

# Architektura Komputerów, Projekt Intel

Wiktor Ślęczka

27 stycznia 2014

## Streszczenie

Program wczytuje z pliku listę punktów wysokości, a następnie tworzy dwie bitmapy, jedną przedstawiającą mapę wysokości za pomocą skali szarości, a drugą pokazującą przekrój pomiędzy dwoma podanymi punktami.

## 1. Opis programu

### 1.1. Działanie

Program wczytuje dwa pliki, jeden z mapą wartości punktów wysokości, drugi z parametrami mapy. Na ich podstawie generuje bitmapę pokazującą wysokość w skali szarości, pomiędzy dwoma wartościami, podanymi w pliku konfiguracyjnym. Wartości niższe i wyższe ukazywane są jako jednolity biały lub czarny kolor. Na bitmapie oznaczona jest również czerwona linia, wzdłuż której poprowadzony będzie przekrój. Program generuje również plik z bitmapą zawierającą przekrój pomiędzy dwoma danymi punktami, w skali i o poprawnej długości przekroju.

### 1.2. Wejście

Wejście programu to dwa pliki, "mapa.txt" i "parametry.txt". Pierwszy zawiera 201 linii, w każdej znajduje się 201 liczb rozdzielonych spacjami. Są to wartości poszczególnych punktów pomiaru, reprezentujące kwadraty 20x20 metrów, wyrażone w metrach. W drugim pliku znajdują się 4 linie. Pierwsze dwie to współrzędne punktów, pomiędzy którymi tworzony jest przekrój.

### 1.3. Wyjście

Wyjście programu składa się z dwóch plików - "mapa.bmp" oraz "przekrój.bmp". Pierwszy plik zawiera wykonaną w skali szarości mapę wysokości dla każdego z 40401 punktów podanych na wejściu, przy czym kolor o punktów wartości mniejszych niż podana w konfiguracji wartość minimalna są czarne, natomiast większych od wartości maksymalnej reprezentowane są w postaci barwy białej. Barwa punktów o wartości pomiędzy wartością minimalną a maksymalną zależy w sposób liniowy od wartości wysokości punktu<sup>1</sup>. W pliku tym połączono również czerwonym odcinkiem dwa punkty, pomiędzy którymi wykonany zostanie przekrój.

Drugi plik zawiera przekrój pomiędzy dwoma punktami podanymi w konfiguracji, w kolejności pierwszy punkt rysowany z lewej. Przekrój jest wykonany w odpowiedniej skali, 1m/1pixel pionowo oraz 20m/1pixel poziomo. Szerokość przekroju zależy od jego długości.

---

<sup>1</sup>Technicznie są to kwadraty o określonym boku, ale w praktyce pomiar wykonuje się w konkretnym punkcie lub uśrednia, co można sprowadzić do punktu.

## 2. Struktura programu

## 3. Struktura

Program składa się z dwóch funkcji napisanych w assemblerze x86.

**int mapa(int \*mapa, unsigned char \*image, MapaStruct \*par);** tworzy wewnątrz podanej tablicy image obraz przedstawiający mapę wysokości zgodny z formatem BMP, a następnie rysuje na nim czerwoną linię oznaczającą przekrój.

**int przekroj(int \*mapa, unsigned char \*image, PrzekStruct \*przek);** tworzy obraz przedstawiający przekrój pomiędzy podanymi punktami.

Poza tym w pliku "data.c" znajdują się definicje struktur danych oraz funkcji służących do generowania poprawnych nagłówków dla plików BMP.

## 4. Struktury danych

Program używa przede wszystkim trzech buforów, jednego na przechowywanie tablicy integerów, które reprezentują mapę wysokości wczytaną z pliku, o wielkości 161604 bajtów, drugiego na przechowywanie wynikowego obrazu przedstawiającego wynikową mapę wysokości, o długości 121404 bajtów oraz trzeci, zawierający przekrój, o zmiennym rozmiarze pomiędzy 121404 i 171692 bajtów.

Poza tym w programie używana jest również struktura pomocnicza zawierająca dane parametry. Służy ona do przekazania do obu funkcji parametrów podanych w pliku z konfiguracją. Zawiera ona sześć pól typu całkowitego, opisujące kolejno minimalną oraz maksymalną wysokość pól, oraz położenie punktów przekroju w kolejności x1, y1, x2, y2.

## 5. Algorytmy użyte w programie.

### 5.1. Rysowanie mapy w skali szarości

Dla każdego wczytanego punktu obliczana jest wartość koloru, jaki będzie go przedstawiał ze wzoru

$$\text{kolor} = \frac{\text{wysokosc pola} - \text{minimalna wartosc}}{\text{maksymalna wartosc}} * 256$$

Operacje o tym samym poziomie priorytetu są wykonywane w kolejności najpierw wzrastająca, potem zmniejszająca, co pozwala na pominięcie ułamków, oraz, ponieważ liczba i tak byłaby ucięta do wartości całkowitej<sup>2</sup>, pozwala zachować maksymalną możliwą w tym przypadku precyzję. Znajdowanie punktu w bitmapie polega na iterowaniu po całości, dla każdej wartości o 3 bajty, z wyjątkiem ostatniej kolumny każdego wiersza, która wymaga zaokrąglenia offsetu do pełnych 4 bajtów.

### 5.2. Rysowanie czerwonej linii

Najpierw określany jest, czy większa jest odległość w osi x czy y. Następnie odbywa się iteracja wzdłuż wybranej osi, w taki sposób, że dla każdego wiersza/kolumny prostopadłego do osi obliczane są współrzędne czerwonego punktu w następujący sposób:

$$x = x_1 + \text{sgn}(x_2 - x_1) * \text{numer kroku}$$

$$y = y_1 + \text{numer kroku} * \frac{y_2 - y_1}{x_2 - x_1}$$

Po wykonaniu tych czynności uzyskuje się linię łączącą punkty  $(x_1, y_1)$ ,  $(x_2, y_2)$  prostą linią, z dokładnością do jednego pixela względem linii prowadzonej optymalnie.

<sup>2</sup>Format bitmapy przyjmuje wartości całkowite

### 5.3. Rysowanie przekroju

Rysowanie przekroju odbywa się analogicznie do rysowania linii, z tą różnicą, że dla każdego wiersza wartość pobrana jest z punktu wyliczonego z tego wzoru z tablicy wczytanych wartości, natomiast wynikowa bitmapa jest przechodzona iteracyjnie kolumna po kolumnie, i zamalowywana jest podana wartość odczytana z tabeli.

## 6. Testy

Do programu dostarczony jest skrypt w Pythonie 3, pozwalający na wygenerowanie zestawu punktów (plik mapa.txt) na podstawie podanej funkcji. Punkty te mają podane wartości z zakresu  $< 0, 0 > x < 201, 201 >$ . W związku z tym testy przeprowadzono na następujących funkcjach:

- $f(x, y) = x$
- $f(x, y) = y$
- $f(x, y) = 200 - x$
- $f(x, y) = 200 - y$
- $f(x, y) = \frac{x^2}{200}$
- $f(x, y) = \frac{x^2 + y^2}{400}$
- $f(x, y) = x + y$

oraz dla kombinacji następujących parametrów:

- min: 0 50
- max: 50 100 200 400
- p1: (0, 0) (200, 0) (0 200) (200, 200) (100, 100) (100, 110) (110, 100)
- p2: (0, 0) (200, 0) (0,200) (200, 200) (110, 110) (110, 100) (100, 110)