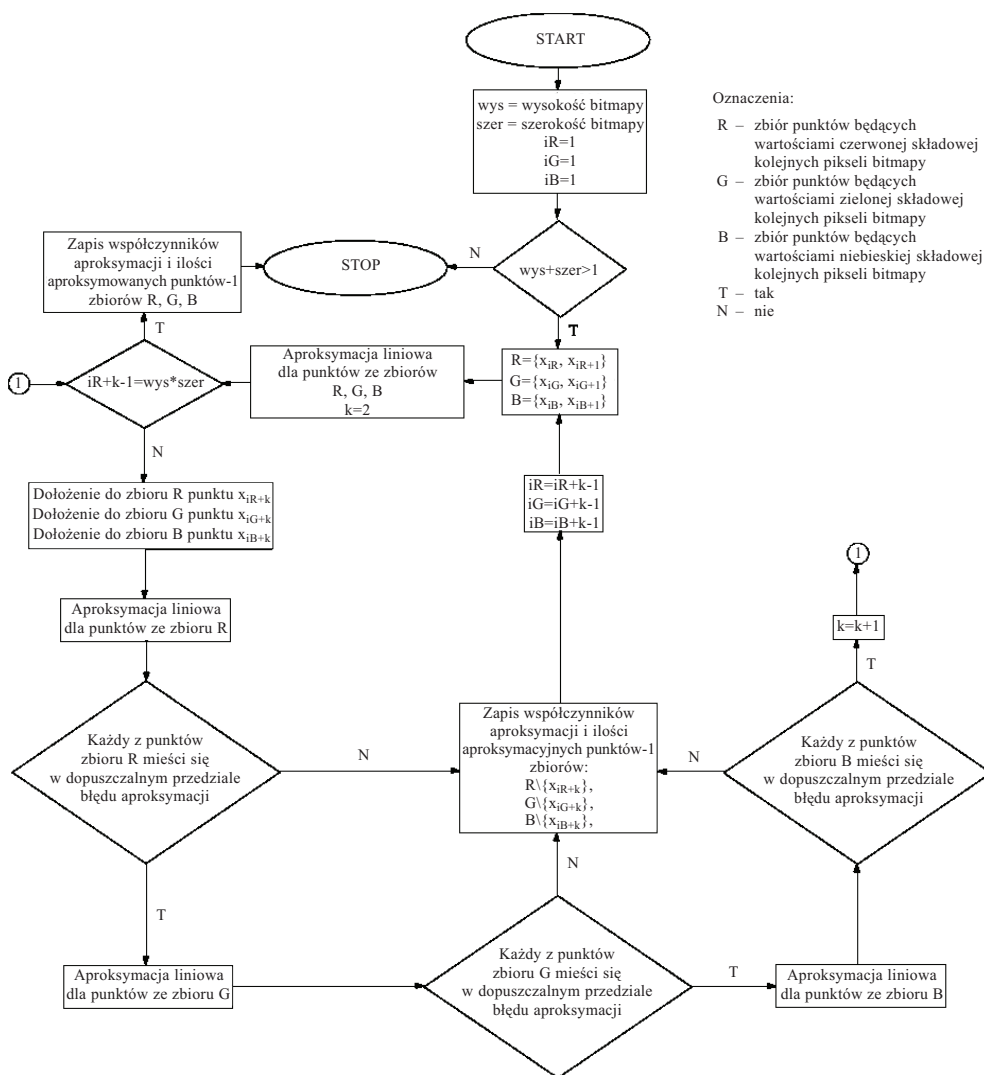


Rys. 5. Wczytywanie obrazu: w liniach poziomych (a) i w liniach pionowych (b)

Omawiana metoda kompresji została zaimplementowana do wczytywania 24-bitowych bitmap. Kolor pojedynczego piksela zapisywany jest zatem na trzech bajtach w postaci RGB [2].

Szczytany ciąg poddawany jest aproksymacji liniowej: każda składowa barwy aproksymowana jest osobno.

