



ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

Úloha č. 4: **Energetické spliny**

Bc. Taťána Bláhová, Bc. Tomáš Krauz, Bc. Adéla Kučerová

17. ledna 2023

Obsah

1	Zadání	2
2	Popis a rozbor problému energetických splinů	2
2.1	Operace Partial Displacement	5
2.2	Energetické spliny - 1 prvek, 1 bariéra	6
2.3	Energetické spliny - 2 prvky, žádná bariéra	7
3	Funkce aplikace	8
4	Vzhled aplikace	9
5	Vstupní data	10
6	Výstupní data	10
7	Problematické situace a jejich rozbor	11
8	Dokumentace	12
8.1	Třída Algorithms	12
8.2	Třída Draw	13
8.3	Třída CSV	14
8.4	Třída Mainform	14
9	Závěr	15

1 Zadání

Úloha č. 4: Energetické spliny

S využitím generalizačního operátoru Partial Modification realizujte odsun a částečnou změnu tvaru

- jednoho blízkého prvku vůči pevnému prvku (bariéře),
- dvou blízkých prvků vzájemně,

aby v generalizované mapě nedošlo k jejich grafickému konfliktu. Hodnotu minimální vzdálenosti prvků 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.

Hodnocení:

Krok	Hodnocení
Partial Modification: 1 prvek, 2 prvky	20b
Partial Modification: 2 prvky a překážka	+10b
Partial Modification: 3 prvky	+15b
Variabilní počet iterací, polohová chyba menší než grafická přesnost mapy.	+10b
Max celkem:	55b

Čas zpracování: 3 týdny

Obrázek 1.1: Zadání úlohy

2 Popis a rozbor problému energetických splinů

Úloha je založena na využití generalizačního operátoru Partial modification pro realizování odsunu či částečnou změnu tvaru jednoho prvku vůči pevné bariéře, nebo dvou prvků vzájemně. Tímto odsunem má dojít k odstranění možného grafického konfliktu. Je zde počítáno s jistou minimální vzdáleností prvků. Tato vzdálenost je volena v závislosti na měřítkovém čísle mapy (např. 1 mm v mapě). Implementace je provedena metodou energetických splinů.

Energetický spline je používán v oblasti PC grafiky, kde pro potřeby kartografické generalizace došlo k úpravě definice splinu. Operátor částečné modifikace (partial modification) je tak reprezentován parametrickou křivkou:

$$d(s) = (s(x) - x_{(0)}(s), y(s) - y_{(0)}(s)), \quad (2.1)$$

kde $x(0)$, $y(0)$ představují souřadnice vstupního prvku a souřadnice x, y jsou souřadnicemi výstupního prvku. Pro parametr s platí: $s \in \langle 0, 1 \rangle$.

Příslušný energetický model pro křivku L a její délku l má tvar

$$E(d) = \int_l E_i(s) ds + \int_l E_e(s) ds, \quad (2.2)$$

a zahrnuje vnitřní energii splinu E_i (internal energy) a vnější energii E_e (external energy), které ovlivňují tvar splinu. Cílem metody je nalézt jednu z mnoha deformací, která minimalizuje celkovou energii splinu. Výsledný spline musí zohledňovat vnitřní energii (tj. mechanické vlastnosti materiálu), tak i působení vnějších sil. Vnitřní spline zajišťuje, aby nebyl modifikován vnějšími silami.

Vnitřní energie splinu je definována vztahem [1] a ovlivňuje průběh splinu a jeho tvar.

$$E_i(s) = \frac{1}{2} \left(\alpha(s) \|d(s)\|^2 + \beta(s) \left\| \frac{\partial d(s)}{\partial s} \right\|^2 + \gamma(s) \left\| \frac{\partial^2 d(s)}{\partial s^2} \right\|^2 \right), \quad (2.3)$$

První člen měří vzdálenost mezi splinem a původním elementem, druhý elasticitu splinu a poslední tuhost splinu. Vliv těchto faktorů je modelován s využitím trojice parametrů $\alpha(s), \beta(s), \gamma(s) \in \mathbb{R}^+$. Spline tedy může více či méně sledovat původní prvek a kopírovat jeho tvar [2], [3].

Vnější energie řídí deformaci způsobenou vnějšími silami. Energetická funkce popisující silový model má mnoho podob. Z matematického pohledu by měla být spojitá v bodě, diferencovatelná a mít jednoduchý průběh bez oscilací. Čím větší jsou funkční hodnoty, tím silnější je jejich vliv na deformaci tvaru. Z kartografické generalizace existuje způsob, jak navrhnout energetickou funkci, jejichž cílem je generalizační operace partial displacement, při kterém se snažíme nepřiblížit k jinému prvku na vzdálenost menší než d .

Minimalizace celkové energie,

$$E(d(s)) = \int_l F(s, d(s), d'(s), d''(s)) ds, \quad (2.4)$$

vede k využití Eulerovy-Lagrangeovy rovnice

$$\frac{\partial F}{\partial y} - \frac{\partial}{\partial x} \frac{\partial F}{\partial y'} + \frac{\partial^2}{\partial x^2} \frac{\partial F}{\partial y''} - \dots + (-1)^n \frac{\partial^n}{\partial x^n} \frac{\partial F}{\partial y^n} = 0 \quad (2.5)$$

