

Matematická kartografie

Úloha 1 Digitální model terénu

František Macek, Josef Zátka



Univerzita Karlova
Přírodovědecká fakulta
Katedra aplikované geoinformatiky a kartografie

Praha, 2025

1 Zadání

Vstup: množina $P = \{p_1, \dots, p_n\}$, $p_i = \{x_i, y_i, z_i\}$.

Výstup: polyedrický DMT nad množinou P představovaný vrstevnicemi doplněný vizualizací sklonu trojúhelníků a jejich expozicí.

Metodou inkrementální konstrukce vytvořte nad množinou P vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů), popř. navrhněte algoritmus pro generování syntetických vstupních dat představujících významné terénní tvary (kupa, údolí, spočinek, hřbet, ...).

Vstupní množiny bodů včetně níže uvedených výstupů vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku **QT**. Dynamické datové struktury implementujte s využitím **STL**.

Nad takto vzniklou triangulací vygenerujte polyedrický digitální model terénu. Dále proveďte tyto analýzy:

- S využitím lineární interpolace vygenerujte vrstevnice se zadaným krokem a v zadaném intervalu, proveďte jejich vizualizaci s rozlišením zvýrazněných vrstevnic.
- Analyzujte sklon digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich sklonu.
- Analyzujte expozici digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich expozici ke světové straně.

Zhodnoťte výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabými místy algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích (různé terénní tvary) nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

Zhodnocení činnosti algoritmu včetně ukázek proveďte alespoň na 3 strany formátu A4.

1.1 Hodnocení

Krok	Hodnocení
Delaunay triangulace, polyedrický model terénu	10 b
Konstrukce vrstevnic, analýza sklonu a expozice	10 b
Triangulace nekonvexní oblasti zadané polygonem	+5 b
Výběr barevných stupnic při vizualizaci sklonu a expozice	+3 b
Automatický popis vrstevnic	+3 b
Automatický popis vrstevnic respektující kartografické zásady (orientace, vhodné rozložení)	+10 b
Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet, ...)	+10 b
3D vizualizace terénu s využitím promítání	+10 b
Barevná hypsometrie	+5 b
Maximálně celkem	65 b

Čas zpracování: 4 týdny

2 Úvod

Digitální model terénu (DMT) představuje digitální reprezentaci skutečného povrchu Země ve formě dvourozměrného nebo trojrozměrného modelu. Tento model je tvořen na základě reálných, naměřených dat, která jsou doplněna interpolačními metodami sloužícími k odhadu hodnot v místech s chybějícími měřeními. Digitální model terénu však nepředstavuje pouze reprezentaci zemského povrchu, ale zahrnuje také řadu kartografických technik, které slouží k vizualizaci průběhu terénu. Mezi tyto techniky patří například izočáry, stínování terénu, terénní šrafy nebo znázornění sklonu svahů. V rámci modelovaného území lze následně určit nadmořskou výšku v libovolném bodě.

Samotná implementace probíhala nad daty z laserového skenování v jazyce Python, s využitím knihoven PyQt6 a Laspy s bindy Lazrs (využito na import dat). Byl implementován polyedrický model terénu pomocí Delaunayho triangulace, dále byla provedena konstrukce vrstevnic a analýza sklonu a expozice.

3 Tvorba DMT

Konstrukce digitálního modelu terénu (DMT)

Při tvorbě digitálního modelu terénu je dána množina bodů $P = \{p_1, p_2, \dots, p_n\} \subset \mathbb{R}^2$, která reprezentuje výšková měření povrchu. Cílem je vytvořit triangulaci T , tedy síť trojúhelníků nad množinou P .

3.1 Definice triangulace

Triangulace T nad množinou bodů P je takové planární rozdělení, které tvoří množinu trojúhelníků $T = \{t_1, t_2, \dots, t_m\}$, přičemž platí následující:

- Libovolné dva trojúhelníky sdílejí nejvýše jednu společnou hranu.
- Žádný bod z množiny P neleží uvnitř jiného trojúhelníku.
- Sjednocení všech trojúhelníků tvoří konvexní obal $H(P)$.