Następnie brane są kolejno: czerwone, zielone i niebieskie składowe koloru dwóch pierwszych pikseli i dla nich, oddzielnie, liczone są współczynniki aproksymacji. W schemacie na rysunku 6 składowe te są oznaczone jako litery  $x$  z odpowiednim indeksem.



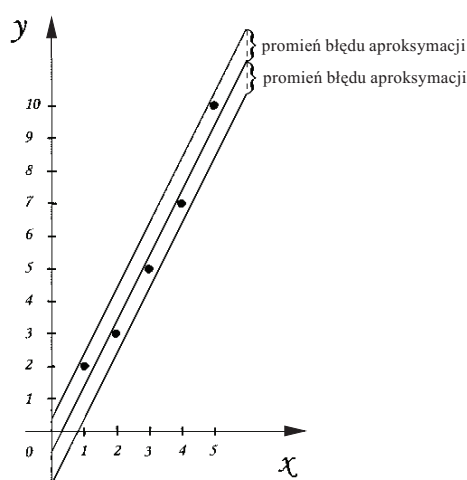
Rys. 6. Schemat przebiegu aproksymacji liniowej w algorytmie kompresji



W przedstawionej metodzie kompresji pojawia się pojęcie promienia  $r$  błędu aproksymacji. Jeśli aproksymowana barwa piksela wynosi  $b_0$ , a barwa rzeczywista  $b$ , to będziemy akceptować taką aproksymację jeżeli

$$b \in (b_0 - r, b_0 + r) \quad (12)$$

czyli  $b$  będzie się zawierać w jednowymiarowym kole o środku  $b_0$  i promieniu  $r$  (rys. 7).



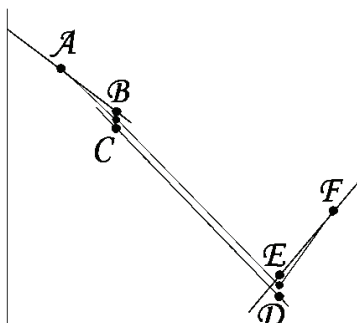
**Rys. 7.** Aproksymacja liniowa przykładowych punktów z przedziałem wyznaczonym przez promień błędu aproksymacji

W momencie gdy któryś z punktów poddanych aproksymacji nie mieści się w dopuszczalnym przedziale błędu aproksymacji (ze względu na którąkolwiek ze składowych barw), zapisywane są współczynniki aproksymacji i liczba aproksymowanych pikseli dla zbiorów punktów bez piksela ostatnio dodanego [4].

#### 4.2. Uśrednianie początków i końców aproksymujących prostych

Zapamiętywanie współczynników aproksymujących prostych oparte jest na zapamiętaniu rzędnej składowej barwy pierwszego aproksymowanego piksela oraz rzędnej składowej barwy ostatniego aproksymowanego piksela, czyli rzędnej początku i końca aproksymującej prostej.

Ponieważ koniec jednej prostej ma zazwyczaj inną rzędną niż początek kolejnej prostej (rys. 8), dlatego stosuje się ich uśrednianie. Bierze się rzędną końca jednej prostej oraz rzędną początku kolejnej prostej (punkty o tej samej odciętej) i wylicza się ich średnią arytmetyczną. Uzyskuje się w ten sposób punkt pośredni, który od tej pory będzie wspólnym końcem jednej prostej i początkiem kolejnej prostej. Położenie prostych ulega minimalnej zmianie [4].



Rys. 8. Uśrednienie końców i początków aproksymujących prostych

### 4.3. Wybranie najczęściej występujących różnic pomiędzy składowymi barw

Mając ciągi punktów, które łączą proste aproksymujące, oraz ciąg ilości pikseli aproksymowanych przez każdą z tych prostych, możemy już zapisywać te dane do pliku (zapis trzech składowych barwy pierwszego piksela, następnie ilości aproksymowanych składowych barw pikseli pomniejszonej o 1 i trzech składowych barwy piksela, który jest końcem jednej prostej, a zarazem początkiem kolejnej).

