# Projekt 3. Rafał Szczepanik 293471 1AR.

Spis treści

[Projekt 3. Rafał Szczepanik 293471 1AR. 1](#_Toc503456480)

[Temat projektu: 1](#_Toc503456481)

[Opis zastosowanych rozwiązań: 1](#_Toc503456482)

[Algorytmy filtrów: 1](#_Toc503456483)

[Algorytm negatywu: 1](#_Toc503456484)

[Algorytm szarości: 1](#_Toc503456485)

[Schemat blokowy programu: 2](#_Toc503456486)

[Struktury i zmienne globalne : 3](#_Toc503456487)

[Funkcje użyte w programie: 3](#_Toc503456488)

[Biblioteki użyte w programie: 4](#_Toc503456489)

## Temat projektu:

Napisać program do obróbki cyfrowej bitmap. Program powinien generować negatyw, obraz w skali szarości, filtr uśredniający, medianowy, maksymalny i minimalny o dowolnym rozmiarze maski. Obsługiwane typy bitmap to 8, 24 i 32 bitowe.

## Opis zastosowanych rozwiązań:

Działanie programu sprowadza się do wczytania nagłówka pliku i obrazu oraz odczytania z niego ile bitów opisuje jeden piksel. Następnie należy wpisać wartości opisujące kolory jednego piksela do dynamicznej dwuwymiarowej tablicy struktur (lub liczb całkowitych w przypadku 8bitowej mapy). Cyfrowa obróbka sprowadza się do operowania na wartościach (RGB w przypadku map 24 i 32 bitowej lub numerach indeksów w palecie kolorów w przypadku mapy 8 bitowej). Jedynie w przypadku konwersji obrazu 8 bitowego na skalę szarości operacja zostaje wykonana na palecie kolorów, a nie na indeksach pikseli.

### Algorytmy filtrów:

Filtry uzyskuje się poprzez przypisanie pikselowi centralnemu wartości która jest funkcją wartości pikseli sąsiadujących. Rozmiar maski określa ile pikseli jest branych jako argumenty. Przykładowo maska o rozmiarze 3x3 bierze pod uwagę wartości 9 pikseli.

Filtr uśredniający przypisuje centralnemu pikselowi wartość średnią pikseli, które go otaczają; filtr minimalny – najmniejszą z wartości; filtr maksymalny – największą z wartości; filtr medianowy – wartość mediany. W przypadku map 24 i 32 bitowych oddzielnie wylicza się wartość każdej ze składowych R, G, B. W przypadku mapy 8 bitowej operuje się na indeksach kolorów w palecie kolorów.

### Algorytm negatywu:

Każdej wartości (R, G, B w przypadku 24 i 32bitowej lub indeksowi w przypadku 8 bitowej) zostaje przypisana wartość dopełnienia do 255. (Np. wartości 55 zostanie przypisana wartość 200.)

### Algorytm szarości:

Wartości każdej ze składowej RGB zostaje przypisana wartość: 0,229\*R+0,587\*G+0,114\*B. W przypadku 24 i 32bitowej mapy wartości te zostają przypisane bezpośrednio pikselom, w przypadku 8 bitowej mapy te wartości przypisuje się składowym kolorów w palecie kolorów.