Aplikací na $E(d)$ obdržíme

$$\frac{\partial F}{\partial y} = \alpha(s)d(s) + \nabla E_e(x(s), y(s)), \quad (2.6)$$

$$\frac{\partial}{\partial x} \frac{\partial F}{\partial y'} = \frac{\partial}{\partial s} \left(\beta(s) \frac{\partial d(s)}{\partial s} \right) = \beta(s) \frac{\partial^2 d(s)}{\partial s^2}, \quad (2.7)$$

$$\frac{\partial^2}{\partial x^2} \frac{\partial F}{\partial y''} = \frac{\partial^2}{\partial s^2} \left(\gamma(s) \frac{\partial^2 d(s)}{\partial s^2} \right) = \gamma(s) \frac{\partial^4 d(s)}{\partial s^4}, \quad (2.8)$$

optimální řešení má pak tvar

$$\alpha(s)d(s) + \beta(s) \frac{\partial^2 d(s)}{\partial s^2} - \gamma(s) \frac{\partial^4 d(s)}{\partial s^4} + \nabla E_e(x(s), y(s)) = 0, \quad (2.9)$$

Rozepsáním po složkách dostaneme soustavu dvou parciálních rovnic. Ale počítáme s tím, že hodnoty $\alpha(s), \beta(s), \gamma(s)$ jsou hodnoty konstantní při ekvidistantním kroku h , tak je možno získat Eulerovy rovnice v tomto tvaru

$$\alpha x + \beta \frac{\partial^2 x}{\partial s^2} - \gamma \frac{\partial^4 x}{\partial s^4} + \frac{\partial}{\partial x} E_e(x(s), y(s)) = 0 \quad (2.10)$$

$$\alpha y + \beta \frac{\partial^2 y}{\partial s^2} - \gamma \frac{\partial^4 y}{\partial s^4} + \frac{\partial}{\partial y} E_e(x(s), y(s)) = 0 \quad (2.11)$$

Ale pokud je spline vzorkován konstantním krokem h , můžeme použít diskrétní aproximaci. Proto je možné parciální derivace nahradit centrálními derivacemi.

Pokud je vložíme do Eulerových rovnic, získáme soustavu lineárních rovnic

$$\alpha x_i + \beta(x_{i-1} - 2x_i + x_{i+1}) + \beta(x_{i-1} - 2x_i + x_{i+1}) + \gamma(x_{i-2} - 4x_{i-1} + 6x_i - 4x_{i+1} + x_{i+2}) + E_{e,x} \quad (2.12)$$

$$\alpha y_i + \beta(y_{i-1} - 2y_i + y_{i+1}) + \beta(y_{i-1} - 2y_i + y_{i+1}) + \gamma(y_{i-2} - 4y_{i-1} + 6y_i - 4y_{i+1} + y_{i+2}) + E_{e,y} \quad (2.13)$$

Hodnoty $E_{e,x}, E_{e,y}$ představují parciální derivace vnější energie podle proměnných x_i, y_i . Z toho vyplývá, že spline musí být vzorkován nejméně pěti body $p_{i-2}, p_{i-1}, p_i, p_{i+1}, p_{i+2}$ a tak příslušná maticová reprezentace má tvar

$$A \triangle X + E_{e,x} = 0, \quad (2.14)$$

$$A \triangle Y + E_{e,y} = 0, \quad (2.15)$$

kde A je pentadiagonální matice

$$\begin{bmatrix} a & b & c & 0 & \cdots & 0 & 0 & 0 & 0 \\ b & a & b & c & \cdots & 0 & 0 & 0 & 0 \\ c & b & a & b & \cdots & 0 & 0 & 0 & 0 \\ 0 & c & b & a & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a & b & c & 0 \\ 0 & 0 & 0 & 0 & \cdots & b & a & b & c \\ 0 & 0 & 0 & 0 & \cdots & c & b & a & b \\ 0 & 0 & 0 & 0 & \cdots & 0 & c & b & a \end{bmatrix}, \quad (2.16)$$

jejíž prvky vypočítáme z

$$a = \alpha + \frac{2\beta}{h^2} + \frac{6\gamma}{h^4}, \quad b = -\frac{\beta}{h^2} - \frac{4\gamma}{h^4}, \quad c = \frac{\gamma}{h^4} \quad (2.17)$$

Protože je matice A singulární, tak hodnoty $\triangle X_{(i)}, \triangle Y_{(i)}$ nemůžeme určit přímo a je potřeba je vypočítat pomocí iterace

$$\triangle X_{(i)} = (A + \lambda I)^{-1}(\lambda \triangle X_{i-1} - E_{e,x}), \quad (2.18)$$

$$\triangle Y_{(i)} = (A + \lambda I)^{-1}(\lambda \triangle Y_{i-1} - E_{e,y}), \quad (2.19)$$

kde

$$\triangle X_{(i)} = X_{(i)} - X_{(0)} \quad (2.20)$$

$$\Delta Y_{(i)} = Y_{(i)} - Y_{(0)} \quad (2.21)$$

jsou představeny jako souřadnicové rozdíly vrcholů splinu v jejich i -té iteraci a počáteční aproximací představované lomenou čarou. Pro $i = 0$, platí

$$\Delta X_{(0)} = Y_{(0)} = 0 \quad (2.22)$$

Parametr λ ovlivňuje rychlost konvergence iteračního procesu.

2.1 Operace Partial Displacement

Jedná se o generalizační operaci, která provádí komplexní korekci tvaru geometrické polohy a generalizačního prvku. Zahrnuje posun a změnu tvaru částí prvků, které se přiblíží k jinému prvku na určitou mez danou hodnotou \underline{d} . Tato generalizace se používá u prvků, které se nacházejí blízko sebe a může dojít k vzájemnému překrytí. A díky tomu existují základní generalizační schémata. V úloze byla použita tato 2 generalizační schémata.