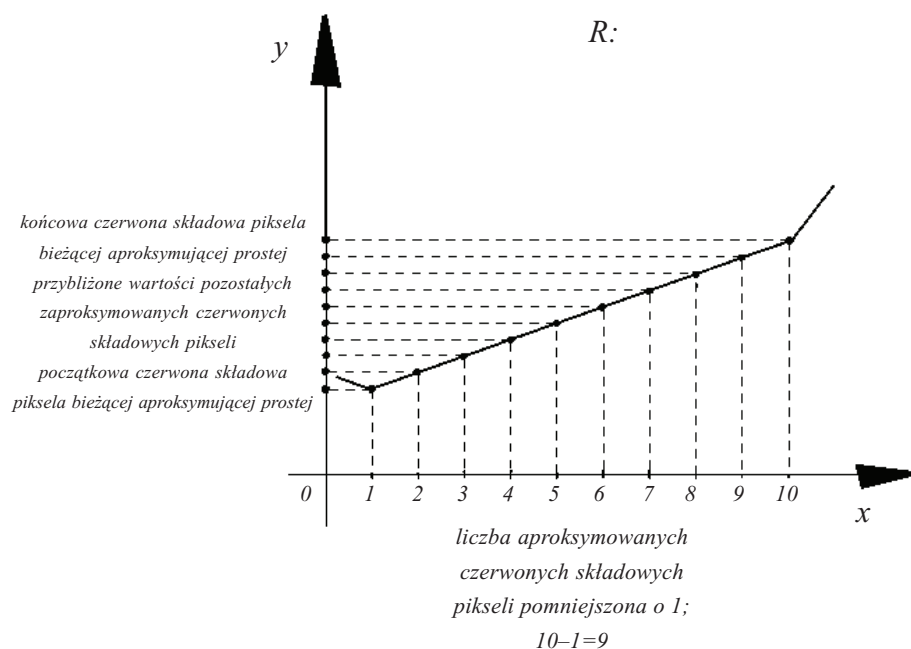
Na zapis barwy piksela zużywamy 24 bity. Prezentowana metoda kompresji umożliwia redukcję tych bitów poprzez przypisanie krótszych ciągów bitowych (od 1 do 7 bitów) najczęściej występującym różnicom pomiędzy składowymi barw. Możliwe jest przypisanie innej ilości bitów do zakodowania składowej czerwonej, zielonej i niebieskiej koloru oddzielnie.

Użycie powyższego rozwiązania prowadzi do uzyskania większego stopnia kompresji dla większych obrazów, wydłuża jednak nieznacznie sam algorytm kompresji (wyszukiwanie najczęściej występujących różnic pomiędzy składowymi barw) [4].

### 4.4. Algorytm dekompresji

Mając początek i koniec aproksymujących prostych oraz liczbę aproksymowanych pikseli pomniejszoną o 1, można przybliżyć wartości wszystkich aproksymowanych pikseli.

Różnicę pomiędzy składową barwy czerwonej końcowego piksela i początkowego należy podzielić przez liczbę aproksymowanych pikseli pomniejszoną o 1 i otrzymaną wartość dodawać do składowej czerwonej początkowego piksela, otrzymując wartości czerwonych składowych kolejnych pikseli (rys. 9). Podobnie postępujemy w przypadku zielonej i niebieskiej składowej. Uzyskane piksele zapisujemy w odpowiedni sposób do pliku z bitmapą (w zależności od tego, czy kompresja była pozioma czy pionowa) [2, 4].



