

Úloha 4 – Energetické spliny

Algoritmy počítačové kartografie

Sabina Fukalová, Klára Linková

Praha 2023

1. Zadání úlohy

1.1. Povinná část

S využitím generalizačního operátoru Partial Modification realizujte odsun a částečnou změnu tvaru jednoho blízkého prvku vůči blízkému pevnému prvku (bariéře) tak, aby v generalizované mapě nedošlo k jejich grafickému konfliktu. Hodnotu minimální vzdálenosti prvku d volte v závislosti na měřítkovém čísle mapy (např. 1 mm v mapě). Pro implementaci použijte metodu energetických splinů.

Jako vstupní data použijte existující kartografická data (např. silniční či železniční síť, vodstvo), která budou načítána ze dvou textových souborů ve vámi zvoleném formátu.

Grafické rozhraní realizujte s využitím frameworku QT, výsledky generalizačních operací vizualizujte. Porovnejte dosažené výsledky s ruční generalizací prováděnou kartografickým expertem.

1.2. Volitelná část

/

2. Popis a rozbor problému

Dle Bayer (2008, s. 161) „Automatizovaná kartografická generalizace představuje poměrně aktuální problém, který souvisí se stále rostoucím spektrem a zvětšujícím se objemem dat geoinformačního charakteru, jež jsou k dispozici o zájmovém území“. Kartografickou generalizací jsou prováděny prostorové a atributové transformace kartografických dat a výsledkem je nové kartografické dílo, které vzniklo z původního kartografického díla např. změnou měřítka. Právě při změně měřítka dochází často ke konfliktu objektů, které je třeba řešit, aby se objekty nedotýkaly a nevytvářely tak chybný obraz reality.

2.1. Energetické spliny

Energetický spline (angl. *snake*) je metoda používaná v oblasti počítačové grafiky, která se v posledních letech dostala v upravené formě i do oblasti kartografie. Operátor *partial modification* je reprezentován parametrickou křivkou

$$d(s) = (x(s) - x_{(0)}(s), y(s) - y_0(s)),$$

kde $x_{(0)}$, $y_{(0)}$ jsou souřadnice vstupního prvku, x , y souřadnice generalizovaného prvku, s je parametr ležící v intervalu od 0 do 1 (Bayer 2023b). Energetický model pro křivku L s délkou l má tvar

$$E(d) = \int_l E_i(s)ds + \int_l E_e(s)ds,$$

kde E_i označuje vnitřní energii splinu, E_e vnější energii splinu (Bayer 2023b). Cílem metody je minimalizovat vnitřní a vnější energii a nalézt tak rovnovážnou polohu mezi těmito dvěma stavby.

Vnitřní energie je dána mechanickými vlastnostmi materiálu (tuhost, pevnost,...) a ovlivňuje průběh splinu a jeho tvar. Vnitřní energie je definována vztahem

$$E_i(s) = \frac{1}{2} \left(\alpha(s) \|d(s)\|^2 + \beta(s) \left\| \frac{\partial d(s)}{\partial s} \right\|^2 + \gamma(s) \left\| \frac{\partial^2 d(s)}{\partial s^2} \right\|^2 \right).$$

První člen měří vzdálenost splinu od původního elementu, druhý napětí (elasticitu) splinu a poslední člen tuhost (křivost) splinu. Parametry α , β , γ modifikují chování křivky a určují tak, jak moc se budou jednotlivé členy projevovat.

Vnější energie představuje silové působení blízkých předmětů a to tak, jako by na sebe vzájemně působily odpudivou silou. Velikost silového působení klesá se vzdáleností. Hlavními faktory ovlivňující míru deformace je gradient (strmost) a omezenost funkce shora (Bayer 2023b). Energetická funkce popisující silový model může mít různé podoby a je volená uživatelem (Bayer 2023a).

