

Specyfikacja rastrowego pliku graficznego: .pakim (Projekt grafika Komputerowa i Multimedialna)

Temat Projektu

Należy utworzyć specyfikację **rastrowego** pliku graficznego rejestrującego obraz **kolorowy** (z wykorzystaniem 16 narzuconych oraz 16 dedykowanych barw) i w **16 stopniowej** skali szarości we wszystkich przypadkach opierającego się na kompresji **LZW**. Alfabet wejściowy to wartość **4 bitowa**.

Należy napisać aplikacje, które zgodnie ze stworzoną specyfikacją dokonają filtracji danych wejściowych (przystosowanie danych do alfabetu wejściowego) i konwersji z pliku BMP do nowego rodzaju pliku graficznego oraz z nowego rodzaju pliku do formatu BMP. Użytkownik powinien mieć możliwość m.in. wyboru jednego z trzech trybów barwnych (paleta narzucona, paleta dedykowana lub skala szarości).

Specyfikacja pliku .pgkim

Użyty algorytm kompresji: **LZW**.

Rozszerzenie: *.pgkim

Typ: grafika rastrowa.

Ilość kolorów: 16

Obsługiwane tryby koloru: RGB

Opis naszego formatu:

Format .pgkim jest typem pliku graficznego, który jest zaprojektowany dla środowiska Windows. Został on utworzony po to, aby przechowywać grafikę rastrową. Został on opracowany w celu przechowywania obrazów które zostaną wygenerowane przez nasz program, który będzie konwersował pliki o rozszerzeniu .BMP. W formacie .pgkim do zapisu obrazu wykorzystywane jest 4 bity na piksel.

Nazwa struktury	Rozmiar	Przeznaczenie
Nagłówek pliku	14 bajtów	Informacje o pliku
Nagłówek DIB	52 bajty	Informacje o obrazie
Paleta kolorów	Różny rozmiar	Definiuje kolory użyte przez obraz
Tablica pikseli	Różny rozmiar	Definiuje wartości poszczególnych pikseli

Nagłówek pliku
Przechowuje informacje o pliku.

Offset	Nazwa	Rozmiar	Opis
0x0	Rozszerzenie pliku	5 bajty	Rozszerzenie pliku zakodowane w ASCII
0x5	Rozmiar pliku	4 bajty	Rozmiar naszego pliku w bajtach
0x9	Offset	4 bajty	Początkowy adres bitów w tablicy pikseli
0xE	Tryb koloru	1 bajt	Zawiera informacje o trybie koloru

Nagłówek DIB
Przechowuje informacje o obrazie.

Offset	Nazwa	Rozmiar	Opis
0xF	biWysokos	5 bajtów	Wysokość obrazu w pikselach
0x14	biSzerokosc	5 bajtów	Szerokość obrazu w pikselach
0x19	biRozmiarObrazu	5 bajtów	Rozmiar rysunku w bajtach
0x1E	biRozmiar	5 bajtów	Rozmiar nagłówka informacyjnego
0x23	biKolory	16 bajty	Tabela kolorów
0x33	biSłownik	16 bajty	Rozmiar słownika

Tablica pikseli:

Gromadzi piksele (początkową wartość piksela jest [0,0], będziemy przechodzić po pikselach od lewego górnego rogu w prawo i w dół).

Tablica pikseli				
P[0, 0]	P[1, 0]	P[2, 0]	...	P[w-1, 0]
P[0, 1]	P[1, 1]	P[2, 1]	...	P[w-1, 1]
...
P[0, h-2]	P[1, h-2]	P[2, h-2]	...	P[w-1, h-2]
P[0, h-1]	P[1, h-1]	P[2, h-1]	...	P[w-1, h-1]

Paleta kolorów:

Przechowuje dane o kolorach pikseli, które są adresowane od indeksu 0 do indeksu 15 (16 kolorów).

Indeks	R	G	B	A
0	R_0	G_0	B_0	-
1	R_1	G_1	B_1	-
2	R_2	G_2	B_2	-
...
15	R_{15}	G_{15}	B_{15}	-

Do obliczenia iluminacji piksela w 16 stopniowej skali szarości używamy wzoru:

$$Y = 0,299 * R + 0,587 * G + 0,114 * B$$

Paleta kolorów (skala szarości):

Indeks	Y
0	Y_0
1	Y_1
2	Y_2
...	...
15	Y_{15}

Alfabet wejściowy:

Naszym alfabetem wejściowym jest wartość 4 bitowa, tzn. Iż możemy zakodować na niej 16 barw, które w postaci symboli czterobitowych będą przechowywane w pamięci naszego komputera. Dla wersji 16-kolorowej na czerwoną i niebieską składową przeznaczamy po 1 bicie, a na zieloną 2. Dzięki czemu uzyskamy następującą paletę barw narzuconych.