**Rys. 9.** Odczyt wartości czerwonej składowej pikseli z przykładowej prostej aproksymującej

#### 4.5. Testowanie

Testom zostały poddane trzy rodzaje 24-bitowych bitmap (rozszerzenie .bmp):

bitmapa 1 – oryginalny obraz próbki srebra w temperaturze 982°C – rozmiar: 101 178 bajtów, 212×159 pikseli;

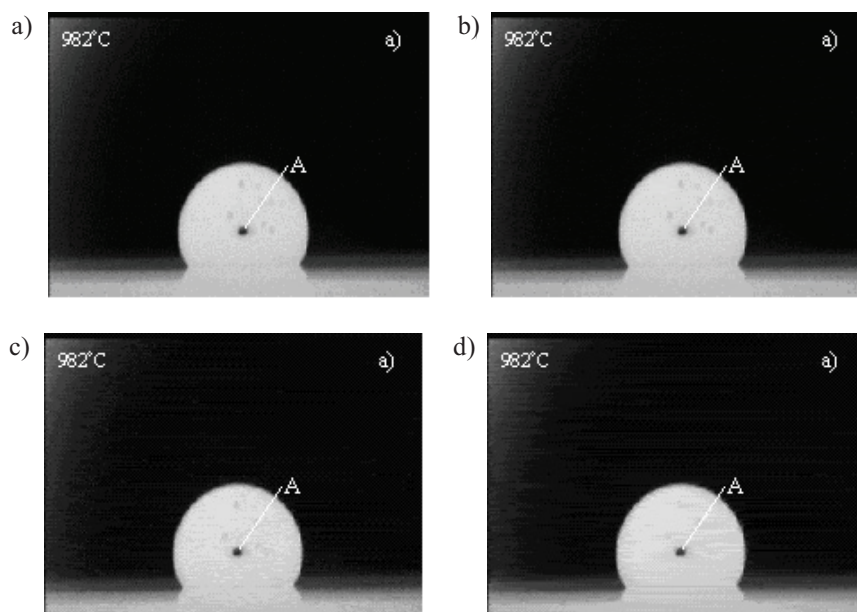
bitmapa 2 – rysunek 9, rozmiar: 983 922 bajtów, 670×489 pikseli;

bitmapa 3 – krajobraz, rozmiar: 1 440 054 bajty, 800×600 pikseli.

Publikowane w artykule wyniki obejmują tylko kompresję poziomą, z zapisem najczęściej występujących różnic pomiędzy składowymi barw na 4 bitach.

##### Kompresja bitmapy 1

Zastosowanie przedstawionej metody kompresji w przypadku bitmapy 1 (rys. 10) daje zadowalające rezultaty (tab. 1). Już przy promieniu aproksymacji rzędu 8 uzyskiwany jest dobry stopień kompresji przy nieznacznej utracie informacji z obrazu. Kształt próbki jest zachowany, widoczne są nieliczne rozmycia poziome pikseli w obrębie jednolitych obszarów.