3.2 Požadavky na triangulaci T

Požadavky na triangulační algoritmus jsou následující:

- Jednoduchost algoritmu a snadná implementace,
- Výpočetní efektivita pro velké objemy dat, ideální složitost: $\mathcal{O}(n \log n)$.
- Malá citlivost na singulární případy.
- Možnost škálování do vyšších dimenzí (3D)
- Schopnost paralelizace algoritmu.
- Optimální tvar trojúhelníkové sítě.

V praxi jsou tyto požadavky často v kontrastu, zejména jednoduchost implementace a výpočetní efektivita. Triangulační algoritmy patří mezi velmi dobře teoreticky prozkoumané metody. V praxi se nejčastěji využívá Delaunayova triangulace, která byla využita i pro tuto úlohu.

3.3 Typy triangulací

Triangulace lze rozdělit podle následujících hledisek:

Metod geometrické konstrukce:

- Greedy triangulace.
- Delaunay triangulace.
- MWT (Minimum Weight Triangulation).
- Constrained triangulace (s povinnými hranami).
- Datově závislé triangulace.

Optimalizačních kritérií:

- **Lokálně optimální triangulace:** Optimalizace na úrovni sousedních dvojic trojúhelníků.
- **Globálně optimální triangulace:** Celá síť optimalizována vzhledem ke zvolenému kritériu (současně i lokálně optimální).
- **Multikriteriálně optimalizované triangulace:** Kombinace několika lokálních i globálních kritérií (dosud nejsou známy efektivní algoritmy).

(Bayer 2025)

3.4 Delaunay triangulace

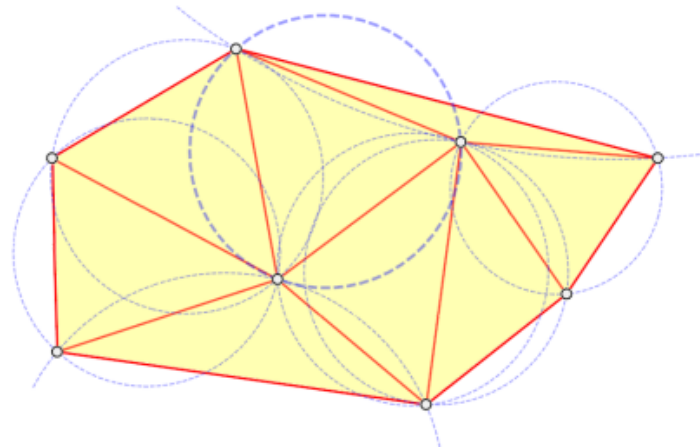
Delaunayho triangulace je nejčastěji používanou metodou při konstrukci digitálních modelů terénu a v oblasti GIS je de-facto standardem. Je definována jak v prostoru \mathbb{R}^2 , tak i v \mathbb{R}^3 .

$DT(P)$ splňuje následující vlastnosti:

- Kružnice opsaná libovolnému trojúhelníku $t_j \in DT$ neobsahuje žádný jiný bod množiny P .

- DT maximalizuje minimální úhel v trojúhelnících (neminimalizuje ale maximální úhel).
- DT je lokálně i globálně optimální vůči kritériu minimálního úhlu.
- Pokud žádné čtyři body neleží na jedné kružnici, pak je DT jednoznačná.

Trojúhelníky v DT se tedy nejvíce blíží rovnostranným trojúhelníkům. Na obrázku 1 je možné vidět ukázkou DT .



Obrázek 1: DT . (Bayer 2025)

3.5 Metody konstrukce Delaunay triangulace

Existuje několik metod konstrukce DT , které se liší výpočetní složitostí i vhodností pro různé typy dat.

3.5.1 Přímé metody

- Lokálního prohazování: $\mathcal{O}(n)$ + složitost triangulačního algoritmu
- Inkrementální konstrukce: $\mathcal{O}(n^2)$, modifikace algoritmu Bowyer-Watson dosahuje složitosti $\mathcal{O}(n \log n)$.
- Divide and Conquer: $\mathcal{O}(n \log n)$, Guibas-Stolfi a DeWall.
- Sweep Line algoritmus: $\mathcal{O}(n^2)$.

Mezi nepřímé metody patří například konstrukce přes Voronoi Diagram, která ale v praxi není téměř využívána. Naopak nejčastěji jsou využívány metody inkrementální konstrukce a divide and conquer. Právě inkrementální konstrukce byla využita v rámci této úlohy.