## Schemat blokowy programu:

## Struktury i zmienne globalne :

|  |  |
| --- | --- |
| struct FileHeader{  short bfType;  int bfSize;  short bfReserved1;  short bfReserved2;  short bfOffBits;} | struktura danych nagłówka pliku  typ pliku  rozmiar pliku w bajtach  dwie wartości zarezerwowane  zmienna – pozycja danych obrazkowych |
| struct RGBQUAD{  short rgbBlue;  short rgbGreen;  short rgbRed;  short rgbReserved;} | struktura – komórka palety kolorów  wartość barwy niebieskiej  wartość barwy zielonej  wartość barwy czerwonej  wartość zarezerwowana |
| struct PictureHeader{  int biSize;  int biWidth;  int biHeight;  short biPlanes;  short biBitCount;  int biCompression;  int biSizeImage;  int biXPelsPerMeter;  int biYPelsPerMeter;  int biClrUsed;  int biClrImportant;} | struktura danych nagłówka obrazu (DIB)  wielkość nagłówka informacyjnego  szerokość obrazu w pikselach  wysokość obrazu w pikselach  liczba warstw kolorów  liczba bitów na piksel  algorytm kompresji  rozmiar samego rysunku  rozdzielczość pozioma  rozdzielczość pionowa  liczba kolorów w palecie  liczba ważnych kolorów w palecie |
| struct KoloryRGB{  short R;  short G;  short B;  short A;} | struktura – piksel mapy 24/32 bitowej  wartość barwy niebieskiej  wartość barwy zielonej  wartość barwy czerwonej  wartość alfa |
| short padding; | wartość marginesu (dopisania do wielokrotności 4) |
| PictureHeader Picture; | dane nagłówka bieżącego obrazu |
| FileHeader File; | dane nagłówka bieżącego pliku |
| RGBQUAD pallette[256]; | paleta kolorów, używana przy obróbce obrazu 8 bitowego |

## Funkcje użyte w programie:

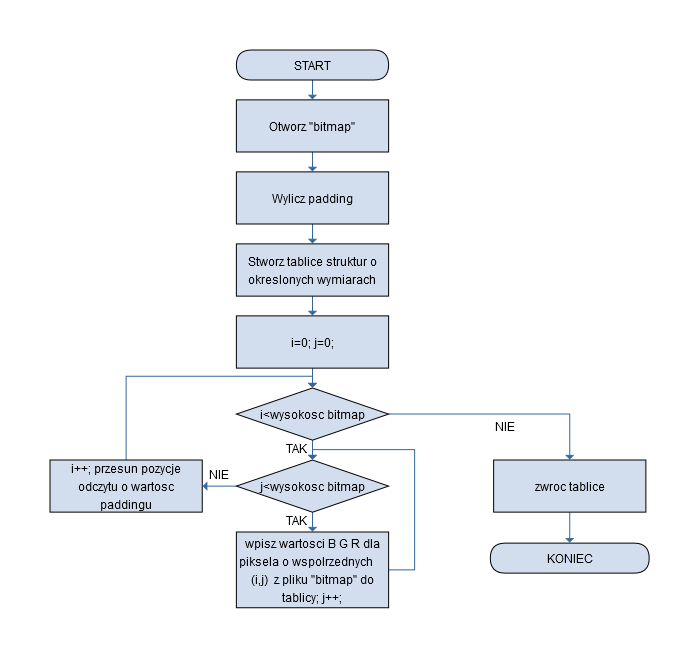
|  |  |
| --- | --- |
| **int compare( const int \* arg1, const int \* arg2 )** | funkcja podawana jako argument qsort, używanego w funkcji mediana |
| **int mediana(int liczby[],int rozmiar)** | funkcja zwracająca medianę nieposortowanej tablicy integer; jako parametry przyjmuje tablicę integer oraz jej rozmiar używana do wyznaczenia mediany wartości kolorów poszczególnych pikseli w filtrze medianowym |
| **int is\_a\_bmp(char\* file\_name)** | funkcja sprawdza czy podany plik jest bitmapa: 1) czy rozszerzenie to .bmp 2) czy zgadza sie typ pliku (pierwsze bity naglowka); jako parametr przyjmuje nazwę pliku; użyta jest do sprawdzenia poprawności wprowadzonych danych do programu |
| **void\*\* create\_pixmap(int bits,int width, int height)** | funkcja zwracająca wskaźnik na dynamiczna, dwuwymiarową tablicę przygotowana do zapisu bitmapy 8,24 lub 32 bitowej; jako parametry przyjmuje: liczbę bitów na piksel, szerokość i wysokość bitmapy; używana we wszystkich funkcjach filtrujących bitmapę. |
| **short\*\* read\_Bits8(char \*bitmap,FileHeader File)** | funkcja odczytuje mapę 8 bitowa - zapisuje ja w dwuwymiarowej tablicy dynamicznej, która jest przez nią zwracana; jako parametry przyjmuje nazwę pliku, z którego ma być dokonywany odczyt oraz strukturę danych nagłówka pliku |
| **KoloryRGB\*\* read\_Bits24(char \*bitmap,FileHeader File)** | funkcja odczytuje mapę 24 i 32 bitową - zapisuje ja w dwuwymiarowej tablicy dynamicznej, która jest przez nią zwracana; jako parametry przyjmuje nazwę pliku, z którego ma być dokonywany odczyt oraz strukturę danych nagłówka pliku |
| **void copy\_Header8Sz(char \*bitmap,char\* nowy)** | funkcja kopiuje nagłówek pliku, nagłówek obrazu bez palety kolorów z bitmapy o nazwie podanej jako pierwszy argument do pliku o nazwie podanej jako drugi argument; używana do tworzenia plików z danymi 8 bitowej mapy po konwersji na skalę szarości |