2.2. Partial displacement

Tato kapitola vychází z Bayer (2023b). Generalizační operace *Partial Displacement*, neboli částečná změna tvaru, zahrnuje posun a změnu tvaru částí prvku v místech, kde se přiblíží k jinému prvku blíže, než je zadaná hodnota \underline{d} . V praxi se tato metoda používá v těchto případech – částečná modifikace jednoho prvku, částečná modifikace obou prvků a kombinace obou případů. V případě modifikace jednoho prvku je jeden prvek pevný (tzv. bariéra), nesmí se měnit. Polohu druhého prvku je možno měnit. V případě částečné modifikace obou prvků dochází ke generalizaci obou prvků – je možné měnit jejich polohu a tvar. Při kombinaci obou případů se jedná o model bariéry a generalizovaných prvků, kdy je možné měnit polohu a tvar generalizovaných prvků vůči sobě i vzhledem k bariéře.

Aby nedošlo k přiblížení dvou prvků na vzdálenost menší než \underline{d} (v kartografii lze chápat jako minimální vzdálenost prvků v mapě, kdy jsme schopni je ještě rozlišit, než dojde k jejich grafickému slití), je zavedena tzv. energetická funkce. Energetická funkce má tvar

$$E_e(x, y) = \begin{cases} c(1 - \frac{d}{\underline{d}}), & d < \underline{d}, \\ 0, & \text{jinak,} \end{cases}$$

kde c je konstanta, která udává, jak moc tento člen ovlivní tvar splinu. Iterativní řešení diskretizované varianty splinu využívá parciální derivace $E_e(x, y)$ dle x, y . Pro $d < \underline{d}$ má energetická funkce tvar

$$E_e(x, y) = c(1 - \frac{\sqrt{(x - x_n)^2 + (y - y_n)^2}}{\underline{d}}),$$

kde $q_n = [x_n, y_n]$ je nejbližší vrchol k vrcholu $p = [x, y]$, parciální derivace mají tvar

$$\frac{\partial E_e(x, y)}{\partial x} = -c \frac{x - x_n}{d \underline{d}},$$

$$\frac{\partial E_e(x, y)}{\partial y} = -c \frac{y - y_n}{d \underline{d}}.$$

Pro $d > \underline{d}$ mají parciální derivace tvar

$$\frac{\partial E_e(x, y)}{\partial x} = \frac{\partial E_e(x, y)}{\partial y} = 0.$$

V rámci tohoto úkolu je řešen případ modifikace pouze jednoho prvku, proto je následující kapitola zaměřená na tuto problematiku.

2.2.1. Modifikace jednoho prvku

Jak už bylo popsáno výše, tato varianta se skládá z jednoho pevného prvku (bariéry) a jednoho generalizovaného prvku. Bariéra je polylinie $B = \{q_1, q_2, \dots, q_m\}$ s m vrcholy $q_j = [x_j, y_j]$. Generalizovaný prvek je představován polylinií $L = \{p_1, p_2, \dots, p_n\}$ s n vrcholy $p_i = [x_i, y_i]$. Částečná změna tvaru a polohy je aplikována pouze na generalizovaný prvek. Energetická funkce má tvar

$$E_e(x_i, y_i) = \begin{cases} c(1 - \frac{d_i}{\underline{d}}), & d_i < \underline{d}, \\ 0, & \text{jinak,} \end{cases}$$

kde d_i je vzdálenost mezi vrcholem p_i na generalizované polylinii L a nejbližším bodem q_n na bariéře. Vzdálenost d_i se spočítá pomocí vztahu

$$d_i = \sqrt{(x_i - x_n)^2 + (y_i - y_n)^2}.$$

Pro $d_i < \underline{d}$, mají parciální derivace $E_e(x_i, y_i)$ tvar

$$\frac{\partial E_e(x_i, y_i)}{\partial x_i} = -c \frac{x_i - x_n}{d_i \underline{d}},$$

$$\frac{\partial E_e(x_i, y_i)}{\partial y_i} = -c \frac{y_i - y_n}{d_i \underline{d}},$$

jinak

$$\frac{\partial E_e(x_i, y_i)}{\partial x_i} = \frac{\partial E_e(x_i, y_i)}{\partial y_i} = 0.$$

2.3. Další výpočty

Pro vypracování této úlohy hraje důležitou roli **vzdálenost bodu od přímky** $d(A, p)$, která se spočítá podle vztahu

$$d(A, p) = \frac{x_a(y_1 - y_2) + x_1(y_2 - y_a) + x_2(y_a - y_1)}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}},$$

kde x_a, y_a jsou souřadnice bodu A , x_1, y_1 a x_2, y_2 jsou souřadnice bodů definujících přímku p (Bayer 2022).

