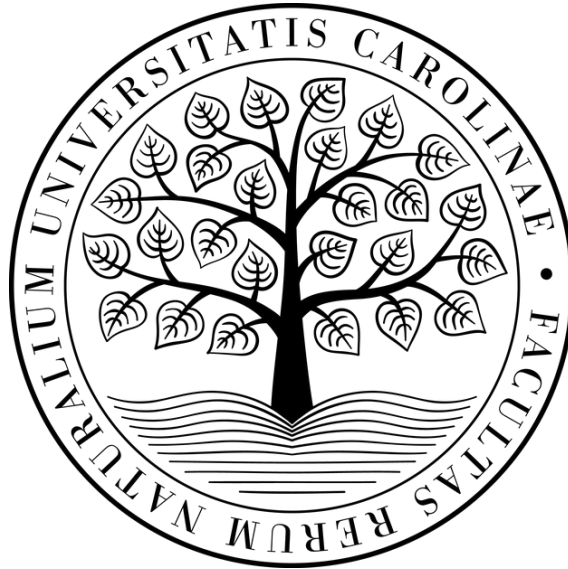


Univerzita Karlova

Přírodovědecká Fakulta



Úvod do programování

Algoritmus Visvalingam-Whyatt

Technická zpráva

Eliška Pospěchová
Praha 2022
2. ročník
B-FGG

Zadání

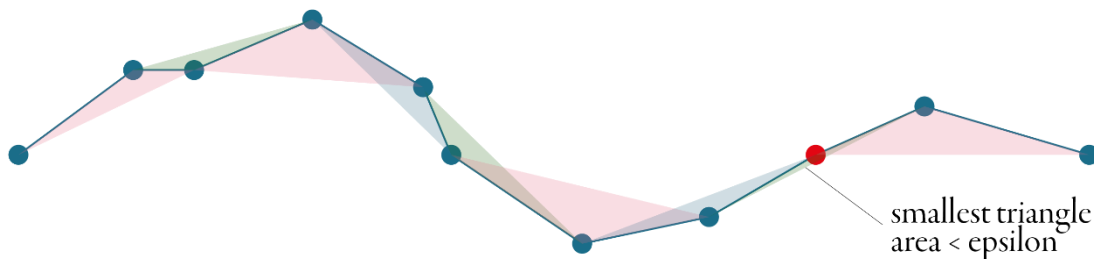
Příklad č. 123: Zjednodušení lomené čáry algoritmem Whyatt.

Rozbor problému

Zjednodušením linie je myšleno odstranění určitého počtu vertexů na této linii, aby její průběh byl jednodušší, ale byl co nejlépe zachován její tvar. Je to jedna z nejvýznamnějších metod generalizace v kartografii, kdy při malém měřítku u příliš složitých linií dochází ke slévání. Algoritmy simplifikace se dají použít jak na liniové prvky, tak na prvky polygonové, avšak u polygonů se často musí počítat s topologií, pokud spolu polygony sousedí. Mezi nepoužívanější algoritmy patří Douglas-Peckerův nebo Visvalingam-Whyattův.

Použitý algoritmus

Algoritmus Visvalingam-Whyatt pracuje s obsahy trojúhelníků, jejichž vrcholy jsou 3 po sobě jdoucí vertexy na linii. Spočítá obsahy trojúhelníků pro každou takovou trojici vertexů a najde ten s nejmenším obsahem. Vertex, který je vrcholem trojúhelníku s nejmenším obsahem a na linii se nachází mezi dvěma zbylými vrcholy tohoto trojúhelníku, je odstraněn v případě, že obsah trojúhelníku je menší než epsilon. Takto algoritmus pracuje, dokud nemají všechny trojúhelníky obsah větší než epsilon. Někdy se ovšem místo epsilon jako parametr používá podíl vertexů, které chceme ponechat.



Vstupní data

Program pracuje se souborem *input_d1.geojson*. Jedná se o soubor typu GeoJSON s jedním prvkem, konkrétně linií reprezentující dálnici D1 z databáze ArcČR500. Pro správné načtení musí být nainportován modul *json*. Program získá potřebné souřadnice vertexů z načteného souboru a uloží je do proměnné. Pro lepší práci se souřadnicemi byla na tuto proměnnou použita funkce *zip*, která vrátí iterátor, který vygeneruje 2 *n*-tice, v tomto případě jsou v jedné *n*-tici *x*-ové souřadnice a v druhé *y*-ové.

Výstup

Výstupem je grafická vizualizace původní i zjednodušené linie za použití knihovny *matplotlib*.

Požadavky

Pro správný chod programu je třeba mít připravená vstupní data ve správném adresáři a mít nainstalovanou knihovnu *matplotlib*.

Metody

Pythagorova věta – *pythagoras (x1, x2, y1, y2)*

Spočítá vzdálenost 2 bodů se známými souřadnicemi za použití Pythagorovy věty.

Heronův vzorec – *heron (a, b, c)*

Spočítá obsah trojúhelníku se známými délkami stran za použití Heronova vzorce.