1. Energetické spliny - 1 prvek, 1 bariéra
2. Energetické spliny - 2 prvky, žádná bariéra

Máme energetickou funkci,

$$E_e(x, y) = \begin{cases} c(1 - \frac{d}{\underline{d}}), & d < \underline{d}, \\ 0, & \text{jinak}, \end{cases} \quad (2.23)$$

kteřá je navržena tak, aby zabránila přiblížení dvou prvků ve vzdálenosti menší než \underline{d} . Můžeme \underline{d} chápat jako minimální vzdálenost při které nedojde ke kartografickému slití v měřítku digitalizace. Vzdálenost je měřena mezi vrcholy p_i jednoho prvku a liniovým segmentem druhého prvku. Konstanta $c, c \in R^+$ ovlivňuje hodnotu gradientu a reguluje spád funkce. Iterativní řešení diskretizované varianty splinu využívá parciální derivaci $E_e(x, y)$ dle x, y . Pokud $d < \underline{d}$ funkci přepíšeme do tvaru

$$E_e(x, y) = c \left(1 - \sqrt{\frac{(x - x_n)^2 + (y - y_n)^2}{\underline{d}}} \right), \quad (2.24)$$

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

$$\frac{\partial E_e(x, y)}{\partial x} = -c \frac{x - x_n}{d\underline{d}} \quad (2.25)$$

$$\frac{\partial E_e(x, y)}{\partial y} = -c \frac{y - y_n}{d\underline{d}} \quad (2.26)$$

Pro $d \geq \underline{d}$, parciální derivace mají tvar

$$\frac{\partial E_e(x, y)}{\partial x} = \frac{\partial E_e(x, y)}{\partial y} = 0 \quad (2.27)$$

Změna polohy vrcholů splinu, ovlivňující tvar splinu, probíhá pouze ve směru $p \rightarrow q_n$. Výsledné posuny vycházejí z požadavku vyrovnaného stavu mezi vnitřní a vnější energií.

2.2 Energetické spliny - 1 prvek, 1 bariéra

V tomto případě je do aplikace načtena právě jedna bariéra a jeden element, který je bariérou ovlivněn. Při spuštění funkce dojde k modifikaci pouze elementu, který je posunut. Bariéra je považována za pevný prvek, kterým pohnuto nebylo. Generalizovaný tvar je představován polynií $L = \{p_1, \dots, p_n\}$ s n vrcholy $p_i = [x_i, y_i]$, bariéra je polynie $B = [q_1, \dots, q_m]$ s m vrcholy $q_j = [x_j, y_j]$.

Energetická funkce

$$E_e(x_i, y_i) = \begin{cases} c(1 - \frac{d_i}{\underline{d}}), & d_i < \underline{d}, \\ 0, & \text{jinak}, \end{cases} \quad (2.28)$$

Vzdálenost d_i je měřena mezi vrcholem $p_i \in L$ a nejbližším bodem $q_n \in B$.

$$d_i = \sqrt{(x_i - x_n)^2 + (y_i - y_n)^2} \quad (2.29)$$

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

$$\frac{\partial E_e(x_i, y_i)}{\partial x_i} = -c \frac{x_i - x_n}{d_i \underline{d}} \quad (2.30)$$

$$\frac{\partial E_e(x_i, y_i)}{\partial y_i} = -c \frac{y_i - y_n}{d_i \underline{d}} \quad (2.31)$$

jinak

$$\frac{\partial E_e(x_i, y_i)}{\partial x_i} = \frac{\partial E_e(x_i, y_i)}{\partial y_i} = 0. \quad (2.32)$$

V rámci celého předzpracování je provedena konverze polylinie na maticovou podobu. Generalizovaný prvek L je popsán maticemi $X(n, 1)$ a $Y(n, 1)$, bariéra B maticemi $X'(m, 1)$ a $Y'(m, 1)$. Nejdříve se vypočítá krok h ze souřadnicových rozdílů.

$$\delta X = x_{i+1} - x_i, \quad \delta Y = y_{i+1} - y_i, \quad (2.33)$$

a vzdálenost mezi vrcholy generalizované polylinie

$$H = \sqrt{\|\delta X\|^2 + \|\delta Y\|^2} \quad (2.34)$$

Výpočet normy můžeme realizovat s využitím násobení po složkách. Výsledný krok h určíme jako střední hodnotu H .

Pro zadané hodnoty α, β, γ a krok h určíme hodnoty koeficientů a, b, c a naplní se matice A , která je během iteračního procesu konstantní. Pokud $B = A + \lambda I$, můžeme vypočítat inverzi B^{-1} . Pro iterační proces nejdříve musíme inicializovat hodnoty matic $X_{(0)} = X$ a $Y_{(0)} = Y$. Pro předem zadaný počet iterací vytvoříme matice $E_x(n, 1)$ a $E_y(n, 1)$ a vypočítáme hodnoty prvků. Pro každý bod $p_i \in L, p_i = [x_i, y_i]$ a nalezneme nejbližší bod $q_n \in B, q_n = [x_n, y_n]$, a vypočteme hodnoty parciálních derivací vnější energie $\frac{\partial E_e(x_i, y_i)}{\partial x_i}, \frac{\partial E_e(x_i, y_i)}{\partial y_i}$ v tomto bodě. Určíme hodnoty posunů $\Delta X_{(i)}, \Delta Y_{(i)}$

$$\Delta X_{(i)} = B^{-1}(\lambda \Delta X_{(i-1)} E_{e,x}), \quad \Delta Y_{(i)} = B^{-1}(\lambda \Delta Y_{(i-1)} E_{e,y}) \quad (2.35)$$

a vypočítáme souřadnice vrcholů splinu

$$X_{(i)} = X + \Delta X_{(i)}, \quad Y_{(i)} = Y + \Delta Y_{(i)} \quad (2.36)$$

Inkrementuje se index $i = i + 1$. Pokud i \geq maximální počet iterací, jdi na bod a) jinak ukončí iterační proces. Poté je provedena konverze z maticové reprezentace na spojový seznam vrcholů reprezentující generalizovanou polylinii.

