Operacje morfologiczne na obrazie

0.97

Wygenerowano przez Doxygen 1.8.16

Rozdział 1

Indeks struktur danych

1.1 Struktury danych

Tutaj znajdują się struktury danych wraz z ich krótkimi opisami:

Rozdział 2

Indeks plików

2.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

Rozdział 3

Dokumentacja struktur danych

3.1 Dokumentacja struktury BITMAPHEADER

Struktura przechowująca nagłówek DIB.

`#include <bmp.h>`

Pola danych

- uint32_t size

  Wartość ta wskazuje na ilość bajtów, z których składa się drugi podnagłówek pliku mapy bitowej.
- uint32_t width

  Szerokość obrazu.
- uint32_t height

  Wysokość obrazu.
- uint16_t numBitPlanes

  Liczba używanych przez obraz płaszczyzn kolorów.
- uint16_t numBitsPerPlane

  Ilość bitów na piksel.
- uint32_t comp

  Parametr, wskazujący na typ kompresji (w tym przypadku 0).
- uint32_t image_size
- uint32_t Xres

  Rozdzielczość pozioma.
- uint32_t Yres

  Rozdzielczość pionowa.
- uint32_t colors_used

  Ilość kolorów w palecie.
- uint16_t colors_importance_rotation

  Ilość ważnych kolorów w palecie (w tym przypadku 0; wszystkie kolory są jednokowo ważne).
- uint16_t reserved

  Bajty zarezerwowane przez aplikację.

3.1.1 Opis szczegółowy

Struktura przechowująca nagłówek DIB.

3.1.2 Dokumentacja pól

3.1.2.1 colors_importance_rotation

```
uint16_t colors_importance_rotation
```

Ilość ważnych kolorów w palecie (w tym przypadku 0; wszystkie kolory są jednokowo ważne).

3.1.2.2 colors_used

```
uint32_t colors_used
```

Ilość kolorów w palecie.

3.1.2.3 comp

```
uint32_t comp
```

Parametr, wskazujący na typ kompresji (w tym przypadku 0).

3.1.2.4 height

```
uint32_t height
```

Wysokość obrazu.

3.1.2.5 image_size

```
uint32_t image_size
```

3.1.2.6 numBitPlanes

```
uint16_t numBitPlanes
```

Liczba używanych przez obraz płaszczyzn kolorów.

3.1.2.7 numBitsPerPlane

```
uint16_t numBitsPerPlane
```

Ilość bitów na piksel.

3.1.2.8 reserved

```
uint16_t reserved
```

Bajty zarezerwowane przez aplikację.

3.1.2.9 size

```
uint32_t size
```

Wartość ta wskazuje na ilość bajtów, z których składa się drugi podnagłówek pliku mapy bitowej.

3.1.2.10 width

```
uint32_t width
```

Szerokość obrazu.

3.1.2.11 Xres

```
uint32_t Xres
```

Rozdzielczość pozioma.

3.1.2.12 Yres

```
uint32_t Yres
```

Rozdzielczość pionowa.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- bmp.h

3.2 Dokumentacja struktury BMPFILEHEADER

Pierwsza w kolejności podstruktura struktury nagłówkowej pliku mapy bitowej.

```
#include <bmp.h>
```

Pola danych

- uint8_t s1
  Pierwszy znak typu pliku w kodzie ASCII (powinien wskazywać 'M').
- uint8_t s2
  Drugi znak typu pliku w kodzie ASCII (powinien wskazywać 'M').
- uint32_t size
  Rozmiar pliku obrazu mapy bitowej (powinno być 70 bajtów).
- uint16_t r1
  Bajt zarezerwowany dla aplikacji.
- uint16_t r2
  Bajt zarezerwowany dla aplikacji.
- uint32_t offset
  Numer bajtu, który rozpoczyna macierz pikseli.

3.2.1 Opis szczegółowy

Pierwsza w kolejności podstruktura struktury nagłówkowej pliku mapy bitowej.

3.2.2 Dokumentacja pól

3.2.2.1 offset

```
uint32_t offset
```

Numer bajtu, który rozpoczyna macierz pikseli.

3.2.2.2 r1

```
uint16_t r1
```

Bajt zarezerwowany dla aplikacji.

3.2.2.3 r2

```
uint16_t r2
```

Bajt zarezerwowany dla aplikacji.