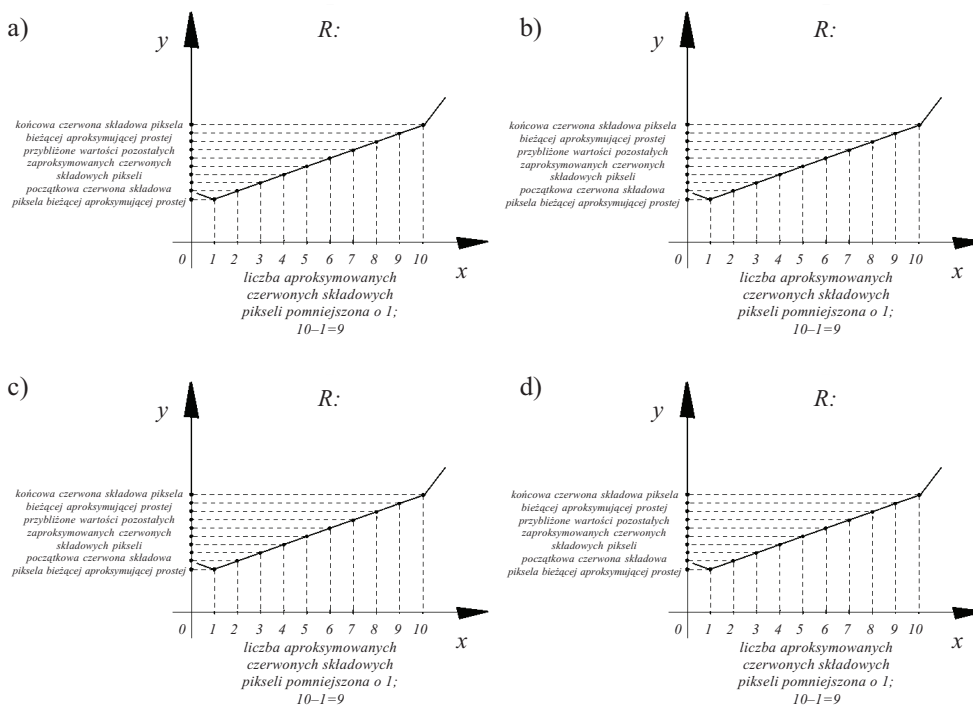
**Rys. 10.** Kompresja bitmapy 1: oryginalny obraz (a) oraz zdekompresowane obrazy dla rosnących wartości promienia aproksymacji (b, c, d)

**Tabela 1**  
Wyniki kompresji bitmapy 1

Promień błędu aproksymacji	Stopień kompresji	Błąd bezwzględny	Maksymalna wielkość błędu	Rysunek
4,0	2,9792	1,0873	7	10b
8,0	5,9193	2,0834	13	10c
12,0	10,8467	2,9513	20	10d

### Kompresja bitmapy 2

W przypadku kompresji bitmapy 2 (rys. 11) już przy małym promieniu aproksymacji uzyskujemy wysoki stopień kompresji (tab. 2). Wraz ze wzrostem promienia aproksymacji stopień kompresji nieznacznie rośnie. Dalsze zwiększanie promienia kompresji (powyżej 12) będzie powodowało poziome rozmywanie się przedstawionych na obrazie liter i wykresu. Przy dość wysokim promieniu aproksymacji (rzędu 12) obraz jest nadal czytelny i przejrzysty (rys. 11).



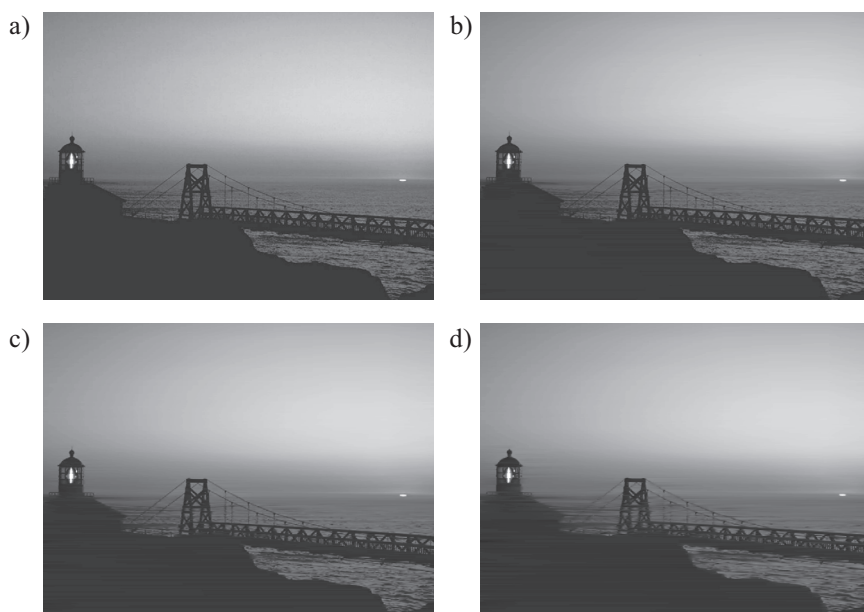
Rys. 11. Kompresja bitmapy 2: oryginalny obraz (a) oraz zdekompresowane obrazy dla rosnących wartości promienia aproksymacji (b, c, d)