2.3 Energetické spliny - 2 prvky, žádná bariéra

V tomto případě jsou do aplikace načteny dva elementy, které se vzájemně ovlivní a posunou se vůči sobě. Ani jeden z prvků není brán za pevný. První prvek je představován polynií $L = \{p_1, \dots, p_n\}$ s n vrcholy $p_i = [x_i, y_i]$, druhý prvek je polynie $L' = [q_1, \dots, q_m]$ s m vrcholy $q_j = [x_j, y_j]$. Energetická funkce zohledňuje vzájemný vliv obou prvků. Energetická funkce pro polylinii L' má tvar

$$E_e(x_i, y_i) = \begin{cases} c(1 - \frac{d_j}{\underline{d}}), & d_j < \underline{d}, \\ 0, & \text{jinak}, \end{cases} \quad (2.37)$$

kde d_j představuje vzdálenost mezi vrcholem $q_j \in L'$ a nejbližším bodem $p_n \in L$

$$d_j = \sqrt{(x_j - x_n)^2 + (y_j - y_n)^2} \quad (2.38)$$

$$\frac{\partial E_e(x_j, y_j)}{\partial x_j} = -c \frac{x_j - x_n}{d_j \underline{d}} \quad (2.39)$$

$$\frac{\partial E_e(x_j, y_j)}{\partial y_j} = -c \frac{y_j - y_n}{d_j \underline{d}} \quad (2.40)$$

jinak

$$\frac{\partial E_e(x_j, y_j)}{\partial x_j} = \frac{\partial E_e(x_j, y_j)}{\partial y_j} = 0. \quad (2.41)$$

Vrcholy obou polylinií, pro které platí $d_i < d$ nebo $d_j < \underline{d}$, se od sebe vzájemně posunují ve směrech $p_i \rightarrow q_n$ a $q_j \rightarrow p_n$.

Výpočty parciální derivací energetické funkce je nutné mít tak, aby na změny vrcholu první polylinie mohla reagovat i polylinie druhá. Takže každá polylinie bude mít vlastní matici A . Generalizovaný prvek L je popsán maticemi $X(n, 1)$ a $Y(n, 1)$, generalizovaný prvek L' je popsán maticemi $X'(m, 1)$ a $Y'(m, 1)$. Vypočte se krok h ze souřadnicových rozdílů.

$$\delta X = x_{i+1} - x_i, \quad \delta Y = y_{i+1} - y_i, \quad \delta X' = x_{j+1} - x_{ij}, \quad \delta Y' = y_{j+1} - y_{ij}, \quad (2.42)$$

a vzdálenosti mezi vrcholy obou generalizovaných polylinií

$$H = \sqrt{\|\delta X\|^2 + \|\delta Y\|^2}, \quad H' = \sqrt{\|\delta X'\|^2 + \|\delta Y'\|^2} \quad (2.43)$$

Výsledné kroky h, h' určíme jako střední hodnoty H, H' .

Pro zadané hodnoty α, β, γ a krok h, h' určíme hodnoty koeficientů a, b, c a naplní se matice A, A' , které jsou během iteračního procesu konstantní. Pokud $B = A + \lambda I$, $B' = A' + \lambda I$, můžeme vypočítat inverzi B^{-1} . Pro iterační proces nejdříve musíme inicializovat hodnoty matic $X_{(0)} = X$ a $Y_{(0)} = Y$

$$X_{(0)} = X \quad Y_{(0)} = Y \quad X'_{(0)} = X' \quad Y'_{(0)} = Y' \quad (2.44)$$

Pro předem zadaný počet iterací vytvoříme matice $E_x(n, 1)$, $E_x(n, 1)$, $E'_x(m, 1)$ a $E'_x(m, 1)$ a vypočítají se hodnoty prvků. Pro každý bod $p_i \in L$, $p_i = [x_i, y_i]$ a nalezneme nejbližší bod $q_n \in B$, $q_n = [x_n, y_n]$, a vypočteme hodnoty parciálních derivací vnější energie $\frac{\partial E_e(x_i, y_i)}{\partial x_i}$, $\frac{\partial E_e(x_i, y_i)}{\partial y_i}$ v tomto bodě. Pro každý bod $q_j \in L$, $q_j = [x_j, y_j]$ a nalezneme nejbližší bod $p_n \in B$, $p_n = [x_n, y_n]$, a vypočteme hodnoty parciálních derivací vnější energie $\frac{\partial E_e(x_j, y_j)}{\partial x_j}$, $\frac{\partial E_e(x_j, y_j)}{\partial y_j}$ v tomto bodě. Určíme hodnoty posunů $\Delta X_{(i)}$, $\Delta Y_{(i)}$

$$\Delta X_{(i)} = B^{-1}(\lambda \Delta X_{(i-1)} E_{e,x}), \quad \Delta Y_{(i)} = B^{-1}(\lambda \Delta Y_{(i-1)} E_{e,y}) \quad (2.45)$$

A určíme hodnoty posuny $\Delta X'_{(i)}$, $\Delta Y'_{(i)}$

$$\Delta X'_{(i)} = B'^{-1}(\lambda \Delta X'_{(i-1)} E'_{e,x}), \quad \Delta Y'_{(i)} = B'^{-1}(\lambda \Delta Y'_{(i-1)} E'_{e,y}) \quad (2.46)$$

a vypočítáme souřadnice vrcholů splinu.

$$X_{(i)} = X + \Delta X_{(i)}, \quad Y_{(i)} = Y + \Delta Y_{(i)} \quad X'_{(i)} = X' + \Delta X'_{(i)}, \quad Y'_{(i)} = Y' + \Delta Y'_{(i)} \quad (2.47)$$

Inkrementuje se index $i = i + 1$. Pokud je i menší než maximální počet iterací, jde na bod a), jinak ukončí iterační proces. Poté je provedena konverze z maticové reprezentace na spojový seznam vrcholů reprezentující generalizovanou polylinii.