3.6 Inkrementální konstrukce

Tato metoda je vhodná pro 2D i 3D prostor (prázdná kružnice / prázdná sféra). Vychází z postupného přidávání bodů do již vytvořené DT .

Algorithm 1 Delaunay Triangulation Incremental(P, AEL, DT)

```

1:  $AEL = \{\}, DT = \{\}$ 
2:  $p_1 = \text{rand}(P)$ 
3:  $p_2 = \arg \min_{p_i \in P} \|p_1 - p_i\|^2$ 
4: Vytvoř hrany  $e = (p_1, p_2)$ ,  $e' = (p_2, p_1)$ 
5:  $AEL \leftarrow e$ ,  $AEL \leftarrow e'$ 
6: while  $AEL$  not empty:
7:    $e_1 = AEL.\text{pop}()$ ,  $e_1 = (p_1, p_2)$ 
8:    $e'_1 = (p_2, p_1)$ 
9:    $p' = \arg \max_{p_i \in \sigma_L(e'_1)} \angle(p_1, p_i, p_2)$ 
10:  if existuje  $p'$  then:
11:     $e_2 = (p_2, p')$ ,  $e_3 = (p', p_1)$ 
12:     $DT \leftarrow e'_1$ ,  $DT \leftarrow e_2$ ,  $DT \leftarrow e_3$ 
13:    UPDATEAEL( $e_2, AEL$ ), UPDATEAEL( $e_3, AEL$ )
14:  end if
15: end while

```

Algorithm 2 updateAEL($e = (a, b), AEL$)

```

1: Vytvoř hranu  $e' = (b, a)$ 
2: if  $e' \in AEL$  then:
3:    $AEL \rightarrow e$ 
4: else
5:    $AEL \leftarrow e$ 
6: end if

```

(Bayer 2025)

3.7 Konstrukce vrstevnic lineární interpolací

Vrstevnice digitálního modelu terénu byly vytvořeny lineární interpolací. Tato metoda je založena na analytické geometrii – hledá se průsečnice mezi rovinou trojúhelníku ($t \in DT$) a vodorovnou rovinou s danou výškou $z = h$. Tento postup je opakován pro každý trojúhelník v síti.

Výpočet průsečíku probíhá určením souřadnic bodů A a B , které leží na hranách trojúhelníku a zároveň v rovině vrstevnice. Pro výpočet lze využít podobnosti trojúhelníků:

$$\begin{aligned}x_a &= \frac{x_3 - x_1}{z_3 - z_1}(z - z_1) + x_1 & y_a &= \frac{y_3 - y_1}{z_3 - z_1}(z - z_1) + y_1 \\x_b &= \frac{x_2 - x_1}{z_2 - z_1}(z - z_1) + x_1 & y_b &= \frac{y_2 - y_1}{z_2 - z_1}(z - z_1) + y_1\end{aligned}$$

Pro zjištění, zda rovina vrstevnice dané výšky z protíná danou hranu trojúhelníku (p_i, p_{i+1}) , lze využít následující test:

$$\Delta z_i \cdot \Delta z_{i+1} < 0; \quad \text{kde } \Delta z_i = z_i - z, \quad \Delta z_{i+1} = z_{i+1} - z$$

Tento výraz určuje, zda rovina skutečně protíná výškový interval dané hrany.

3.8 Analýza sklonu terénu

Výpočet sklonu se provádí nad každým trojúhelníkem pomocí gradientu (vektor maximálního spádu).

Gradient $\nabla f(x_0, y_0, z_0)$ funkce $f(x, y, z)$ v bodě $p = [x_0, y_0, z_0]$ má tvar:

$$\nabla f(x_0, y_0, z_0) = \left(\frac{\partial f}{\partial x}(x_0), \frac{\partial f}{\partial y}(y_0), \frac{\partial f}{\partial z}(z_0) \right)$$

Rovnice roviny ρ má tvar:

$$\rho : \quad ax + by + cz + d = 0$$

Gradient $\nabla f(x_0, y_0, z_0)$ roviny ρ lze tedy zapsat následovně:

$$\nabla \rho(x_0, y_0, z_0) = \left(\frac{\partial \rho}{\partial x}(x_0), \frac{\partial \rho}{\partial y}(y_0), \frac{\partial \rho}{\partial z}(z_0) \right) = (a, b, c)$$

