

03-SFC – DOKUMENTACE

Command line Arguments

-i, --input Required. (Default:) Input file-name.

-m, --method (Default: 0) Pixel order method (-1 for the best one).

Program:

- Program obsahuje 5 základních 2D křivek a to [ScanLine](#), [SimpleDiagonal](#), [DFSCurve](#), [Spirala](#), [Hilbert](#), [Scan2](#)
- [ScanLine](#) – Vykreslování postupných řádků.
- [SimpleDiagonal](#) – Vykreslování podle postupných diagonál.
- [DFSCurve](#) – Vykreslování podle grafového algoritmu DFS.
- [Spirala](#) – Vykreslování pomocí spirály
- [Hilbert](#) – Vykreslování pomocí HilbertFunkce.
- [Scan2](#) – Vykreslování pomocí sloupců

Algorithm:

[ScanLine](#) – Řešení pomocí dvou for cyklů

[SimpleDiagonal](#) – postupně měníme diagonálu. – používám while cyklus

[DFSCurve](#) – používám vykreslovací grafový algoritmus DFS.

[Spirala](#) – postupně kreslím čtverce až do středu.

[Hilbert](#) – Pomocí rekurze generuju křivku.

[Scan2](#) – Pomocí dvou for cyklů.

Kod:

Kod obsahuje tři soubory program.cs, program1.cs a program2.cs.

- Program.cs je hlavní program pro generování typu křivky
- Program2.cs je soubor, který obsahuje třídy křivek s funkcemi

```
// Pixel order method.
IPixelOrder sfc;
switch (o.Method)
{
    case 0:
    case -1:
        sfc = new ScanLine(pixelAction);
        break;
    case 1:
        sfc = new SimpleDiagonal(pixelAction);
        break;
    case 2:
        sfc = new DFSCurve(pixelAction);
        break;
    case 3:
        sfc = new Spirala(pixelAction);
        break;
    case 4:
        sfc = new Hilbert(pixelAction);
        break;
    case 5:
        sfc = new Scan2(pixelAction);
        break;

    default:
        Console.WriteLine($"Invalid method {o.Method}.");
        return;
}
```

Output:

```
95998987
Image: C:\Users\Admin\source\repos\ConsoleApp64\bin\Debug\net6.0\Pattern1.png[4096x4096]
Order: Hilbert
Average entropy: 5.72 bits per pixel
```

- Kde první řádek je hodnota entropie