Principem energetických splinů je realizace odsunu či změna tvaru prvků vůči jinému za účelem odstranění grafického konfliktu. K výpočtu je brána v potaz minimální vzdálenost dotčených prvků, která má být mezi objekty.

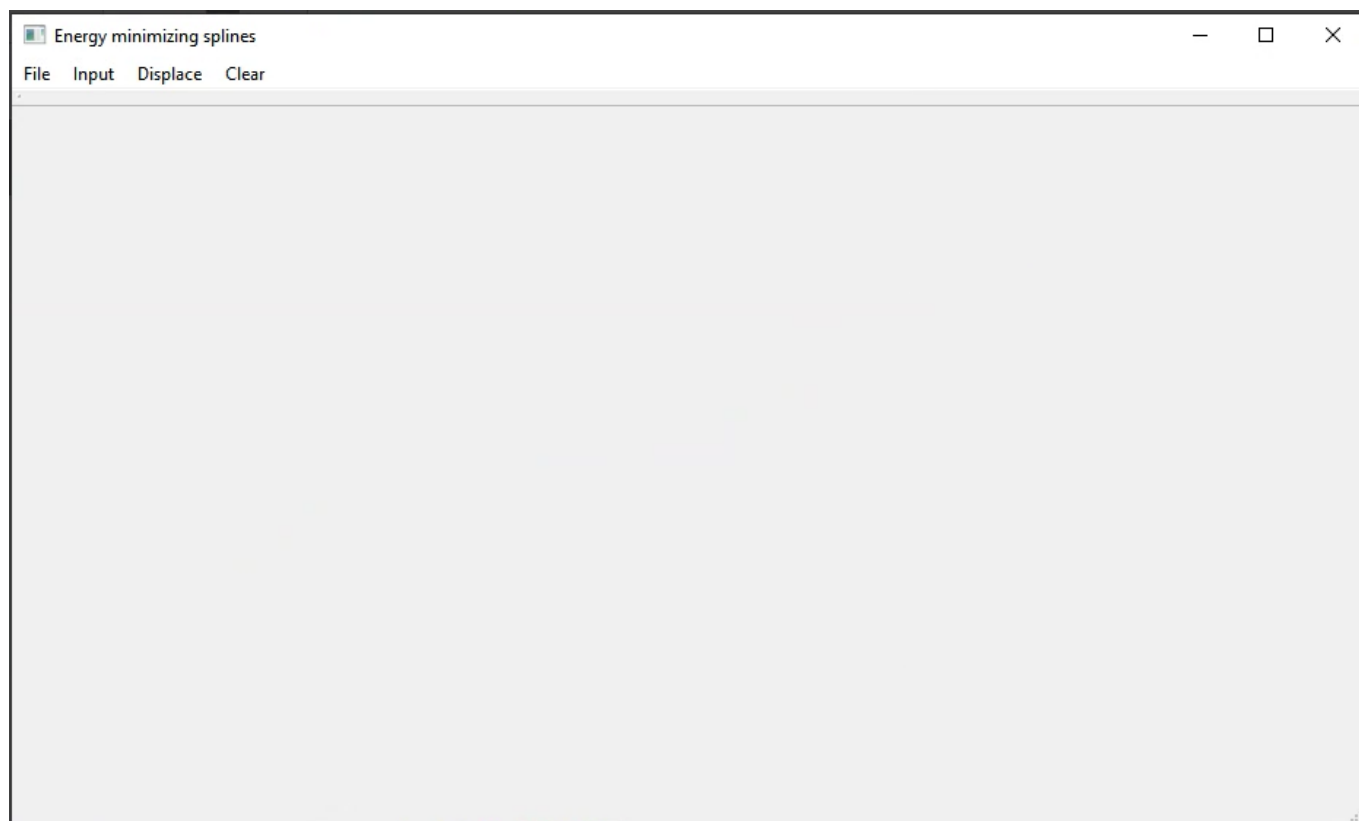
Do výpočtu vstupuje vnitřní a vnější energie splinu, které ovlivňují tvar splinu. Jak je uvedeno výše, cílem metody je nalezení minimální celkové energie splinu. Pokud je spline vzorkován s konstantním krokem h , lze použít také jeho diskretní aproximaci, která je pro praktické výpočty vhodnější. Parciální derivace tak lze nahradit centrálními diferencemi.

3 Funkce aplikace

Výsledkem jsou jedna nebo více posunutých křivek v závislosti na vybrané metodě. Vstupními daty jsou existující kartografická data (liniové prvky). Využitím generalizačního operátu Partial Modification dojde k posouzení jejich vzájemného umístění. V závislosti na vzdálenosti lomových bodů a typu prvku (bariéra, prvek) je buď zachován stávající stav, pokud jsou od sebe prvky dostatečně vzdálené, nebo dojde k posunu prvku elementu, nebo prvků vzájemně, jedná-li se o dva elementy.