Odchylka φ mezi rovinami ρ a π , tedy hledaný sklon, se určí podle vztahu:

$$\varphi = \arccos \left(\frac{\mathbf{n}_1 \cdot \mathbf{n}_2}{\|\mathbf{n}_1\| \cdot \|\mathbf{n}_2\|} \right) = \arccos \left(\frac{c}{\|\mathbf{n}_1\|} \right)$$

$$\mathbf{n}_1 = (a, b, c), \quad \mathbf{n}_2 = (0, 0, 1)$$

3.9 Analýza expozice terénu

Expozice terénu je definována jako azimut průmětu gradientu $\nabla\rho$ do roviny xy :

$$\vec{v} = \left(\frac{\partial\rho}{\partial x}(x_0), \frac{\partial\rho}{\partial y}(y_0), 0 \right) = (a, b, 0)$$

Azimut A vektoru \vec{v} měřen od osy y určuje vztah:

$$A = \arctan\left(\frac{a}{b}\right)$$

4 Implementace

Implementace probíhala nad daty z laserového skenování v jazyce Python, s využitím knihoven PyQt6 a Laspy s bindy Lazrs (využito na import dat). Byl implementován polyedrický model terénu pomocí Delaunayho triangulace, dále byla provedena konstrukce vrstevnic a analýza sklonu a expozice. Výsledkem je uživatelské rozhraní.

Byly zpracovány následující úlohy:

Krok	Hodnocení
Delaunay triangulace, polyedrický model terénu	10 b
Konstrukce vrstevnic, analýza sklonu a expozice	10 b
Maximálně celkem	20 b

4.1 Vstupní data

Jako vstupní data byla vybrána data (ČÚZK 2025b), DMR 5G, vycházející z dat laserového skenování. Je možné data importovat z formátů .las, .laz a .txt. Pro formát .las je nezbytná instalace knihovny Laspy a pro data .laz instalace knihovny Laspy s bindy Lazrs. Pokud knihovny nejsou k dispozici, program uživatele upozorní. Stále však bude možné načíst data z formátu .txt.

Pro účely této úlohy byla vybrána data z mapového listu turn 04 a proveden random subsampling na 300 bodů v softwaru CloudCompare.

4.2 Třídy a metody

Funkční chod aplikace si vyžaduje několik povinných skriptů, v nichž jsou implementovány potřebné třídy a metody: `mainform.py`, `draw.py`, `algorithms.py`, `dialog.py`, `Edge.py`, `QPoint3DF.py` a `triangle.py`. Tato část textové zprávy je sepsána s pomocí umělé inteligence.

Třída `Ui_MainForm`

Třída `Ui_MainForm` ze souboru `mainform.py` inicializuje hlavní okno aplikace, nabídky, panelu nástrojů a propojuje jednotlivé komponenty s příslušnými metodami. Uživatel zde může například načíst soubor, spustit algoritmus triangulace nebo analyzovat sklon a orientaci. Část této třídy byla vygenerována v prostředí Qt Creator.

- `dtClick()` – Provádí Delaunay triangulaci a překreslí plátno.
- `contourLinesClick()` – Vygeneruje vrstevnice podle aktuálních parametrů.
- `settingsClick()` – Otevře dialogové okno pro nastavení výškových parametrů vrstevnic.
- `analyzeSlopeClick()` – Analyzuje sklon DTM a zobrazí jej.
- `analyzeAspectClick()` – Analyzuje orientaci DTM a zobrazí ji.
- `pointsChanged()` – Změní viditelnost bodů.
- `DTChanged()` – Změní viditelnost triangulace.
- `contourLinesChanged()` – Změní viditelnost vrstevnic.
- `slopeChanged()` – Změní viditelnost sklonu.
- `aspectChanged()` – Změní viditelnost orientace.
- `clearData()` – Vymaže výpočetní data z plátna.
- `clearAll()` – Vymaže všechna data včetně bodů.
- `closeApp()` – Ukončí aplikaci.
- `openFileDialog()` – Otevře soubor s mračnem bodů a načte jej.