3.2.2.4 s1

```
uint8_t s1
```

Pierwszy znak typu pliku w kodzie ASCII (powinien wskazywać 'M').

3.2.2.5 s2

```
uint8_t s2
```

Drugi znak typu pliku w kodzie ASCII (powinien wskazywać 'M').

3.2.2.6 size

```
uint32_t size
```

Rozmiar pliku obrazu mapy bitowej (powinno być 70 bajtów).

Dokumentacja dla tej struktury została wygenerowana z pliku:

- bmp.h

3.3 Dokumentacja struktury PIXEL

Struktura przechowująca poszczególne składowe kolorów pikseli.

```
#include <bmp.h>
```

Pola danych

- uint8_t blue

  Składowa niebieska koloru piksela.
- uint8_t green

  Składowa zielona koloru piksela.
- uint8_t red

  Składowa czerwona koloru piksela.

3.3.1 Opis szczegółowy

Struktura przechowująca poszczególne składowe kolorów pikseli.

3.3.2 Dokumentacja pól

3.3.2.1 blue

```
uint8_t blue
```

Składowa niebieska koloru piksela.

3.3.2.2 green

```
uint8_t green
```

Składowa zielona koloru piksela.

3.3.2.3 red

```
uint8_t red
```

Składowa czerwona koloru piksela.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- bmp.h

3.4 Dokumentacja struktury pixel_structure

Struktura listy jednokierunkowej elementu strukturalnego. Składa się ona ze wskaźnika na następny element elementu strukturalnego, który zawiera znak. Ta struktura jest synchronizowana z siatką pikselu obrazu za pomocą odpowiednich funkcji.

`#include <bmp.h>`

Pola danych

- pixel_structure * next

  Wskaźnik na następny element (znak) elementu strukturalnego.
- uint8_t type

  Znak, który zawiera dany element elementu strukturalnego.

3.4.1 Opis szczegółowy

Struktura listy jednokierunkowej elementu strukturalnego. Składa się ona ze wskaźnika na następny element elementu strukturalnego, który zawiera znak. Ta struktura jest synchronizowana z siatką pikselu obrazu za pomocą odpowiednich funkcji.

3.4.2 Dokumentacja pól

3.4.2.1 next

`pixel_structure* next`

Wskaźnik na następny element (znak) elementu strukturalnego.

3.4.2.2 type

`uint8_t type`

Znak, który zawiera dany element elementu strukturalnego.

Dokumentacja dla tej struktury została wygenerowana z pliku:

- bmp.h

Rozdział 4

Dokumentacja plików

4.1 Dokumentacja pliku bmp.h

Plik nagłówkowy zawierający wszelkie struktury, potrzebne do działania programu.

```
#include <inttypes.h>
```

Struktury danych

- struct BMPFILEHEADER

  Pierwsza w kolejności podstruktura struktury nagłówkowej pliku mapy bitowej.
- struct BITMAPHEADER

  Struktura przechowująca nagłówek DIB.
- struct PIXEL

  Struktura przechowująca poszczególne składowe kolorów pikseli.
- struct pixel_structure

  Struktura listy jednokierunkowej elementu strukturalnego. Składa się ona ze wskaźnika na następny element elementu strukturalnego, który zawiera znak. Ta struktura jest synchronizowana z siatką pikselu obrazu za pomocą odpowiednich funkcji.

Definicje typów

- typedef struct BMPFILEHEADER BMPFILEHEADER

  Pierwsza w kolejności podstruktura struktury nagłówkowej pliku mapy bitowej.
- typedef struct BITMAPHEADER BITMAPHEADER

  Struktura przechowująca nagłówek DIB.
- typedef struct PIXEL PIXEL

  Struktura przechowująca poszczególne składowe kolorów pikseli.
- typedef struct pixel_structure pixel_structure

  Struktura listy jednokierunkowej elementu strukturalnego. Składa się ona ze wskaźnika na następny element elementu strukturalnego, który zawiera znak. Ta struktura jest synchronizowana z siatką pikselu obrazu za pomocą odpowiednich funkcji.

4.1.1 Opis szczegółowy

Plik nagłówkowy zawierający wszelkie struktury, potrzebne do działania programu.

Autor

Aleksander Augustyniak

4.1.2 Dokumentacja definicji typów

4.1.2.1 BITMAPHEADER

```
typedef struct BITMAPHEADER BITMAPHEADER
```

Struktura przechowująca nagłówek DIB.