Słownik:
















Kolor	RGB	Kod
	[0, 0, 0]	0000
	[0, 0, 255]	1000
	[0, 107, 0]	1100
	[0, 97, 255]	0100
	[0, 190, 0]	0110
	[0, 185, 255]	1110
	[0, 255, 0]	1010
	[0, 255, 255]	0010
	[255, 0, 0]	0011
	[255, 0, 255]	1011
	[255, 93, 0]	1111
	[255, 82, 255]	0111
	[255, 183, 0]	0101
	[255, 179, 255]	1101
	[255, 255, 0]	1001
	[255, 255, 255]	0001

Tabela kolorów dla **skali szarości** jest wyznaczana w następujący sposób:

1. Wyodrębniamy składowe **R(RED)** , **G(GREEN)** oraz **B(BLUE)** piksela,
2. Wyznaczamy wartość **BW** na podstawie wzoru: $BW = 0,299 * R + 0,587 * G + 0,114 * B$
3. Zamieniamy wartość dziesiętną **BW** na binarną,
4. Odczytujemy 5 najbardziej znaczących bitów składowej **BW**,
5. Na podstawie punktu 4 składamy wartość 5-bitową **BW**,

Zapisujemy wartość **BW** w tabeli kolorów, gdzie wartość dziesiętna **BW** stanowi indeks, a **R = G = B = BW**.

Tabela kolorów dla **palety narzuconej** jest wyznaczana w następujący sposób:

1. Wyodrębniamy jedną ze składowych piksela,
2. Zamieniamy wartość dziesiętną składowej na binarną,
3. Odczytujemy najbardziej znaczące bity składowej.

Powyższe kroki wykonujemy dla składowych: **R(RED)** - gdzie wyodrębniamy 1 najbardziej znaczący bit, **G(GREEN)** – gdzie wyodrębniamy 2 najbardziej znaczące bity, **B(BLUE)** – gdzie wyodrębniamy 1 najbardziej znaczący bit. Następnie na ich podstawie składamy wartość 4-bitową **RRGB** i zapisujemy w tabeli kolorów, gdzie wartość dziesiętna **RRGB** stanowi indeks, a **R**, **G** i **B** to kolejne wartości składowych.

Tabela kolorów dla **palety dedykowanej** jest wyznaczana na podstawie algorytmu **Median Cut**.

Algorytm kompresji

Kompresja LZW

Kompresja LZW jest metodą bezstratnej kompresji słownikowej. Metoda ta wykorzystuje fakt, że w grafice występują powtarzające się ciągi punktów. Algorytm ten oparty jest nie tylko na powtarzalności pojedynczych elementów o takiej samej barwie (informacja nadmiarowa), ale potrafi także „zapamiętywać” różne połączenia znaków. Jest to możliwe dzięki temu, że w trakcie kompresji tworzy się słownik takich połączeń. Interesujące jest to, że słownik nie musi być przesłany wraz ze skompresowanym plikiem, bowiem podczas dekompresji dekodery potrafią stworzyć taki słownik w trakcie dekompresowania informacji. Powtarzające się znaki czy też zbitki znaków zastępowane są przez kody liczbowe i zapisywane w tablicy kodowej.

Alfabet kompresora – zestaw wszystkich znaków, jakie mogą się pojawić w danych nieskompresowanych. W określonym zastosowaniu kompresora alfabet jest zwykle stały i znany z góry.

Słownik kompresora – jest to struktura zawierająca części wiadomości wejściowej oraz przypisane im kody na wyjściu kompresora. Słownik powstaje zwykle dynamicznie w trakcie kodowania wiadomości (nie jest stały i znany z góry), zatem zależy on od kompresowanej treści. W zależności od algorytmu kompresji słownik jest bądź też nie jest przesyłany w skompresowanej wiadomości.

Przypadek kompresji LZW:

Kompresja LZW nie wymaga przesłania słownika wraz ze skompresowaną wiadomością, gdyż dynamicznie jest w stanie odtworzyć słownik w trakcie dekompresji. Po określeniu ile bitów zajmą zakodowane wszystkie symbole z alfabetu cały alfabet jest wstawiany do słownika. Zarówno kompresor jak i dekompresor „znają” alfabet, dlatego też nie ma potrzeby przesyłania go do dekompresji. Pozostała część słownika zawierająca symbole złożone jest odtwarzana przez dekodery na bieżąco, także też nie ma potrzeby jej przesyłania. Zatem słownik w ogóle nie musi być przesłany do procesu dekompresji.