Dále je možné zjistit souřadnice bodu q , který leží na přímce p a je nejbližší bodu A . Pro výpočet souřadnic x_q a y_q jsou použit vztahy

$$x_q = x_1 + \frac{k(x_2 - x_1)}{d_{12}},$$

$$y_q = y_1 + \frac{k(y_2 - y_1)}{d_{12}},$$

příčemž hodnota k je spočtena ze vztahu

$$k = \sqrt{d_1^2 - d^2},$$

kde d_1 značí vzdálenost bodu P_1 od bodu A .

3. Dokumentace

Vytvořený program byl vytvořen v programu *PyCharm* a UI frameworku *QT*. Je rozdělen do několika modulů – *MainForm.py*, *draw.py*, *algorithms.py*, *setting.py*, jež obsahují názvům příslušné třídy. K celému programu pak přísluší adresář se vstupními reprezentačními daty a adresář *icons*, jenž obsahuje grafickou reprezentaci jednotlivých funkcí programu.

3.1. MainForm.py

Tento modul obsahuje třídu *Ui_MainForm*, která byla vytvořena automaticky pomocí QT Creatoru. Dále jsou doplněny metody, které se spustí při interakci s uživatelským rozhraním.

Metoda *runSettings* existuje k inicializaci vstupních parametrů pro vytvoření energetických splinů. Při spuštění se otevře dialogové okno, v němž může uživatel upravit hodnoty daných parametrů. Tyto informace v reálu pak metoda ukládá do specifických proměnných.

Následující metoda je nazvána *openFile*. Ta ukládá do příslušných proměnných ukládá zvolenou šířku a výšku uživatelského okna, ve kterém jsou vykreslovány body. Zároveň na to volá další metodu *loadData* příslušející třídě *Draw*, jež načítá uživatelem zvolená vstupní data a následně je v uživatelském okně vykreslí.

Metoda *displaceClick* vytvoří posunutý vstupní prvek vůči zvolené bariéře. Nejdříve jsou však z třídy *Draw* načteny seznamy, které obsahují body tvořící vstupní prvek a danou bariéru. Seznamům je zavolána metoda *minEnergySpline*, jež se nachází ve třídě *Algorithms*. Výsledek této metody je předán třídě *Draw*. Následně je vykreslen dosavadní výsledek.

Další metodou je metoda *drawLineClick*, která nahraje uživatelem zvolená data, jež reprezentují prvky, jež mají být pozměněny. Následně dochází k nastavení proměnné *add.L* příslušící třídě *Draw* a je vyvolána metoda *openFile*.

Pomocí metody *drawBarrierClick* se nahrají data, která tvoří bariéru. Princip je prakticky stejný jako u předchozí metody.

Metoda *clearClick* slouží k vymazání jak nahraných dat, tak vytvořených výsledků. K tomu je využita metoda *clearAll* ze třídy *Draw*, po které jiná metoda překreslí uživatelské okno.

3.2. draw.py

Modul *draw.py* obsahuje třídu *Draw*, kterou dědí od třídy *QWidget* z knihovny *PyQt6*. Díky této funkci jsou vykreslovány polylinie a bariéry a výsledná posunutá polylinie. Vykreslení je definováno v metodě *paintEvent*.

Metoda *loadData* načítá vstupních dat vybraných uživatelem a zároveň vykresluje linie. Nejprve je vyvoláno dialogové okno, z něhož je získána cesta k souboru v interním počítači, jenž obsahuje zvolená vstupní data. Metoda prochází zvolený soubor po řádcích a do interních seznamů ukládá x a y souřadnice bodů souboru. Následně je soubor vykreslen v uživatelském okně.

Díky metodě *add.L* se určí, zdali nahrané souřadnice tvoří prvek nebo bariéru.

Metody *getL* a *getB* obsahují seznamy, jež nesou body, které tvoří nahraný prvek a bariéru.