Třída Draw

Třída `Draw` ze souboru `draw.py` dědí z `QWidget` a slouží jako vlastní vykreslovací plátno aplikace. Obsahuje metody pro vizualizaci bodů, triangulace, vrstevnic, sklonu a orientace trojúhelníků.

Hlavní metody:

- `pointsInput(points)` – Načte seznam bodů do plátna.
- `paintEvent(e)` – Metoda pro překreslení všech aktivních prvků.
- `clearData()` – Vymaže DTM, vrstevnice a trojúhelníky.
- `clearAll()` – Vymaže všechna data včetně bodů.
- `getAspectColor(aspect)` – Vrátí barvu odpovídající azimutu (aspektu) svahu.

Třída Algorithms

Třída `Algorithms` ze souboru `algorithms.py` obsahuje implementaci algoritmů pro Delaunay triangulaci, generování vrstevnic a analýzu sklonu a orientace DTM.

- `get2VectorsAngle()` – Vrací úhel mezi dvěma vektory ve 2D.
- `analyzePointAndLinePosition()` – Určí polohu bodu vůči přímce (vlevo, vpravo, na přímce).
- `distance2D()` – Vypočítá 2D vzdálenost mezi dvěma body.
- `getNearestPoint()` – Vrátil nejblíže bod k zadanému.
- `findDelaunayPoint()` – Najde třetí bod tvořící Delaunay trojúhelník.
- `delaunayTriangulation()` – Provádí Delaunay triangulaci metodou přidávání trojúhelníků.
- `update_ael()` – Aktualizuje aktivní seznam hran.
- `contourPoint()` – Interpoluje bod na hraně pro danou výšku vrstevnice.
- `createContourLines()` – Generuje vrstevnice ze seznamu hran Delaunay triangulace.
- `computeSlope()` – Vypočítá sklon trojúhelníku.
- `convertDTToTriangles()` – Převeď seznam hran na seznam trojúhelníků.
- `analyzeDTMSlope()` – Spočítá sklony všech trojúhelníků DTM.
- `computeAspect()` – Vypočítá orientaci (aspect) trojúhelníku.
- `analyzeDTMAsspect()` – Spočítá orientaci všech trojúhelníků DTM.

Třída Edge

Třída `Edge` ze souboru `edge.py` reprezentuje hranu v rovině, danou dvěma body typu `QPoint3DF`. Používá se zejména při práci s triangulací.

- `__init__(p1, p2)` – Vytvoří hranu mezi dvěma 3D body.
- `switch_orientation()` – Vrátil hranu s opačným směrem.
- `__eq__(other)` – Porovná dvě hrany podle počátečního a koncového bodu.

Třída `QPoint3DF`

Třída `QPoint3DF` ze souboru `qpoint3df.py` rozšiřuje `QPointF` o třetí souřadnici `z`. Používá se ke komplexnímu popisu bodu v 3D prostoru (např. při načítání mračna bodů).

- `__init__(x, y, z)` – Inicializuje bod s 3 souřadnicemi.
- `getZ()` – Vrátil výškovou souřadnici bodu.

Funkce `load_file`

Funkce `load_file()` ze souboru `read_laz_file.py` slouží k načítání mračna bodů z formátů `.las`, `.laz` a `.txt`. Zajistí také převod souřadnic do měřítka pro zobrazení na plátně. Pokud nejsou k dispozici knihovny na načtení `.las` nebo `.laz` souborů, vyvolá chybu.

- Podporuje formáty: `.las`, `.laz` (vyžaduje `laspy`, případně `lazrs`), `.txt`.
- Převádí souřadnice do plátna (škálování podle šířky a výšky).
- Vrací seznam bodů typu `QPoint3DF`.

Třída `Ui_Settings`

Třída `Ui_Settings` reprezentuje dialogové okno pro nastavení parametrů vrstevnic.

- `__init__()` – Inicializuje výchozí hodnoty parametrů: `zmin`, `zmax`, `dz`.
- `setupUi(Settings)` – Vytváří UI komponenty dialogového okna.
- `retranslateUi(Settings)` – Překládá texty prvků do lokalizovaného rozhraní.

Třída `Triangle`

Třída `Triangle` ze souboru `triangle.py` reprezentuje trojúhelník tvořený třemi body typu `QPoint3DF`. Uchovává také informace o sklonu a orientaci (azimutu) trojúhelníku.