4.1.2.2 BMPFILEHEADER

```
typedef struct BMPFILEHEADER BMPFILEHEADER
```

Pierwsza w kolejności podstruktura struktury nagłówkowej pliku mapy bitowej.

4.1.2.3 PIXEL

```
typedef struct PIXEL PIXEL
```

Struktura przechowująca poszczególne składowe kolorów pikseli.

4.1.2.4 pixel_structure

```
typedef struct pixel_structure pixel_structure
```

Struktura listy jednokierunkowej elementu strukturalnego. Składa się ona ze wskaźnika na następny element elementu strukturalnego, który zawiera znak. Ta struktura jest synchronizowana z siatką pikselu obrazu za pomocą odpowiednich funkcji.

4.2 Dokumentacja pliku functions.c

Plik nagłówkowy zawierający definicje funkcji.

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <inttypes.h>
#include "bmp.h"
#include "functions.h"
```

Funkcje

- uint8_t *** AllocateInputMemory (const uint32_t *imgHeight, const uint32_t *imgWidth)

  Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów (wysokość x szerokość x 3 składowe).
- uint8_t *** DeallocateInputMemory (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *img↩ Width)

  Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateInputMemory.
- void ReadPixels (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *imgWidth, const FILE *input_file, const uint32_t *offset_byte)
- uint8_t *** AllocateOutputMemory (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *img↩ Width)

  Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów dla pliku wyjściowego.
- uint8_t *** DeallocateOutputMemory (uint8_t ***image_out, const uint8_t *imgHeight, const uint8_t *img↩ Width)

  Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateOutputMemory.
- void SaveBMPFile (const uint32_t *imgHeight, const uint32_t *imgWidth, const uint8_t ***image_out, const FILE *output_file)

  Funkcja zapisująca bajty składowych kolorów pikseli do pliku wyjściowego.
- void ReadToStructure (const BMPFILEHEADER *FileHeader, const BITMAPHEADER *Header, const FILE *input_file)

  Funkcja zapisująca do struktury dane nagłówka pliku, zawierającego informacje nt. obrazu, m.in. rozdzielczość, wysokość, szerokość, waga pliku, etc.
- void WriteFromStructure (const BMPFILEHEADER *FileHeader, const BITMAPHEADER *Header, const F↩ ILE *output_file)

  Funkcja zapisująca do pliku wyjściowego pozyskane przez funkcję ReadToStructure dane, zapisane do struktury.
- uint8_t * change_pixel_color (uint8_t *pixel, uint8_t color)

  Funkcja zamienia kolor danego przez adres piksela.
- uint8_t check_pixel_color (uint8_t *pixel)

  Funkcja sprawdza jakiego koloru jest podany piksel.
- void operation (const uint8_t type, uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

  Główny algorytm, który odpowiada za wykonanie operacji morfologicznych na obrazie. Funkcja dopasowuje strukturę sita do piksela modyfikowanego tak, by można było sprawdzić warunki zamiany piksela modyfikowanego.
- void opening (uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

  Funkcja otwarcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu usunięcia szczegółow z obrazu (erozja -> dylatacja).
- void closing (uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

Funkcja zamknięcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu "zamalowania" niewypełnionych do końca szczelin (dylatacja-> erozja). Wymagany efekt ściśle zależy of elementu strukturalnego.

- pixel_structure * load_sieve (FILE *stream)

  Funkcja wczytująca do specjalnej struktury element strukturalny.
- uint32_t * main_point (pixel_structure *sieve)

  Algorytm wykrywający współrzędne piksela modyfikowanego (głównego) elementu strukturalnego.
- void deallocate_sieve (pixel_structure *sieve)

  Funkcja dealokująca pamięć, potrzebną by wczytać element strukturalny.

4.2.1 Opis szczegółowy

Plik nagłówkowy zawierający definicje funkcji.

Autor

Aleksander Augustyniak

4.2.2 Dokumentacja funkcji

4.2.2.1 AllocateInputMemory()

```
uint8_t*** AllocateInputMemory (
             const uint32_t * imgHeight,
             const uint32_t * imgWidth )
```

Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów (wysokość x szerokość x 3 składowe).

Parametry

imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

wskaźnik do adresu schowka na dane wejściowe lub NULL.

4.2.2.2 AllocateOutputMemory()