4 Vzhled aplikace

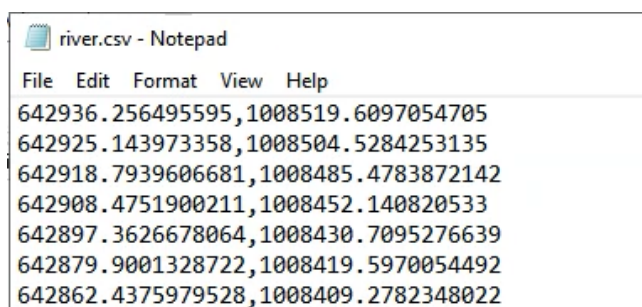
Po spuštění aplikace se zobrazí následující okno, kde je možné provést tyto akce - načíst CSV soubor s prvkem a bariérou (File), změnit aktuálně ručně kreslený prvek (Input), zvolit metodu pro odsunutí (Displace) a vyčistit Canvas (Clear).



Obrázek 4.1: Vzhled aplikace

5 Vstupní data

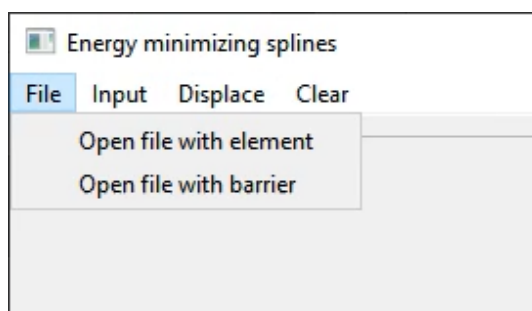
Aplikace očekává na vstupu CSV soubor se souřadnicemi bodů příslušného prvku v souřadnicovém systému S-JTSK.



```
river.csv - Notepad
File Edit Format View Help
642936.256495595,1008519.6097054705
642925.143973358,1008504.5284253135
642918.7939606681,1008485.4783872142
642908.4751900211,1008452.140820533
642897.3626678064,1008430.7095276639
642879.9001328722,1008419.5970054492
642862.4375979528,1008409.2782348022
```

Obrázek 5.1: Formát vstupního souboru

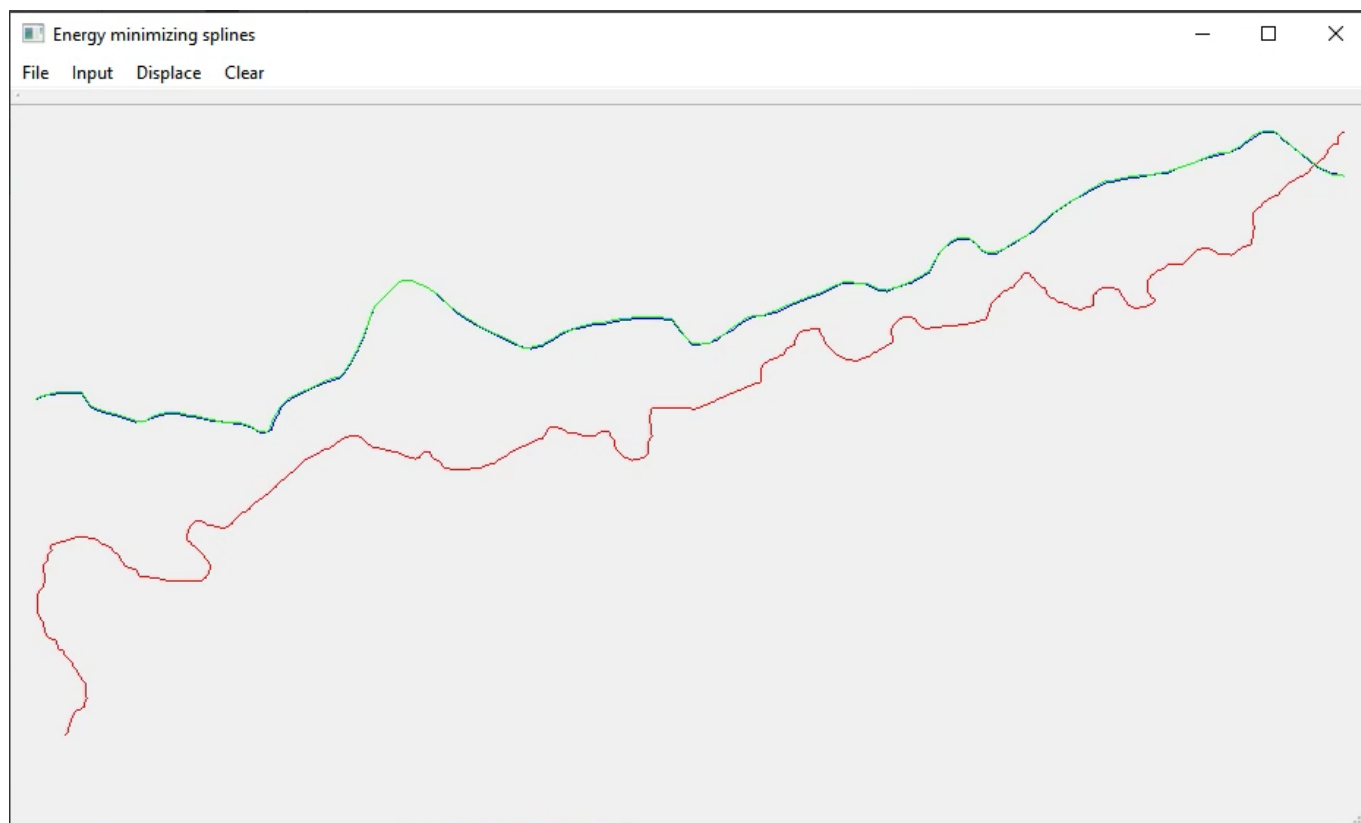
Pomocí tlačítka File je možné postupně načíst soubor s prvkem a bariérou, přičemž nezáleží na pořadí, v jakém uživatel soubory načte.