- `__init__(p1, p2, p3, slope, aspect)` – Vytvoří trojúhelník ze tří bodů a nastaví jeho sklon a orientaci.

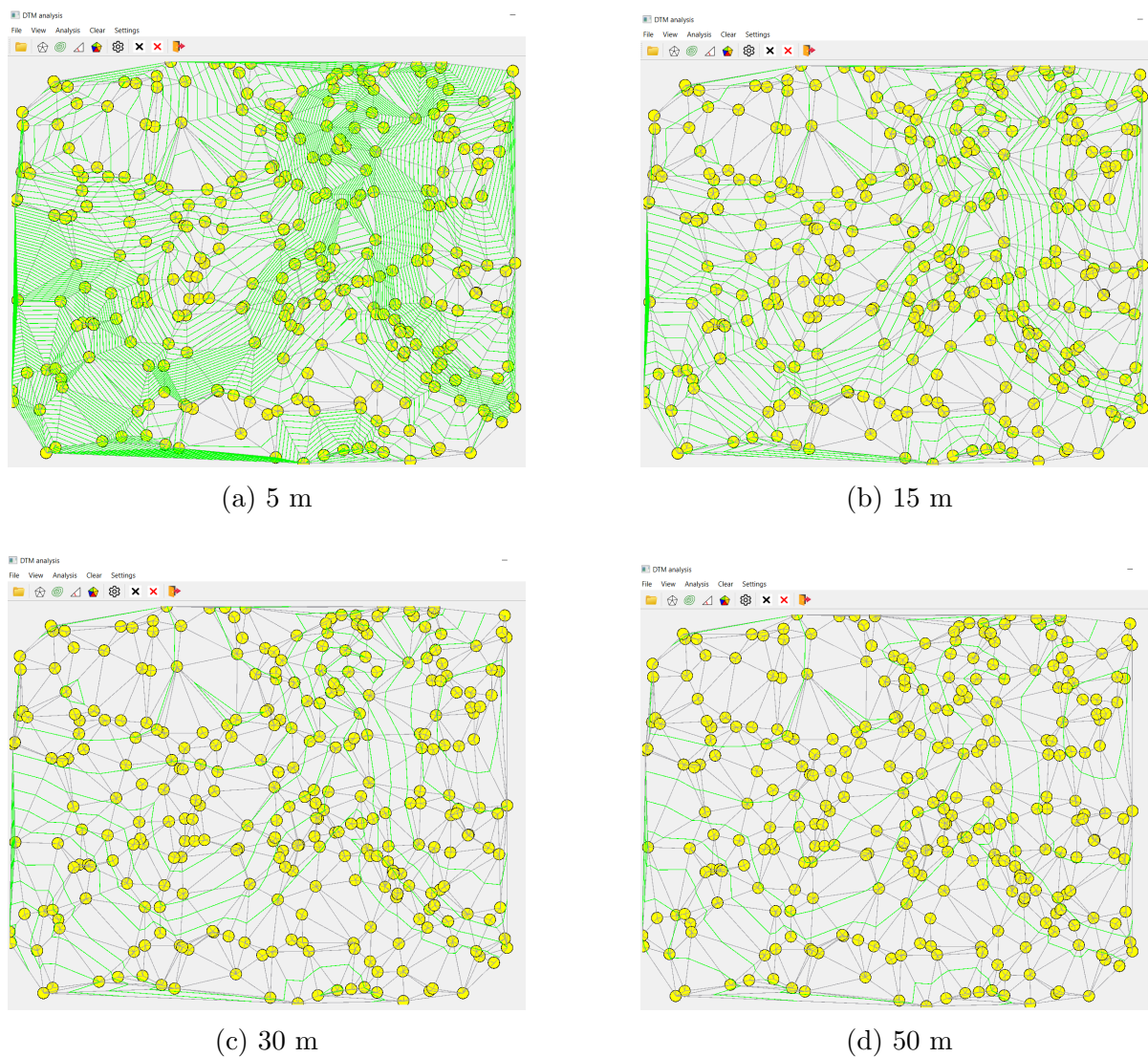
5 Závěr

Jak bylo zmíněno v kapitole Vsutpní Data4.1 data obsahují 300 bodů, ze ZABAGED DMR5G (ČÚZK 2025b), body s podkladovou mapou v okolí Železného Brodu je možno vidět na obrázku 2. Vzhledem k úpravám, které byly provedeny pro účely této úlohy, random subsampling, však není možné algoritmy přímo porovnat s reálnými, úplnými daty.



