

Ohjelman kuvaus

Ohjelma tulee ratkaisemaan sille annetusta valokuvasta määritetyistä aloitus- ja lopetuspisteistä niiden välillä kulkevan lyhimmän polun. Lyhimmällä polulla tarkoitetaan reittiä, mikä kulkee aloitus pisteestä lopetus pisteeseen käyttäen mahdollisimman pieniä RGB värin arvoja. Kuvassa olevat pisteet voi ajatella verkon solmuina ja pisteestä toiseen siirtyminen vastaa verkon kaaria. Pisteeseen johon siirrytään, siitä saatava RGB-värikoodi edustaa pisteiden välisen kaaren painoa

Ohjelman yleisrakenne

Ohjelma sisältää seuraavat luokat: Astar, BellmanFord, Dijkstra, Dijkstra8, EtsiReitti, EtsiReittiUI, Keko, Sijainti.

EtsiReittiUI toteuttaa ohjelman käyttöliittymän ja sen toiminnallisuuden. EtsiReitti luokka toimii ohjelman pääluokkana, josta ohjelmassa olevia algoritmeja kutsutaan.

Luokat Astar, BellmanFord, Dijkstra, Dijkstra8 ovat ohjelmassa luokkia, joissa toteutetaan erilaisia algoritmeja lyhimmän reitin löytämiseksi.

Luokka Astar etsii reitin käyttäen Dijkstran algoritmia, johon on lisätty heuristiikka funktio. Reitti voi kulkea vain pääilmansuuntiin.

Luokka BellmanFord etsii reitin käyttäen BellmanFordin algoritmia liikkuen vain pääilmansuuntiin. Käytettävän kuvan koko on rajoitettu 100x100 pistettä, koska algoritmi on suhteellisen hidas suurella kuvakoolla.

Luokat Dijkstra ja Dijkstra8 etsivät reitin käyttäen Dijkstran algoritmia. Dijkstra etsii reitin kulkien vain pääilmansuuntiin. Dijkstra8 etsii reitin kulkien pääilmansuuntien lisäksi myös väli-ilmansuuntiin.

Sijainti luokka toimii pääohjelman apuna. Se sisältää sijainti- ja etäisyystiedot. Keko luokka toteuttaa minimi keon, johon talletetaan sijainti oliot ja järjestää ne muuttujan etäisyyden mukaan.

Saavutetut aika- ja tilavaativuudet (m.m. O-analyysi pseudokoodista)

Dijkstran algoritmin aikavaativuus minimikekoa käytettäessä on:
 $O((|E| + |V|) \log |V|)$ ja tilavaativuus on puolestaan: $O(|V|)$.

Olen testannut oman ohjelman suoritusaikoja erikokoisilla kuvilla. Kuvasta reitti etsitään aina vasemmasta ylä-reunasta oikeaan ala reunaan. Seuraavassa taulukossa on muutamia suorituskyyä kuvaavia aikoja erikokoisilla syötteillä:

Syötteen koko	Saavutettu aika ms
10x10	0.8
20x20	3.0
40x40	6.4
160x160	13.4
320x320	32.4
640x640	118.2
1280x1280	577.4

Tuloksista voidaan karkeasti nähdä, että ohjelma saavuttaa ainakin aikavaativuuden $O(n^2)$. Tämä aikavaativuus on kyseiselle ohjelmalle riittävä (Antin kommentti ohjauksessa). Ohjelman tilavaativuus on: $O(n)$.

Työn mahdolliset puutteet ja parannusehdotukset

Ohjelmaa voisi vielä yrittää nopeuttaa käyttämällä esimerkiksi binäärikeon sijasta fibonacci-kekoa. Lisäksi ohjelmaan voisi lisätä mahdollisuuden valita piirrettävän reitin värin ja koon.

Lähteet

<http://www.cs.helsinki.fi/u/floreen/tira2013/tira.pdf>

”voiko Google olla lähteenä?”