Obrázek 5.2: Načtení vstupních souborů

6 Výstupní data

Po kliknutí na zvolenou metodu aplikace provede modifikaci buďto jednoho nebo dvou prvků (resp. prvku i bariéry). V případě zvolení metody *Displace 2 elements* se soubor (linie) načtený jako bariéra chová jako prvek a je také posunut.



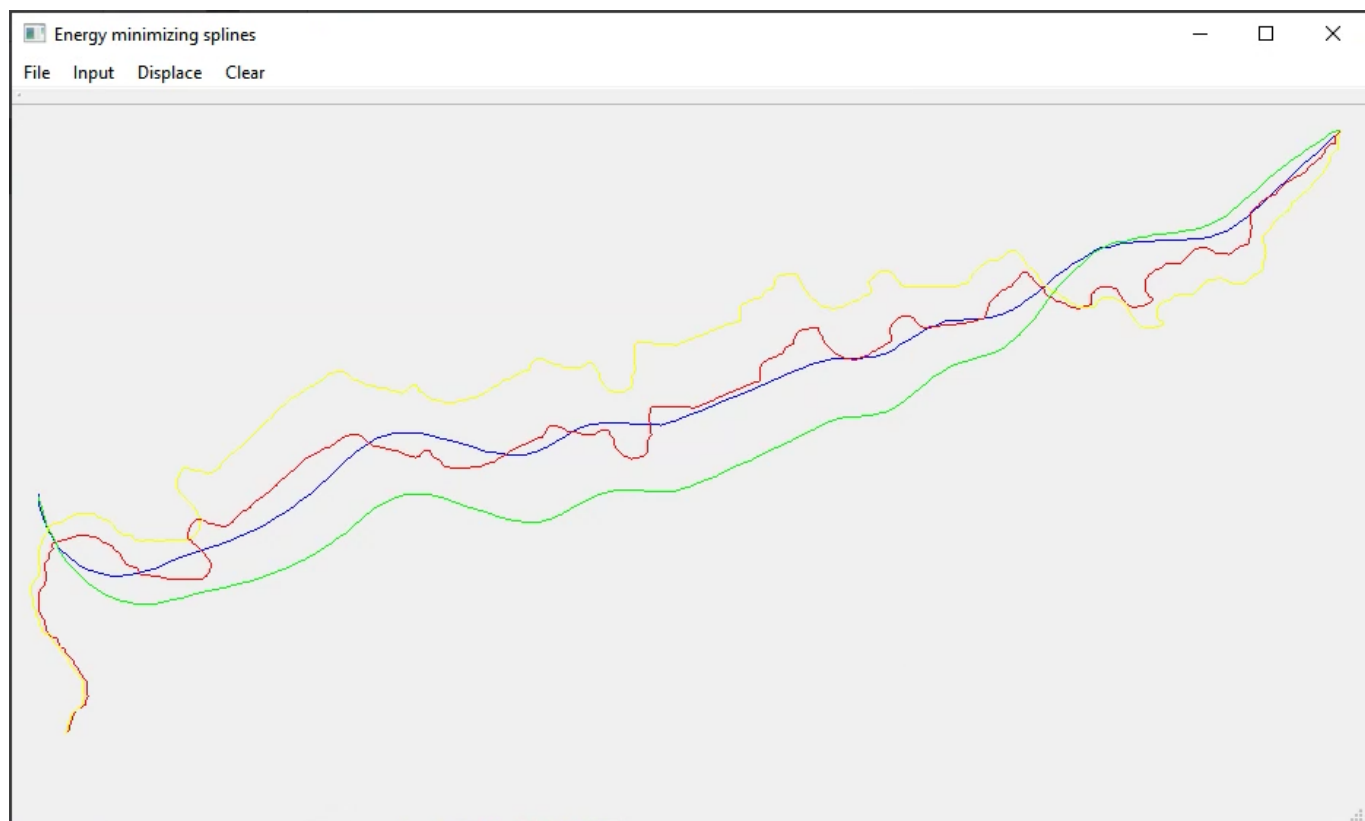
Obrázek 6.1: Modifikace 1 prvku (element) - nová poloha prvku vykreslena zeleně

7 Problematické situace a jejich rozbor

Při spuštění aplikace nefunguje obvyklé vytvoření linií pomocí ručního klikání a funkce je možné spustit po nahrání externího souboru. Tento problém nebyl prakticky vyřešen. Teoretickým řešením by zřejmě byla duplikace a úprava kódu pro samostatné fungování pro nahrání souboru a ručního klikání společně s toggle tlačítkem, kde by došlo k přepínání mezi těmito dvěma kódy.

Aplikace spadne při pokusu o zavření dialogového okna "Open file with points", důvodem je neošetření varianty, kdy nebyl zvolen žádný soubor.

Do aplikace je možnost opakovaného nahrání souboru pro element či bariéru. Tento problém nebyl ošetřen, jelikož se předpokládá, že uživatel nahraje pouze nutné soubory pro element a bariéru, a následně spustí funkci Displace.