Tabela 2  
 Wyniki kompresji bitmapy 2

Promień błędu aproksymacji	Stopień kompresji	Błąd bezwzględny	Maksymalna wielkość błędu	Rysunek
4,0	14,5096	7,8340e-5	1	11b
8,0	14,9728	0,0491	13	11c
12,0	15,5787	0,1623	18	11d

### Kompresja bitmapy 3

Bitmapa 3 (rys. 12) przedstawia krajobraz z poziomym układem barw. Przewidywać można, że dobrze będzie się ona kompresować przy wykorzystaniu kompresji poziomej. I rzeczywiście: już przy promieniu aproksymacji równym 4 uzyskujemy zadawalające wyniki (tab. 3). Obserwujemy proporcjonalny wzrost stopnia kompresji wraz ze wzrostem promienia aproksymacji. Wpływa to jednak na deformacje jakie powstają na zdekompresowanym obrazie (rys. 12). Widoczne są poziome rozmazania latarni morskiej mostu.



**Rys. 12.** Kompresja bitmapy 3: oryginalny obraz (a) oraz zdekompresowane obrazy dla rosnących wartości promienia aproksymacji (b, c, d)

**Tabela 3**  
Wyniki kompresji bitmapy 3

Promień błędu aproksymacji	Stopień kompresji	Błąd bezwzględny	Maksymalna wielkość błędu	Rysunek
4,0	11,9091	1,0214	8	12b
8,0	20,8299	1,5056	16	12c
12,0	28,9069	1,9520	23	12d

### Porównanie z algorytmami kompresji JPEG, GIF, PNG i ZIP

Wspomniana w rozdziale 1 kompresja stratna JPEG jest przeznaczona do obrazów naturalnych, z dużą ilością nieostrych krawędzi i małą ilością detali. Kompresja GIF (*Graphics Interchange Format*) jest kompresją bezstratną operującą na 256 kolorach i przeznaczoną głównie do grafik sztucznych. PNG (*Portable Network Graphics*) został opracowany jako następcą GIF i główną jego zaletą w stosunku do poprzednika jest możliwość obsługi większej palety kolorów. ZIP to bardzo popularna bezstratna kompresja danych ogólnego przeznaczenia.

**Tabela 4**  
Zestawienie stopni kompresji wybranych algorytmów

Bitmapa	JPEG (jakość: 8)	GIF	PNG	ZIP	Zaproponowany algorytm (promień błędu aproksymacji: 8,0)
1	5,8651	9,7315	4,9910	7,5489	5,9193
2	16,0447	40,9461	20,7786	73,8575	14,9728
3	28,5952	9,3098	6,0504	4,7923	20,8299

Wyniki uzyskane przez zaproponowany algorytm przy kompresji bitmapy 1 są zbliżone do wyników kompresji JPEG oraz PNG (tab. 4). Pozostałe kompresje uzyskały lepsze wyniki. Bitmapę 2 najlepiej skompresowały ZIP i GIF. Omawiana kompresja i JPEG uzyskały podobne (najniższe) stopnie kompresji. W przypadku bitmapy 3 stopień kompresji otrzymany przez zastosowanie omawianego algorytmu jest wyższy w stosunku do reszty algorytmów kompresji bezstratnej.

## 5. Wnioski

Przedstawiona kompresja wykorzystuje aproksymację liniową do przybliżania liniowych układów barw punktów, z których złożona jest bitmapa. Jest to kompresja stratna, gdyż tracona jest informacja o dokładnych kolorach większości pikseli. Przy dekompresji są one przybliżane na podstawie prostych, które je aproksymują.

Bitmapy, w których dużo miejsca zajmują jednobarwne obszary lub obszary o podobnym kolorze, na których widać strukturę liniową przedstawianego obrazu, kompresują się z wysokim stopniem kompresji.