```
uint8_t*** AllocateOutputMemory (
             uint8_t *** image_in,
             const uint32_t * imgHeight,
             const uint32_t * imgWidth )
```

Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów dla pliku wyjściowego.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

wskaźnik do adresu schowka na dane wyjściowe lub NULL.

4.2.2.3 change_pixel_color()

```
uint8_t* change_pixel_color (
            uint8_t * pixel,
            uint8_t color )
```

Funkcja zamienia kolor danego przez adres piksela.

Parametry

pixel	wskaźnik do piksela edytowanego.
color	współczynnik mówiący o tym, na jaki kolor ma być zamieniony dany piksel (1: biały / 0: czarny).

Zwraca

wskaźnik na piksel zmodyfikowany.

4.2.2.4 check_pixel_color()

```
uint8_t check_pixel_color (
            uint8_t * pixel )
```

Funkcja sprawdza jakiego koloru jest podany piksel.

Parametry

pixel	wskaźnik do piksela, którego poszczególne składowego koloru mają być sprawdzone.

Zwraca

0 jeżeli wszystkie składowe koloru piksela są równe zero. W przeciwnym wypadku 1.

4.2.2.5 closing()

```
void closing (
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Funkcja zamknięcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu "zamalowania" niewypełnionych do końca szczelin (dylatacja-> erozja). Wymagany efekt ściśle zależy of elementu strukturalnego.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.2.2.6 deallocate_sieve()

```
void deallocate_sieve (
            pixel_structure * sieve )
```

Funkcja dealokująca pamięć, potrzebną by wczytać element strukturalny.

Parametry

sieve	wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

4.2.2.7 DeallocateInputMemory()

```
uint8_t*** DeallocateInputMemory (
            uint8_t *** image_in,
            const uint32_t * imgHeight,
            const uint32_t * imgWidth )
```

Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateInputMemory.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

NULL.

4.2.2.8 DeallocateOutputMemory()

```
uint8_t*** DeallocateOutputMemory (
            uint8_t *** image_out,
            const uint8_t * imgHeight,
            const uint8_t * imgWidth )
```

Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateOutputMemory.

Parametry

image_out	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

NULL.

4.2.2.9 load_sieve()

```
pixel_structure* load_sieve (
            FILE * stream )
```

Funkcja wczytująca do specjalnej struktury element strukturalny.

Parametry

stream	wskaźnik do pliku, który zawiera element strukturalny.

Zwraca

wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

4.2.2.10 main_point()

```
uint32_t* main_point (
            pixel_structure * sieve )
```

Algorytm wykrywający współrzędne piksela modyfikowanego (głównego) elementu strukturalnego.

Parametry

sieve	wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

Zwraca

wskaźnik na tablicę współrzędnych punktu głównego elementu strukturalnego.

4.2.2.11 opening()

```
void opening (
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Funkcja otwarcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu usunięcia szczegółow z obrazu (erozja -> dylatacja).

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.2.2.12 operation()

```
void operation (
            const uint8_t type,
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Główny algorytm, który odpowiada za wykonanie operacji morfologicznych na obrazie. Funkcja dopasowuje strukturę sita do piksela modyfikowanego tak, by można było sprawdzić warunki zamiany piksela modyfikowanego.

Parametry

type	znak warunkujący operację, która ma być wykonana przez funkcję ('e' – erozja; 'd' – dylatacja).

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
image_out	wskaźnik do początku tablicy na dane wyjściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.2.2.13 ReadPixels()

```
void ReadPixels (
            uint8_t *** image_in,
            const uint32_t * imgHeight,
            const uint32_t * imgWidth,
            const FILE * input_file,
            const uint32_t * offset_byte )
```

4.2.2.14 ReadToStructure()

```
void ReadToStructure (
            const BMPFILEHEADER * FileHeader,
            const BITMAPHEADER * Header,
            const FILE * input_file )
```

Funkcja zapisująca do struktury dane nagłówka pliku, zawierającego informacje nt. obrazu, m.in. rozdzielczość, wysokość, szerokość, waga pliku, etc.