Obrázek 2: Vizualizace bodů. vlastní zpracování (ČÚZK 2025b), (ČÚZK 2025a).

Na obrázku 3 lze vidět vykreslené vrstevnice. Nastavení parametrů vrstevnic je zobrazeno na obrázku 4



Obrázek 3: Vrstevnice v různých intervalech (ČÚZK 2025b).

Settings

Contour lines parameters

Contour line minimum height 150 m

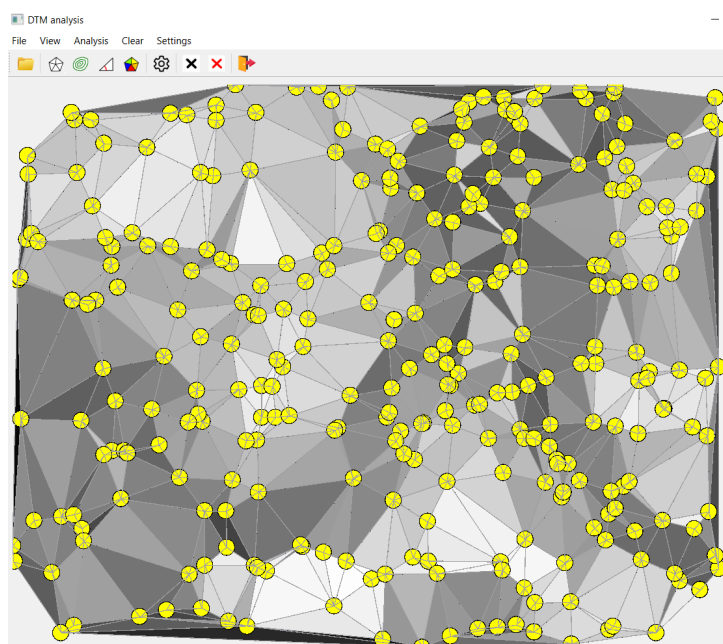
Contour line maximum height 2000 m

Contour line height step 30 m

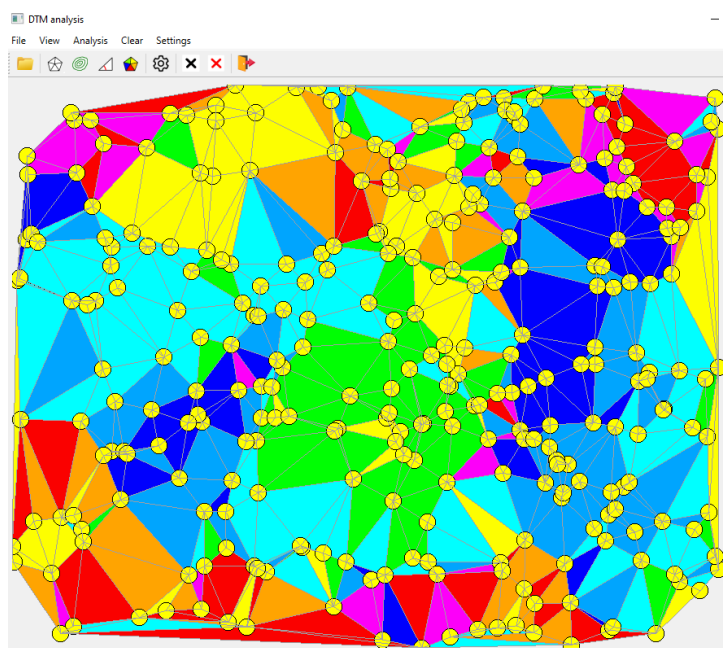
OK Cancel

Obrázek 4: Nastavení parametrů vrstevnic.

Sklon a Orientaci terénu lze vidět na obrázku 5 a 6.