Algoritmus Visvalingam-Whyatt – *visvalingam_whyatt (vertices, epsilon)*

Před zahájením samotného algoritmu byly vytvořeny prázdné 3 seznamy, do nichž se pak budou ukládat délky jednotlivých úseků linie, vzdálenost těch vertexů, mezi nimiž na linii leží právě jeden jiný vertex, a obsahy trojúhelníků tvořených 3 po sobě jdoucími vertexy na linii. Při výpočtu délek úseků a vzdáleností „ob bod“ cyklus *for* iteruje přes *n*-tice *x*-ových a *y*-ových souřadnic a tyto vzdálenosti jsou počítány za použití Pythagorovy věty a následně přidávány do příslušných seznamů.

Pro výpočet obsahů trojúhelníků byl použit Heronův vzorec, pro jehož použití stačí znát délky stran trojúhelníku. Tyto obsahy byly opět uloženy do seznamu.

Algoritmus popsaný výše by mohl nabádat k počítání obsahů všech trojúhelníků při každé iteraci, ale snadno nahlédneme, že stačí přepočítat obsahy pouze trojúhelníků obsahující smazaný bod.

Parametry:

vertices: seznam souřadnic vertexů linie

epsilon: minimální obsah trojúhelníku určeného 3 po sobě jdoucími body na linii, v programu defaultně nastavena na 0,001

Výstupy:

simplified: iterátor, vygeneruje 2 *n*-tice, jedna obsahuje *x*-ové, druhá *y*-ové souřadnice vertexů zjednodušené linie

Pseudokód algoritmu:

založ seznam, kam se budou ukládat délky úseků linie

založ seznam, kam se budou ukládat vzdálenosti ob vertex

založ seznam, kam se budou ukládat obsahy trojúhelníku definovaného 3 po sobě jdoucími body na linii

pro každý úsek linie:

$v1$ = souřadnice počátku úseku

$v2$ = souřadnice konce úseku

d = vzdálenost mezi $v1$ a $v2$ vypočítaná pomocí Pythagorovy věty

přidej d do seznamu délek úseků linie

pro každou vzdálenost vertexů ob vertex:

$w1$ = souřadnice počátku této úsečky

$w2$ = souřadnice konce této úsečky

c = vzdálenost mezi $w1$ a $w2$ vypočítaná užitím Pythagorovy věty

přidej c do seznamu délek úseček ob vertex

pro každý trojúhelník definovaný 3 po sobě jdoucími body na linii:

a = délka 1. strany trojúhelníku ležící na linii

b = délka 2. strany trojúhelníku ležící na linii

c = vzdálenost 1. a 3. vrcholu trojúhelníku (úsečka ob vertex, 3. strana trojúhelníku)

$area$ = obsah trojúhelníku definovaného délkami stran a , b , c

vypočítaný užitím Heronova vzorce

přidej $area$ do seznamu obsahů trojúhelníků

dokud má linie více než 2 vertexy a minimální obsah trojúhelníku menší než ϵ :

$rank$ = index trojúhelníku s nejmenším obsahem

odstraň prostřední vertex trojúhelníků s nejmenším obsahem

délka 1. strany trojúhelníku = délka 3. strany trojúhelníku

odstraň 2. stranu trojúhelníku

pokud trojúhelník s nejmenším obsahem není ten první:

přepočítej 3. stranu předchozího trojúhelníku pomocí

Pythagorovy věty používaje souřadnice jejích koncových bodů

přepočítej obsah předchozího trojúhelníku užitím Heronova

vzorce používaje délky jeho stran

odstraň 3. stranu trojúhelníku s nejmenším obsahem

odstraň obsah nejmenšího trojúhelníku ze seznamu obsahů

pokud trojúhelník s nejmenším obsahem není ten poslední:

přepočítej 3. stranu následujícího trojúhelníku pomocí
Pythagorovy věty používaje souřadnice jejích koncových bodů
přepočítej obsah následujícího trojúhelníku užitím Heronova
vzorce používaje délky jeho stran

vrať upravený seznam vertexů linie

Vizualizace – *visualize (raw_x, raw_y, simplified_x, simplified_y)*

Vizualizuje původní a simplifikovanou linii za použití python knihovny matplotlib.

Možná vylepšení

Program by teoreticky byl schopný zjednodušit i obrys polygonu, ale počáteční a zároveň koncový vertex takového polygonu by nešlo odstranit. Dále by se dal program vylepšit tak, aby byl schopný zjednodušit linii několik a nejen jednu. Také by bylo vhodné, aby výstupem kromě grafické vizualizace byl i soubor obsahující souřadnice vertexů zjednodušené linie.

Dále by mohl být program interaktivnější, například zadáním hodnoty *epsilon* nebo načtením vlastních vstupních dat a být flexibilnější co se typu vstupního souboru týče (program by našel souřadnice vertexů sám). Zároveň by mohl program navrhnout optimální hodnotu (resp. rozsah hodnot) epsilon v závislosti na souřadnicovém systému.

Zdroje

- (1) FLEISCHMANN. M. (2021): Line simplification algorithms. <https://martinflischmann.net/line-simplification-algorithms/> (11. 2. 2022)
- (2) ARCDATA PRAHA, ZÚ, ČSÚ (2016): ArcČR 500 - digitální geografická databáze, verze 3.3. www.arcdata.cz (11. 2. 2022)
- (3) Python Software Foundation: Python 3.10.2 documentation. <https://docs.python.org/3/> (11. 2. 2022)