|  |  |
| --- | --- |
| **void copy\_Header(char \*bitmap,char\* nowy)** | funkcja kopiuje nagłówek pliku, nagłówek obrazu oraz paletę kolorów z bitmapy o nazwie podanej jako pierwszy argument do pliku o nazwie podanej jako drugi argument; używana do tworzenia plików z danymi po konwersji przez dowolny filtr |
| **void write\_Bits8(short\*\* pixmap, char\* nowy)** | funkcja zapisuje tablice utworzona przez funkcje read\_Bits8, podana jako pierwszy argument do pliku o nazwie podanej jako drugi argument |
| **write\_Bits24(KoloryRGB\*\* pixmap, char\* nowy)** | funkcja zapisuje tablice utworzona przez funkcje read\_Bits24, podana jako pierwszy argument do pliku o nazwie podanej jako drugi argument |
| **short\*\* negatyw8(short \*\*pixmap)** | funkcja zwraca bitmapę 8 bitową która jest bitmapą przyjętą jako parametr funkcji przerobioną na negatyw |
| **short\*\* filtrUs8(short \*\*pixmap,int s);**  **short\*\* filtrMax8(short \*\*pixmap,int s);**  **short\*\* filtrMin8(short \*\*pixmap,int s);**  **short\*\* filtrMed8(short \*\*pixmap,int s);** | te funkcje zwracają bitmapy 8 bitowe, które są bitmapami przyjętymi jako pierwszy argument tych funkcji przerobione odpowiednio filtrem: uśredniającym, maksymalnym, minimalnym, medianowym o rozmiarze maski s x s – przyjętym jako drugi argument funkcji |
| **void szary8(char \*nowy);** | funkcja wpisuje paletę kolorów zamienioną na skalę szarości do pliku o nazwie przyjętej jako argument funkcji |
| **KoloryRGB\*\* szary24(KoloryRGB \*\*pixmap)**  **KoloryRGB\*\* negatyw24(KoloryRGB \*\*pixmap)** | funkcje zwracają bitmapy 24 lub 32 bitowe, które są bitmapami przyjętymi jako parametr funkcji przerobione odpowiednio na negatyw i skalę szarości |
| **KoloryRGB\*\* filtrMax24(KoloryRGB \*\*pixmap,int s)**  **KoloryRGB\*\* filtrMin24(KoloryRGB \*\*pixmap,int s)**  **KoloryRGB\*\* filtrMed24(KoloryRGB \*\*pixmap,int s)**  **KoloryRGB\*\* filtrUs24(KoloryRGB \*\*pixmap,int s)** | te funkcje zwracają bitmapy 24 lub 32 bitowe, które są bitmapami przyjętymi jako pierwszy argument tych funkcji przerobione odpowiednio filtrem: maksymalnym, minimalnym, medianowym  i uśredniającym o rozmiarze maski s x s – przyjętym jako drugi argument funkcji |

## Biblioteki użyte w programie:

Oprócz standardowych bibliotek (<stdio.h>,<stdlib.h>) w programie użyto bibliotek: <string.h>, w celu użycia funkcji: strstr(), strcmp().

**Schemat funkcji:** **KoloryRGB\*\* read\_Bits24(char \*bitmap,FileHeader File)**



**Schemat funkcji:** **short\*\* negatyw8(short \*\*pixmap)**