Parametry

FileHeader	wskaźnik na podstrukturę mówiąca o tym, czy plik rzeczywiście jest plikiem mapy bitowej. Podstruktura mówi m.in. od którego bajtu zaczynają się piksele.
Header	wskaźnik na podstrukturę, której dane opisują wymiary obrazu. Podstruktura zawiera przede wszystkim informacje techniczne obrazu.
input_file	wskaźnik na plik wejściowy.

4.2.2.15 SaveBMPFile()

```
void SaveBMPFile (
            const uint32_t * imgHeight,
            const uint32_t * imgWidth,
```

```
const uint8_t *** image_out,
const FILE * output_file )
```

Funkcja zapisująca bajty składowych kolorów pikseli do pliku wyjściowego.

Parametry

imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.
image_out	wskaźnik do początku tablicy na dane wyjściowe.
output_file	wskaźnik na plik wyjściowy.

4.2.2.16 WriteFromStructure()

```
void WriteFromStructure (
            const BMPFILEHEADER * FileHeader,
            const BITMAPHEADER * Header,
            const FILE * output_file )
```

Funkcja zapisująca do pliku wyjściowego pozyskane przez funkcję ReadToStructure dane, zapisane do struktury.

Parametry

FileHeader	wskaźnik na podstrukturę mówiąca o tym, czy plik rzeczywiście jest plikiem mapy bitowej. Podstruktura mówi m.in. od którego bajtu zaczynają się piksele.
Header	wskaźnik na podstrukturę, której dane opisują wymiary obrazu. Podstruktura zawiera przede wszystkim informacje techniczne obrazu.
output_file	wskaźnik na plik wyjściowy.

4.3 Dokumentacja pliku functions.h

Plik nagłówkowy zawierający deklaracje funkcji.

Funkcje

- uint8_t *** AllocateInputMemory (const uint32_t *imgHeight, const uint32_t *imgWidth)

  Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów (wysokość x szerokość x 3 składowe).
- uint8_t *** DeallocateInputMemory (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *img↩ Width)

  Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateInputMemory.
- void ReadPixels (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *imgWidth, const FILE *input_file, uint32_t *offset_byte)

  Funkcja sczytująca do struktury poszczególne składowe każdego piksela obrazu wejściowego. Jeżeli którykolwiek ze składowych koloru piksela ma wartość większą od 0 – wszystkie składowe są maksymalizowane.
- uint8_t *** AllocateOutputMemory (uint8_t ***image_in, const uint32_t *imgHeight, const uint32_t *img↩ Width)

  Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów dla pliku wyjściowego.
- uint8_t *** DeallocateOutputMemory (uint8_t ***image_out, const uint8_t *imgHeight, const uint8_t *img↩ Width)

  Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateOutputMemory.

- void SaveBMPFile (const uint32_t *imgHeight, const uint32_t *imgWidth, const uint8_t ***image_out, const FILE *output_file)

  Funkcja zapisująca bajty składowych kolorów pikseli do pliku wyjściowego.
- void ReadToStructure (const BMPFILEHEADER *FileHeader, const BITMAPHEADER *Header, const FILE *input_file)

  Funkcja zapisująca do struktury dane nagłówka pliku, zawierającego informacje nt. obrazu, m.in. rozdzielczość, wysokość, szerokość, waga pliku, etc.
- void WriteFromStructure (const BMPFILEHEADER *FileHeader, const BITMAPHEADER *Header, const FILE *output_file)

  Funkcja zapisująca do pliku wyjściowego pozyskane przez funkcję ReadToStructure dane, zapisane do struktury.
- uint8_t * change_pixel_color (uint8_t *pixel, uint8_t color)

  Funkcja zamienia kolor danego przez adres piksela.
- uint8_t check_pixel_color (uint8_t *pixel)

  Funkcja sprawdza jakiego koloru jest podany piksel.
- void operation (const uint8_t type, uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

  Główny algorytm, który odpowiada za wykonanie operacji morfologicznych na obrazie. Funkcja dopasowuje strukturę sita do piksela modyfikowanego tak, by można było sprawdzić warunki zamiany piksela modyfikowanego.
- void opening (uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

  Funkcja otwarcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu usunięcia szczegółow z obrazu (erozja -> dylatacja).
- void closing (uint8_t ***image_in, uint8_t ***image_out, uint32_t width, uint32_t height, pixel_structure *sieve, uint32_t *coordinates)

  Funkcja zamknięcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu "zamalowania" niewypełnionych do końca szczelin (dylatacja-> erozja). Wymagany efekt ściśle zależy of elementu strukturalnego.
- pixel_structure * load_sieve (FILE *stream)

  Funkcja wczytująca do specjalnej struktury element strukturalny.
- void deallocate_sieve (pixel_structure *sieve)

  Funkcja dealokująca pamięć, potrzebną by wczytać element strukturalny.
- uint32_t * main_point (pixel_structure *sieve)

  Algorytm wykrywający współrzędne piksela modyfikowanego (głównego) elementu strukturalnego.