Obrázek 6.2: Modifikace 2 prvků (element i bariéra) - nová poloha prvků vykreslena zeleně a žlutě

8 Dokumentace

8.1 Třída Algorithms

- *double getEuclDistance(double x1, double y1, double x2, double y2)*
– Euklidovská vzdálenost dvou bodů
- *std::tuple<double, double, double> getPointLineDistance(double x, double y, double x1, double y1, double x2, double y2)*
– uložení vzdáleností bodu a linie
- *std::tuple<double, double, double> getPointLineSegmentDistance(double x, double y, double x1, double y1, double x2, double y2)*
– uložení vzdáleností bodu a segmentu linie
- *std::tuple<int, double, double, double> getNearestLineSegmentPoint(double xq, double yq, Matrix &X, Matrix &Y)*
– získání nejbližšího segmentu linie

- *Matrix createA(double alfa, double beta, double gamma, double h, int n)*
– vytvoření matice A
- *std :: vector < QPointF > minEnergySpline1Element1Barrier(std :: vector < QPointF > element, std :: vector < QPointF > barrier, double alfa, double beta, double gamma, double lambda, double dmin, int iter)*
– minimální energetické spliny mezi prvkem a bariérou
- *std :: tuple < std :: vector < QPointF >, std :: vector < QPointF >> minEnergySpline2Elements(std :: vector < QPointF > element1, std :: vector < QPointF > element2, double alfa, double beta, double gamma, double lambda, double dmin, int iter)*
– minimální energetické spliny mezi 2 elementy
- *double getEx(double x, double y, double xn, double yn, double dmin)*
– získá souřadnice Ex
- *double getEy(double x, double y, double xn, double yn, double dmin)*
– získá souřadnice Ey
- *static std :: vector < QPointF > transformPoints(std :: vector < QPointF > &points, double &trans_x, double &trans_y, double &scale, int &offset_x, int &offset_y)*
– transformuje souřadnice bodů pro tvorbu min-max boxu

8.2 Třída Draw

- *void mousePressEvent(QMouseEvent * event)*
– vrátí souřadnice kurzoru po kliknutí na Canvas
- *void paintEvent(QPaintEvent * event)*
– vykreslí body na Canvas
- *void switchDrawnElement(){draw_element != draw_element;}*
– změna typu kreslení bariéra/prvek
- *std :: vector < QPointF > getElement(){return pointsElement;}*
– vrátí prvek
- *std :: vector < QPointF > getBarrier(){return pointsBarrier;}*
– vrátí bariéru
- *std :: vector < QPointF > getDisplaced(){return displaced;}*
– posun
- *void setElement(std :: vector < QPointF > &element_){element = element_;}*
– nastavení prvku
- *void setBarrier(std :: vector < QPointF > &barrier_){barrier = barrier_;}*
– nastavení bariéry
- *void setDisplaced(std :: vector < QPointF > &displaced_){displaced = displaced_;}*
– nastavení posunů 1 prvku
- *void setDisplaced2(std :: vector < QPointF > &displaced2_){displaced2 = displaced2_;}*
– nastavení posunů 2 prvků
- *void drawCSVPointsElement(std :: vector < QPointF > &pointsElement);*
– vykreslí načtené body prvku na Canvas
- *void drawCSVPointsBarrier(std :: vector < QPointF > &pointsBarrier);*
– vykreslí načtené body bariéry na Canvas

- *void clearElement(){element.clear();}*
– vymaže prvek z Canvasu
- *void clearBarrier(){barrier.clear();}*
– vymaže bariéru z Canvasu
- *void clearDisplaced(){displaced.clear();}*
– vymaže odsunutý prvek z Canvasu
- *void clearDisplaced2(){displaced2.clear();}*
– vymaže odsunutou bariéru (druhý prvek) z Canvasu
- *void clearPointsElement(){pointsElement.clear();}*
– vymaže načtené body prvku z Canvasu
- *void clearPointsBarrier(){pointsBarrier.clear();}*
– vymaže načtené body bariéry z Canvasu

8.3 Třída CSV

- *static std :: vector < std :: vector < std :: string >> readCSV(std :: string &filename);*
– načte vstupní CSV soubor
- *static std :: vector < QPointF > getPoints(std :: vector < std :: vector < std :: string >> &csv_content, double &x_min, double &x_max, double &y_min, double &y_max);*
– vrací vektor QPointF z CSV souboru

8.4 Třída Mainform

- *void on_actionOpen_element_triggered();*
– akce pro načtení vstupního souboru s elementem
- *void on_actionOpen_barrier_triggered();*
– akce pro načtení vstupního souboru s bariérou
- *void on_actionElement_Barrier_changed();*
– akce po stisknutí tlačítka s prvkem a bariérou
- *void on_actionDisplace_triggered();*
– akce po stisknutí tlačítka Displace 1 element/Displace 2 elements
- *void on_actionDisplace_2_elements_triggered();*
– akce po stisknutí tlačítka se 2 prvky
- *void on_actionClear_all();*
– akce pro vyčištění Canvasu

9 Závěr

Byla vytvořena aplikace pro realizaci odsunu a částečnou změnu tvaru elementu pomocí generalizačního operátoru Partial Modification, aby se předešlo grafickému konfliktu. Aplikace byla napsána pro stavy prvek a prvek, nebo prvek a bariéra, kdy modifikovaným prvkem je vždy element a bariéra zůstane beze změny.

Literatura

- [1] M. Bader, "Energy minimization methods for feature displacement in map generalization," dis. pr., University of Zurich Zurich, 2001.
- [2] M. Kass, A. Witkin a D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, roč. 1, č. 4, s. 321–331, 1988.
- [3] D. Burghardt a S. Meier, "Cartographic displacement using the snakes concept," *Semantic modeling for the acquisition of topographic information from images and maps*, Basel, Birkhäuser Verlag, s. 59–71, 1997.