Wszelkie rysunki techniczne, wykresy, teksty, rysunki, zawierające małą liczbę różniących się od siebie kolorów z „ostrymi” przejściami pomiędzy nimi i dużą ilość obszarów jednobarwnych lub o podobnych barwach, kompresują się z wysokim stopniem kompresji już dla małych promieni błędu aproksymacji. Praktycznie niezauważalna jest w nich strata jakości obrazu.

Bitmapy czarno-białe, w których widoczna jest liniowa struktura, kompresują się również bardzo dobrze. Przy wyższych stopniach kompresji występują jednak dające się zauważyć straty.

Bitmapy kolorowe, które mają dużo liniowych przejść pomiędzy kolorami, jednobarwnych obszarów lub obszarów o podobnym kolorze, kompresują się zadowalająco. Jednakże, podobnie do bitmap czarno-białych, przy wyższych stopniach kompresji, występują widoczne utraty w jakości obrazu.

Bitmapy różnobarwne z dużą zmiennością kolorystyki kompresują się słabo. Zdarza się nawet, że przy małym promieniu błędu aproksymacji, uzyskany po skompresowaniu plik jest większy od kompresowanej bitmapy. Wynika to z faktu, że przy dużej zmienności barw, informacja zapisywana po kompresji jest większa od tej zapisanej w bitmapie. Jeśli

aproksymowane zostały dwa punkty, to w bitmapie zapisywane są ich barwy, podczas gdy przy kompresji dodatkowo zapamiętywana jest liczba aproksymowanych punktów pomniejszona o 1. Przy większych promieniach aproksymacji uzyskujemy dla takich bitmap stopień kompresji sięgający 2. Pojawiają się jednak wtedy straty jakości obrazu.

Bitmapy o przewadze poziomych przejść pomiędzy kolorami lepiej kompresują się za pomocą kompresji poziomej, zaś te o przewadze pionowych – za pomocą kompresji pionowej.

Dobry dobór ilości bitów przeznaczonych na zapis najczęściej występujących różnic pomiędzy składowymi barw pikseli może, przy bitmapach dobrze kompresujących się, zwiększyć stopień kompresji. Z przeprowadzonych testów wynika, że opłaca się stosować kodowanie tych różnic. Najwyższe stopnie kompresji, przy stałym promieniu błędu aproksymacji, uzyskiwane były dla 4 bitów przeznaczonych na zapis różnic pomiędzy składowymi barw. Zdarzały się jednak sytuacje, gdy brak zapisu najczęściej występujących różnic pomiędzy składowymi barw procentował w postaci większego stopnia kompresji.

Przy bardzo małych bitmapach, kodowanie najczęściej występujących różnic powoduje niższe stopnie kompresji. Wynika to z faktu, że zapisywana jest wtedy dodatkowa informacja o tym, jaki ciąg bitowy koduje daną różnicę.

Należy pamiętać, że opisana w pracy kompresja opiera się tylko na jednym algorytmie. Zaawansowane kompresje obrazów korzystają z kilku połączonych ze sobą algorytmów. Dalsze prace mogą być przeprowadzone w związku z połączeniem opisanego algorytmu z istniejącymi.

## Literatura

- [1] Björck Ł., Dahlquist G.: *Metody numeryczne*. Warszawa, PWN 1987
- [2] Kay D.C., Levine J.R.: *Graphics file formats*. Windcrest, McGraw-Hill, Cambridge, 1992
- [3] Sayood K.: *Kompresja danych – wprowadzenie*. RM, Warszawa, 2002
- [4] Raniszewski M.: *Stratna kompresja obrazu z wykorzystaniem aproksymacji liniowej*. Wydział Matematyki, Uniwersytet Łódzki, 2004 (Praca magisterska)
- [5] Przelaskowski A.: *Kompresja danych*. BTC, Warszawa, 2005