4.3.1 Opis szczegółowy

Plik nagłówkowy zawierający deklaracje funkcji.

Autor

Aleksander Augustyniak

4.3.2 Dokumentacja funkcji

4.3.2.1 AllocateInputMemory()

```
uint8_t*** AllocateInputMemory (
            const uint32_t * imgHeight,
            const uint32_t * imgWidth )
```

Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów (wysokość x szerokość x 3 składowe).

Parametry

imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

wskaźnik do adresu schowka na dane wejściowe lub NULL.

4.3.2.2 AllocateOutputMemory()

```
uint8_t*** AllocateOutputMemory (
             uint8_t *** image_in,
             const uint32_t * imgHeight,
             const uint32_t * imgWidth )
```

Funkcja alokująca tyle pamięci, by pomieścić cały zakres kolorów dla pliku wyjściowego.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

wskaźnik do adresu schowka na dane wyjściowe lub NULL.

4.3.2.3 change_pixel_color()

```
uint8_t* change_pixel_color (
             uint8_t * pixel,
             uint8_t color )
```

Funkcja zamienia kolor danego przez adres piksela.

Parametry

pixel	wskaźnik do piksela edytowanego.
color	współczynnik mówiący o tym, na jaki kolor ma być zamieniony dany piksel (1: biały / 0: czarny).

Zwraca

wskaźnik na piksel zmodyfikowany.

4.3.2.4 check_pixel_color()

```
uint8_t check_pixel_color (
            uint8_t * pixel )
```

Funkcja sprawdza jakiego koloru jest podany piksel.

Parametry

pixel	wskaźnik do piksela, którego poszczególne składowego koloru mają być sprawdzone.

Zwraca

0 jeżeli wszystkie składowe koloru piksela są równe zero. W przeciwnym wypadku 1.

4.3.2.5 closing()

```
void closing (
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Funkcja zamknięcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu "zamalowania" niewypełnionych do końca szczelin (dylatacja-> erozja). Wymagany efekt ściśle zależy of elementu strukturalnego.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.3.2.6 deallocate_sieve()

```
void deallocate_sieve (
```

```
            pixel_structure * sieve )
```

Funkcja dealokująca pamięć, potrzebną by wczytać element strukturalny.

Parametry

sieve	wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

4.3.2.7 DeallocateInputMemory()

```
uint8_t*** DeallocateInputMemory (
            uint8_t *** image_in,
            const uint32_t * imgHeight,
            const uint32_t * imgWidth )
```

Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateInputMemory.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

NULL.

4.3.2.8 DeallocateOutputMemory()

```
uint8_t*** DeallocateOutputMemory (
            uint8_t *** image_out,
            const uint8_t * imgHeight,
            const uint8_t * imgWidth )
```

Funkcja dealokująca całą pamięć zajętą przez funkcję AllocateOutputMemory.

Parametry

image_out	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.

Zwraca

NULL.

4.3.2.9 load_sieve()

```
pixel_structure* load_sieve (
            FILE * stream )
```

Funkcja wczytująca do specjalnej struktury element strukturalny.

Parametry

stream	wskaźnik do pliku, który zawiera element strukturalny.

Zwraca

wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

4.3.2.10 main_point()

```
uint32_t* main_point (
            pixel_structure * sieve )
```

Algorytm wykrywający współrzędne piksela modyfikowanego (głównego) elementu strukturalnego.

Parametry

sieve	wskaźnik na dynamiczną listę jednokierunkową zawierającą element strukturalny.

Zwraca

wskaźnik na tablicę współrzędnych punktu głównego elementu strukturalnego.

4.3.2.11 opening()

```
void opening (
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
```

```
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Funkcja otwarcia morfologicznego, która wykonuje na obrazie w odpowiedniej kolejności operacje w celu usunięcia szczegółow z obrazu (erozja -> dylatacja).