MetdatasetLD obsahuje seznam s body, jež tvoří programem upravený prvek a který přiřazuje do příslušné proměnné.

Metoda *setSource* představuje Booleans metriku, a tedy přiřazuje do připravené proměnné hodnotu *add.L*. To způsobí, že se změní dosavadní nastavení, které říká, zdali nahraná data představují prvek nebo bariéru.

Metoda *clearAll* souží k vymazání nahraných dat a vytvořených výsledků.

3.3. algorithms.py

V modulu *algorithms.py* se nachází jednotlivé algoritmy pro kartografickou generalizaci pomocí energetických splinů. Je zde jedna třída *Algorithms*, která obsahuje metody pro realizování kartografické generalizace pomocí energetických splinů.

Metoda *getEuclidDistance* má na vstupu souřadnice dvou bodů x_1, y_1, x_2, y_2 . Metoda vrací vzdálenost mezi těmito dvěma body.

Metoda *getPointLineDistance* počítá vzdálenost bodu od přímky na základě vzorce z kapitoly 2.3. Na vstupu jsou souřadnice x_A, y_A bodu A a souřadnice bodů P_1 a P_2 definujících přímku p . Výstupem této metody je vzdálenost d bodu od přímky a souřadnice bodu q .

Metoda *getPointLineSegmentDistance* má na vstupu souřadnice bodu A a souřadnice vrcholů liniového segmentu. V této metodě dochází k výpočtu vzdálenosti mezi bodem A a liniovým segmentem definovaným body P_1 a P_2 . Metoda vrací vzdálenost d a souřadnice bodu. Bližší popis algoritmu je uveden v kapitole 3.4.1.

Metoda *getNearestLineSegmentPoint* má na vstupu souřadnice bodu A ve formátu *float* a vektor souřadnic x a y lomových bodů bariéry ve formátu *matrix*. V této metodě dochází k určení nejbližšího bodu překážky pro každý liniový segment bariéry. Výstupem je minimální vzdálenost d_{\min} , souřadnice x_{\min} , y_{\min} nejbližšího bodu na nejbližším segmentu a index i_{\min} . Bližší popis algoritmu je uveden v kapitole 3.4.2.

Metoda *createA* má na vstupu koeficienty α , β , γ , průměrnou délku segmentu h a počet řádků m matice. Touto metodou je vytvořena matice A , která má na hlavní diagonále hodnoty a a na ostatních diagonálách hodnoty b a c . Počet řádků a sloupců matice A odpovídá počtu vrcholů vstupní polylinie (bariéry). Metoda vrací matici A .

Metoda *getEx* provádí parciální derivaci vnější energie podle x . Na vstupu jsou souřadnice x_i , y_i , x_n , y_n , d a d_{\min} , přičemž d_{\min} je minimální vzdálenost prvků, kterou jsme schopni akceptovat. Pokud je bod blíže, než je minimální vzdálenost prvků d_{\min} , projeví se vnější energie. Pokud je naopak bod dále než minimální vzdálenost prvků d_{\min} , vliv vnější energie je nulový a nedochází tak k odsunu.

Metoda *getEy* je stejná jako předchozí metoda, avšak provádí parciální derivaci vnější energie podle proměnné y .

Metoda *minEnergySpline* má na vstupu list L vrcholů $QPointF$ linie, list B vrcholů $QPointF$ bariéry, parametry α , β , γ , λ typu *float*, minimální vzdálenost prvků d_{\min} typu *float*, počet iterací *iters* typu *float*. Metoda vrací odsunutou linii LD . Bližší popis algoritmu je uveden v kapitole 3.4.3.