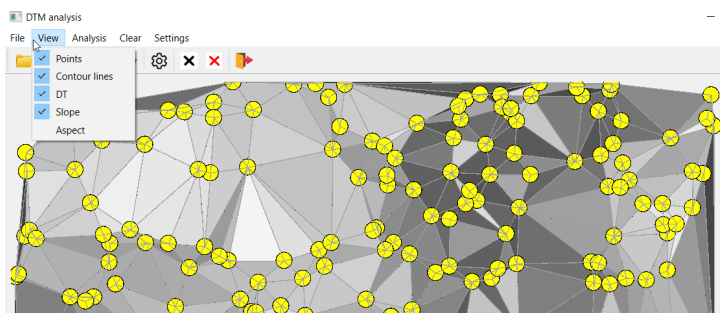


Obrázek 5: Sklon (ČÚZK 2025b).



Obrázek 6: Orientace (ČÚZK 2025b).

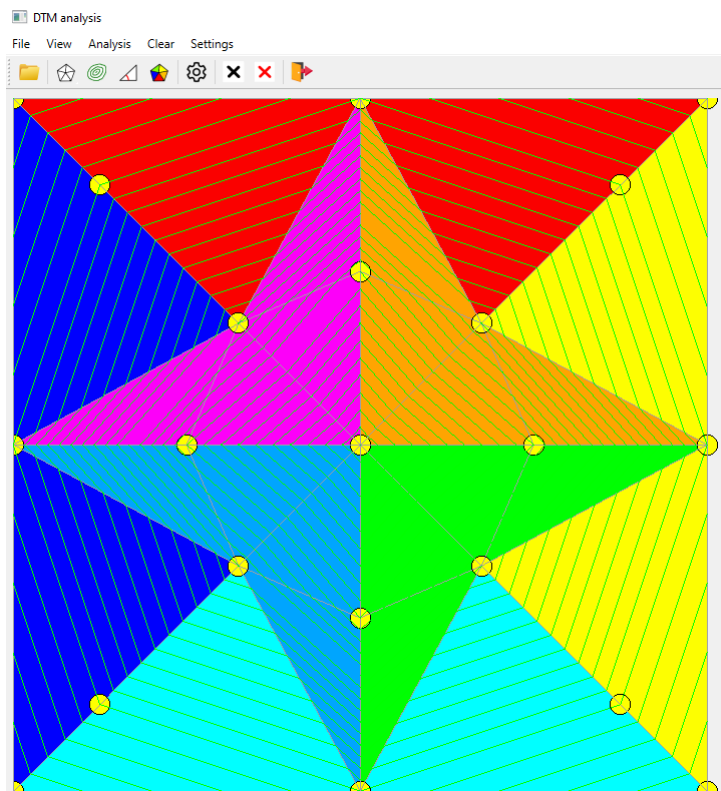
Uživatelské rozhraní také umožňuje vypínání a zapínání jednotlivých vrstev (obrázek 7)



Obrázek 7: Zobrazení vrstev.

I když by pro porovnání byla vhodnější podrobnější bodová síť (pro účely úlohy nepraktické), z hustoty vrstevnic a sklonu terénu je zjevná říční síť s okolním vyvýšeným terénem, ve kterém je rychlý nárůst výšky.

Na obrázku 8 lze vidět test orientace terénu, implementaci orientace terénu lze tedy považovat také za úspěšnou. Test byl prováděn nad uměle vytvořenými daty, kde středový bod má největší výšku.



Obrázek 8: Test orientace.

6 Zdroje

Odkazy

- Bayer, Tomáš (2025). *Rovinné triangulace a jejich využití*. URL: https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk5_new.pdf (cit. 18.05.2025).
- ČÚZK (2025a). *WMS - ZTM100*. URL: <https://ags.cuzk.gov.cz/arcgis1/services/ZTM/ZTM100/MapServer/WSServer?> (cit. 04.05.2025).
- (2025b). *ZABAGED® - Výškopis - DMR 5G*. URL: [https://geoportal.cuzk.cz/\(S\(vsx5paftg5mgsutkke3qmqbg\)\)/Default.aspx?mode=TextMeta&side=vyskopis&metadataID=CZ-CUZK-DMR5G-V&head_tab=sekce-02-gp&menu=302](https://geoportal.cuzk.cz/(S(vsx5paftg5mgsutkke3qmqbg))/Default.aspx?mode=TextMeta&side=vyskopis&metadataID=CZ-CUZK-DMR5G-V&head_tab=sekce-02-gp&menu=302) (cit. 18.05.2025).