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.3.2.12 operation()

```
void operation (
            const uint8_t type,
            uint8_t *** image_in,
            uint8_t *** image_out,
            uint32_t width,
            uint32_t height,
            pixel_structure * sieve,
            uint32_t * coordinates )
```

Główny algorytm, który odpowiada za wykonanie operacji morfologicznych na obrazie. Funkcja dopasowuje strukturę sita do piksela modyfikowanego tak, by można było sprawdzić warunki zamiany piksela modyfikowanego.

Parametry

type	znak warunkujący operację, która ma być wykonana przez funkcję ('e' – erozja; 'd' – dylatacja).
image_in	wskaźnik do początku tablicy na dane wejściowe.
image_out	wskaźnik do początku tablicy na dane wyjściowe.
width	szerokość obrazu.
width	wysokość obrazu.
sieve	wskaźnik na strukturę zawierającą dane elementu strukturalnego.
coordinates	tablica zawierająca współrzędne położenia piksela modyfikowanego (głównego).

4.3.2.13 ReadPixels()

```
void ReadPixels (
            uint8_t *** image_in,
            const uint32_t * imgHeight,
            const uint32_t * imgWidth,
            const FILE * input_file,
            uint32_t * offset_byte )
```

Funkcja sczytująca do struktury poszczególne składowe każdego piksela obrazu wejściowego. Jeżeli którykolwiek ze składowych koloru piksela ma wartość większą od 0 – wszystkie składowe są maksymalizowane.

Parametry

image_in	wskaźnik do początku tablicy na dane wejściowe.
imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.
input_file	wskaźnik na plik wejściowy.
offset_byte	numer bajtu, od którego zaczynają się dane związane z zawartością poszczególnych składowych pikseli.

4.3.2.14 ReadToStructure()

```
void ReadToStructure (
            const BMPFILEHEADER * FileHeader,
            const BITMAPHEADER * Header,
            const FILE * input_file )
```

Funkcja zapisująca do struktury dane nagłówka pliku, zawierającego informacje nt. obrazu, m.in. rozdzielczość, wysokość, szerokość, waga pliku, etc.

Parametry

FileHeader	wskaźnik na podstrukturę mówiąca o tym, czy plik rzeczywiście jest plikiem mapy bitowej. Podstruktura mówi m.in. od którego bajtu zaczynają się piksele.
Header	wskaźnik na podstrukturę, której dane opisują wymiary obrazu. Podstruktura zawiera przede wszystkim informacje techniczne obrazu.
input_file	wskaźnik na plik wejściowy.

4.3.2.15 SaveBMPFile()

```
void SaveBMPFile (
            const uint32_t * imgHeight,
            const uint32_t * imgWidth,
            const uint8_t *** image_out,
            const FILE * output_file )
```

Funkcja zapisująca bajty składowych kolorów pikseli do pliku wyjściowego.

Parametry

imgHeight	wskaźnik na wysokość obrazu.
imgWidth	wskaźnik na szerokość obrazu.
image_out	wskaźnik do początku tablicy na dane wyjściowe.
output_file	wskaźnik na plik wyjściowy.

4.3.2.16 WriteFromStructure()

```
void WriteFromStructure (
            const BMPFILEHEADER * FileHeader,
            const BITMAPHEADER * Header,
            const FILE * output_file )
```

Funkcja zapisująca do pliku wyjściowego pozyskane przez funkcję ReadToStructure dane, zapisane do struktury.

Parametry

FileHeader	wskaźnik na podstrukturę mówiąca o tym, czy plik rzeczywiście jest plikiem mapy bitowej. Podstruktura mówi m.in. od którego bajtu zaczynają się piksele.
Header	wskaźnik na podstrukturę, której dane opisują wymiary obrazu. Podstruktura zawiera przede wszystkim informacje techniczne obrazu.
output_file	wskaźnik na plik wyjściowy.

4.4 Dokumentacja pliku main.c

Plik źródłowy.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#include <inttypes.h>
#include <ctype.h>
#include "bmp.h"
#include "functions.h"
```

Funkcje

- int main (int argc, char **argv)

4.4.1 Opis szczegółowy

Plik źródłowy.

Autor

Aleksander Augustyniak

4.4.2 Dokumentacja funkcji

4.4.2.1 main()

```
int main (
              int argc,
              char ** argv )
```

Indeks