3.4. Bližší popis vybraných algoritmů

3.4.1. *getPointLineSegmentDistance*

Spočítej směrový vektor liniového segmentu

Spočítej normálový vektor liniového segmentu

Vytvoř nové body P_3 a P_4 , které leží na normálových vektorech procházejících body P_1 a P_2

Spočítej vzdálenost d_{13} bodu A od přímky definované body P_1 a P_3

Spočítej vzdálenost d_{24} bodu A od přímky definované body P_2 a P_4

Spočítej testovací kritérium jako součin d_{13} a d_{24}

Když je testovací kritérium menší než 0 (tzn. bod A leží mezi dvěma normálami):

Zjisti vzdálenost a souřadnice bodu q pomocí metody pro výpočet vzdálenosti bodu od přímky

Vrat' absolutní hodnotu vzdálenosti a souřadnice bodu q

Když je vzdálenost d_{13} větší než 0 (bod A leží v levé polorovině):

Vrat' vzdálenost bodu A od bodu P_1 vypočítanou pomocí metody pro výpočet vzdálenosti dvou bodů, a souřadnice bodu P_1 (jedná se o nejbližší bod liniového segmentu k bodu A)

Vrat' vzdálenost bodu A od bodu P_2 vypočítanou pomocí metody pro výpočet vzdálenosti dvou bodů, a souřadnice bodu P_2 (jedná se o nejbližší bod liniového segmentu k bodu A)

3.4.2. getNearestLineSegmentPoint

Inicializuj i_{\min} na -1

Inicializuj d_{\min} na nekonečno

Zjisti velikost m , n matice X

Projdi všechny liniové segmenty bariéry

Zjisti vzdálenost bodu od liniového segmentu

Když aktuální vzdálenost d_i je menší než minimální zapamatovaná vzdálenost d_{\min} :

Aktualizuj d_{\min} na d_i

Aktualizuj i_{\min} na i

Aktualizuj x_{\min} na x_i

Aktualizuj y_{\min} na y_i

Vrat' d_{\min} , i_{\min} , x_{\min} , y_{\min}

3.4.3. minEnergySpline

Do proměnné ml ulož počet lomových bodů linie

Do proměnné mb ulož počet lomových bodů bariéry

Vytvoř nulové matice XL , YL , XB , YB , které budou představovat sloupcové vektory s délkou m nebo b a jedním sloupcem

Projdi všechny body linie

Do matice XB ulož souřadnici x vrcholového bodu linie

Do matice YB ulož souřadnici y vrcholového bodu linie

Projdi všechny body bariéry

Do matice XL ulož souřadnici x vrcholového bodu bariéry

Do matice YL ulož souřadnici y vrcholového bodu bariéry

Spočítej souřadnicové difference dx a dy

Spočítej vzdálenosti mezi jednotlivými vrcholy H generalizované polylinie

Spočítej krok h jako střední hodnotu H

Vytvoř matici A za pomoci metody `createA` se těmito vstupními parametry α , β , γ , krok h a počet vrcholů ml odsouvaného prvku

Vytvoř jednotkovou matici I

Spočítej inverzní matici k B , jestliže $B = A + \lambda I$

Vytvoř nulové matice DX a DY s ml řádky a jedním sloupcem

Inicializuj souřadnice vrcholu po i -té iteraci na XL a YL

Opakuj při indexy od 0 do $iters$

Vytvoř prázdnou matici E_x parciálních derivací potenciálů podle x o „ml“ řádků a jednom sloupci

Vytvoř prázdnou matici E_y parciálních derivací potenciálů podle y o „ml“ řádků a jednom sloupci

Projdi vrcholy j polylinie

Najdi na bariéře nejbližší body o souřadnicích x_n, y_n pomocí metody `getNearestLineSegmentPoint`

Spočítej parciální derivace podle x a y pomocí metody `getEx` a `getEy`

*Spočítej nový posun DX jako $DX = AI * (lam * DX - E_x)$*

*Spočítej nový posun DY jako $DY = AI * (lam * DY - E_y)$*

Spočítej novou souřadnici x (XL_i) posunutého bodu jako $XL_i = XL + DX$

Spočítej novou souřadnici y (YL_i) posunutého bodu jako $YL_i = YL + DY$

Vytvoř prázdný list LD

Projdi všechny vrcholy polylinie

Vytvoř nový `QPointF` z prvků matic XL_i, YL_i a ulož ho do proměnné v

Přidej v do LD

Vrat' LD

3.5. Vstup

Vstupní data lze do aplikace nahrát jako textový soubor .txt. Soubor reprezentuje x a y souřadnice prvků, jež do aplikace vstupují.

