

Median Filter

Zuzanna Santorowska

2 stycznia 2020

Treść zadania

Program, który wczyta zdjęcie w formacie BMP, zastosuje do niego filtr medianowy i zwróci zdjęcie po filtracji. Należy dobrać odpowiednio zaszumione zdjęcie, by przetestować działanie filtra medianowego.

Sposób działania algorytmu

Program wczytuje bitmapę po czym tworząc wynik sam na sobie wylicza medianę z kwadratów 3×3 piksele i wpisuje wynik do środkowego piksela. Ten typ filtra jest szczególnie użyteczny w wypadku zabrudzeń typu pieprz sól.



Rysunek 1: Obraz róży przed zastosowaniem filtra medianowego



Rysunek 2: Obraz róży po zastosowaniu filtra medianowego

Wykaz i opis funkcji

1. main

- Zastosowanie: Główna funkcja która wywołuje tworzenie struktur do przechowywania danych, wywołuje wczytywanie danych, filtr medianowy a następnie wyrzuca dane do pliku.
- Wartości zwracane: return 0 kończący program
- Parametry wywołania: brak

2. GetDataFromFile

- Zastosowanie: Funkcja czytająca dane z pliku i umieszczająca je w odpowiednich, wcześniej przygotowanych strukturach.
- Wartości zwracane: `Pixel* arr (4)` - wskaźnik do struktury danych dotyczących obrazka.
- Parametry wywołania: `int* column` - przekazywany jest wskaźnik do int w którym będzie znajdować się informacja o ilości kolumn w tablicy danych.
`int* row` - przekazywany jest wskaźnik do int w którym będzie znajdować się informacja o ilości wierszy w tablicy danych.
`FileHeader *fileHeader (1)` - przekazywany jest wskaźnik do struktury FileHeader w którym będzie znajdować się informacja danych nagłówkowych pliku.

`PictureHeader *pictureHeader` (2) - przekazywany jest wskaźnik do struktury `PictureHeader` w którym będzie znajdować się informacja danych nagłówkowych obrazka.

3. MedianFilter

- Zastosowanie: Tworzy wynikową strukturę danych po czym wypełnia ją w podwójnej pętli dedykując zadanie znalezienia mediany 3x3 do odpowiedniej funkcji.
- Wartości zwracane: `Pixel* arrOut` - wskaźnik do struktury danych dotyczących obrazka.
- Parametry wywołania: `int column` - przekazywany jest int w którym znajduje się informacja o ilości kolumn w tablicy danych.
`int row` - przekazywany jest int w którym znajduje się informacja o ilości wierszy w tablicy danych.
`Pixel *arr` - przekazywany jest wskaźnik do struktury zawierającej dane o poszczególnych pikselach.

4. Median3x3

- Zastosowanie: Na podstawie dużej tablicy tworzy tablicę lokalną zawierającą pixele o takich samych wartościach jak tablica macierzysta po czym przekazuje ją do funkcji wyliczającej medianę.
- Wartości zwracane: `Pixel p` - Wartość pixela po zastosowaniu na nim mediany.
- Parametry wywołania: `int row` - przekazywany jest int w którym znajduje się informacja o ilości wierszy w tablicy danych.
`Pixel *a` - przekazywany jest wskaźnik do struktury zawierającej dane o poszczególnych pikselach.
`int i` - parametr określający w którym miejscu dużej tablicy znajduje się centrum kwadratu 3x3.

5. Median

- Zastosowanie: Rozdziela otrzymaną tablicę na kolory po czym za pomocą funkcji sortuje je i zwraca pixel którego każdy z kolorów jest medianą.
- Wartości zwracane: `Pixel p` - Wartość pixela po zastosowaniu na nim mediany.

- Parametry wywołania: `Pixel *med` - przekazywany jest wskaźnik do struktury zawierającej dane obecnie przetwarzanych pixelach.

6. Median

- Zastosowanie: Rozdziela otrzymaną tablicę na kolory po czym za pomocą funkcji sortuje je i zwraca pixel którego każdy z kolorów jest medianą.
- Wartości zwracane: `Pixel p` - Wartość pixela po zastosowaniu na nim mediany.
- Parametry wywołania: `Pixel *med` - przekazywany jest wskaźnik do struktury zawierającej dane obecnie przetwarzanych pixelach.

7. Sort(int* arr)

- Zastosowanie: Sortuje zadaną tablicę.
- Wartości zwracane: `void`
- Parametry wywołania: `Pixel *arr` - przekazywany jest wskaźnik do struktury zawierającej dane obecnie przetwarzanych pixelach.

8. SaveBitmap(Pixel *arrOut, int column, int row, FileHeader* fileHeader, PictureHeader* pictureHeader)

- Zastosowanie: Funkcja zapisuje dane do pliku używając do tego odpowiednich, wcześniej przygotowanych struktur.
- Wartości zwracane: `void`
- Parametry wywołania: `int column` - przekazywany jest int w którym znajduje się informacja o ilości kolumn w tablicy danych.
`int row` - przekazywany jest int w którym znajduje się informacja o ilości wierszy w tablicy danych.
`FileHeader *fileHeader` - przekazywany jest wskaźnik do struktury `FileHeader` w którym znajduje się informacja o danych nagłówkowych pliku.
`PictureHeader *pictureHeader` - przekazywany jest wskaźnik do struktury `PictureHeader` w którym będzie znajduje się informacja o danych nagłówkowych obrazka.
`Pixel *arrOut` - przekazywany jest wskaźnik do struktury zawierającej dane o poszczególnych pikselach wyniku.

Listing 1: Struktura FileHeader przechowuje dane nagłówka pliku.

```
typedef struct FileHeader {  
    short bfType;  
    int bfSize;  
    short bfReserved1;  
    short bfReserved2;  
    short bfOffBits;  
}FileHeader;
```

Listing 2: Struktura PictureHeader przechowuje dane nagłówka obrazu.

```
typedef struct PictureHeader {  
    int biSize;  
    int biWidth;  
    int biHeight;  
    short biPlanes;  
    short biBitCount;  
    int biCompression;  
    int biSizeImage;  
    int biXPelsPerMeter;  
    int biYPelsPerMeter;  
    int biClrUsed;  
    int biClrImportant;  
}PictureHeader;
```

Listing 3: Struktura KoloryRGB służy oznaczeniu kolorów.

```
typedef struct KoloryRGB {  
    char R;  
    char G;  
    char B;  
}KoloryRGB;
```

Listing 4: Struktura Pixel służy przechowywaniu danych dotyczących jednego pixela.

```
typedef struct Pixel {  
    int red;  
    int green;  
    int blue;  
} Pixel;
```