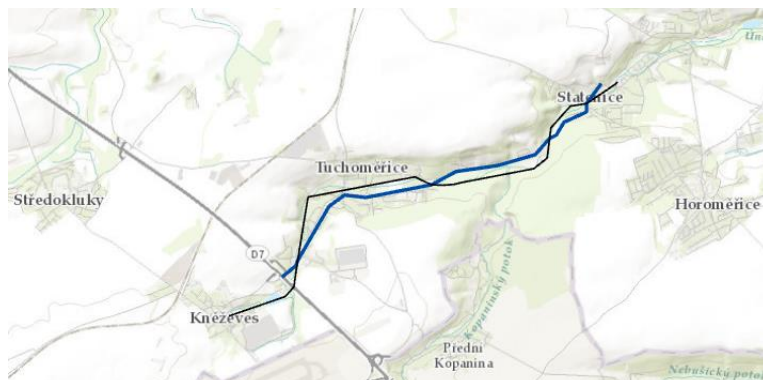
3.6. Výstup

Výstup je zajištěn v podobě vizualizace v uživatelském rozhraní. Vizualizace vstupních prvků je rozdělena na prvek, který má být modifikován, a na prvek, jenž představuje bariéru.

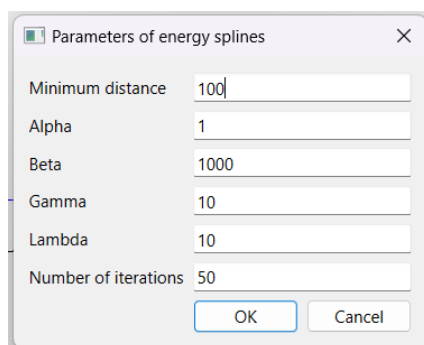
Modifikovaný načtený prvek je v aplikaci reprezentován modrou barvou. Bariéry jsou znázorněny černou linií a výsledný spline je vizualizován červenou barvou (viz. kapitola výsledky).

4. Výsledky

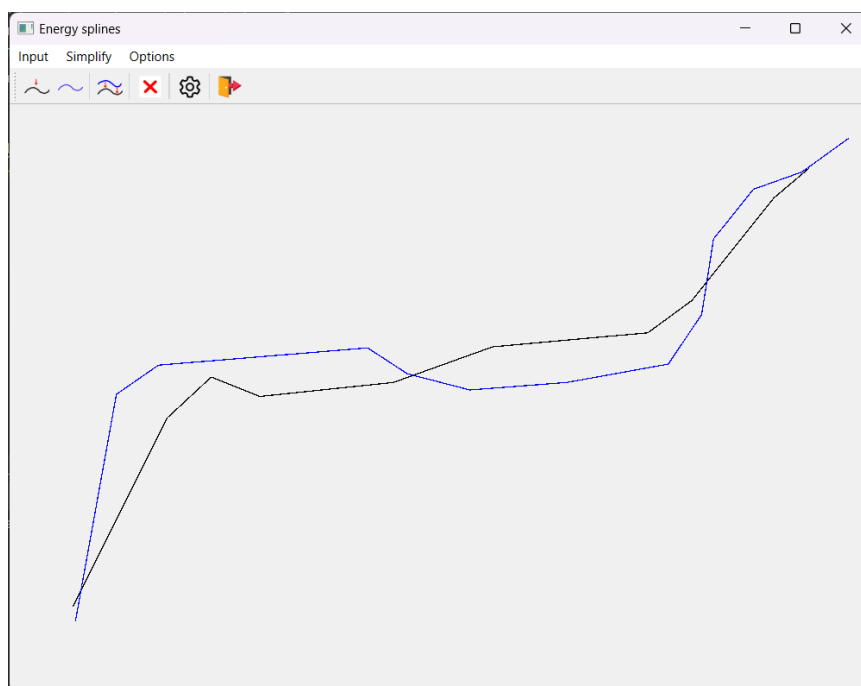
Pro otestování funkcionality aplikace vstupní testovací data, která představují protínající se silnici s vodním tokem. Konkrétně byla vybrána silnice spojující obce Kněžves, Tuchoměřice a Statenice na pomezí severozápadní části Prahy, kterou na více místech protíná Únětický potok. Jako data byla užity vrstvy náležící geodatabázi ArcČR_500.



Obrázek 1: Lokalizovaná vybraná testovací data

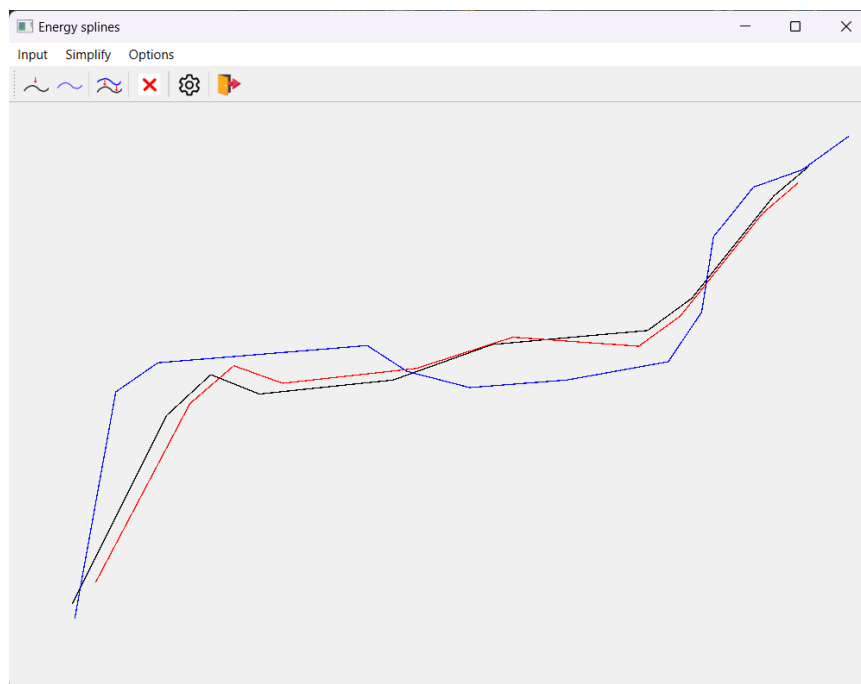


Obrázek 2: Defaultní nastavení parametrů



Obrázek 3: Nahrané vrstvy prvku a bariéry

Algoritmus Partial Modification modifikoval jeden prvek (v tomto případě silnici) vůči bariéře (vodnímu toku) při zachování výchozích parametrů aplikace.



Obrázek 4: Výsledný energetický spline

5. Závěr

Cílem této úlohy byl vytvořit aplikaci, jež dokáže vybraný vstupní prvek posunout logicky vzhledem k zadané bariéře pomocí energetických splinů, tak aby nedocházelo k jejich vzájemné grafické kolizi. Uživatel může do aplikace nahrát svá vlastní data, rovněž může nastavit hodnotu parametrů, jež tvoří samotné energetické spliny, dle svých potřeb. Aplikace obsahuje pouze parciální modifikaci pro jeden prvek s překážkou, pro lepší funkčnost a aplikovatelnost by bylo však vhodné program vylepšit pro práci s více prvky a více bariérami.

Energetické spliny jsou však velmi efektivní a značně pokročilou metodou generalizace, jež představuje jedno z velmi atraktivních témat současné digitální kartografie.

6. Použité zdroje

BAYER, T. (2008): Algoritmy v digitální kartografii. Univerzita Karlova v Praze, Nakladatelství Karolinum, Praha, 251 s.

BAYER, T. (2022): Úvod do výpočetní geometrie, základní vztahy. Prezentace k předmětu Algoritmy počítačové kartografie. Dostupné z: <http://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk2.pdf> (cit. 21.5.2023).

BAYER, T. (2023a): Kartografické generalizační algoritmy. Prezentace k předmětu *Algoritmy počítačové kartografie*. Dostupné z: http://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk8_new.pdf (cit. 21.5.2023).

BAYER, T. (2023b): Partial displacement, energetické spliny: stručný návod na cvičení. Dostupné z: http://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk_ems.pdf (cit. 21.5.2023).