



# ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

## Úloha č. 3:

## Digitální model terénu a jeho analýzy

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

28. listopadu 2022

# Obsah

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>Zadání</b>                                  | <b>2</b>  |
| <b>2</b>  | <b>Bonusové úlohy</b>                          | <b>3</b>  |
| <b>3</b>  | <b>Popis a rozbor problému</b>                 | <b>3</b>  |
| 3.1       | Delaunayho triangulace . . . . .               | 3         |
| 3.2       | Lineární interpolace vrstevnic . . . . .       | 3         |
| 3.3       | Expozice . . . . .                             | 4         |
| 3.4       | Sklon . . . . .                                | 4         |
| <b>4</b>  | <b>Popis použitých algoritmů</b>               | <b>4</b>  |
| 4.1       | Delaunayova triangulace . . . . .              | 4         |
| 4.2       | Generování vrstevnic a jejich popisů . . . . . | 5         |
| <b>5</b>  | <b>Vzhled aplikace</b>                         | <b>6</b>  |
| <b>6</b>  | <b>Vstupní data</b>                            | <b>6</b>  |
| 6.1       | Vložení bodů pomocí kliknutí myši . . . . .    | 7         |
| 6.2       | Načtení bodů ze csv souboru . . . . .          | 7         |
| <b>7</b>  | <b>Výstupní data</b>                           | <b>8</b>  |
| <b>8</b>  | <b>Problematické situace a jejich rozbor</b>   | <b>8</b>  |
| <b>9</b>  | <b>Dokumentace</b>                             | <b>9</b>  |
| 9.1       | Třída Algorithms . . . . .                     | 9         |
| 9.2       | Třída Draw . . . . .                           | 10        |
| 9.3       | Třída CSV . . . . .                            | 11        |
| 9.4       | Třída Mainform . . . . .                       | 12        |
| <b>10</b> | <b>Závěr</b>                                   | <b>12</b> |

# 1 Zadání

## Úloha č. 3: Digitální model terénu

*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. navrhnete 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.

Zhodnot'te výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabinami 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.

### Hodnocení:

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

Obrázek 1.1: Zadání úlohy

## 2 Bonusové úlohy

1. *Triangulace nekonvexní oblasti zadané polygonem.* +5b
2. *Výběr barevných stupnic při vizualizaci sklonu a expozice.* +3b
3. *Automatický popis vrstevnic.* +3b
4. *Automatický popis vrstevnic respektující kartografické zásady (orientace, shodné rozložení)* +10b
5. *Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet,...)* +10b
6. *3D vizualizace terénu s využitím promítání.* +10b
7. *Barevná hypsometrie.* +5b

## 3 Popis a rozbor problému

Na vstupu je množina bodů  $P\{p_i\}_{i=0}^n$ , kde jednomu bodu náleží trojice souřadnic  $[x,y,z]$ . Úkolem je pak vytvořit Delaunayho triangulaci (dále jen 'DT'), která je tvořena trojúhelníky  $t_j$ . Stejně trojúhelníky jsou použity i pro další analýzy terénu.

K výpočtu bylo využito [1]

### 3.1 Delaunayho triangulace

Výsledkem DT je množina trojúhelníků, které se svým tvarem přibližují rovnostranným trojúhelníkům. Triangulace má čtyři základní vlastnosti: v kružnici opsané trojúhelníku neleží žádný další bod, je maximalizován minimální úhel (nedochází k minimalizaci max. úhlu v trojúhelníku), je lokálně i globálně optimální vůči kritériu minimálního úhlu, je jednoznačná, pokud nebudou čtyři body na jedné kružnici.

### 3.2 Lineární interpolace vrstevnic

Trojúhelník je tvořen třemi hranami ( $e_1, e_2, e_3$ ). Je řešen případ vodorovné roviny a jejího průniku s hranami trojúhelníku. Přičemž mohou nastat tři stavy:

1.  $(z - z_i)(z - z_{i+1}) < 0 \rightarrow e_i$  protíná vodorovnou rovinu o urč. výšce
2.  $(z - z_i)(z - z_{i+1}) > 0 \rightarrow e_i$  neprotíná vodorovnou rovinu o urč. výšce
3.  $(z - z_i)(z - z_{i+1}) = 0 \rightarrow e_i$  leží ve vodorovné rovině o urč. výšce

Pro případ 1) jsou pak spočteny polohové souřadnice X a Y. Souřadnice Z je známa jako výška vodorovné roviny.

$$x = \frac{(x_2 - x_1)(z - z_1)}{z_2 - z_1} + x_1 \quad (3.1)$$

$$y = \frac{(y_2 - y_1)(z - z_1)}{z_2 - z_1} + y_1 \quad (3.2)$$

### 3.3 Expozice

Expozicí se rozumí orientace trojúhelníku, která je dána azimutem průmětu gradientu  $\nabla\rho$  do roviny x,y. Azimut je určen podle vzorce  $A = \arctan(n/m)$ , kde m a n jsou vektorové součiny pro vektory  $\vec{u}$  a  $\vec{v}$ .

Hodnota azimutu byla převedena na interval od 0 do  $2\pi$  a transformována na interval barvy od 0 do 255. Převod na interval barvy byl proveden kvůli plynulému navázání azimutů blízkých nule a azimutů blízkých  $360^\circ$ . Bylo tak získáno osvětlení terénu od severního směru.

Barva slouží k vizualizaci expozice.

### 3.4 Sklon

Sklonem je chápán úhel mezi svislicí a normálou trojúhelníku. Sklon je určen vztahem:

$$\phi = \arccos\left(\frac{\vec{n} * \vec{n}_t}{|\vec{n}| |\vec{n}_t|}\right) \quad (3.3)$$

kde  $\vec{n} = (0, 0, 1)$  a  $\vec{n}_t = \vec{u} \times \vec{v}$ .

Ze všech trojúhelníků byly vybrány minimální a maximální hodnoty sklonů a následujícím vzorcem byla určena korespondující barva:

$$barva = (slope - minSlope) * \frac{255}{maxSlope - minSlope} \quad (3.4)$$

Hodnota barvy je pak využita k vizualizaci sklonu pomocí černobílé škály (odstíny šedi), kde všechny tři parametry byly rovny hodnotě barva: RGB(barva, barva, barva).

## 4 Popis použitých algoritmů

### 4.1 Delaunayova triangulace

K tvorbě DT byla použita inkrementální konstrukce, kde vybraný Delaunayho bod musí ležet v levé polorovině úsečky s orientací. Poloměr kružnice opsané trojúhelníku je minimální se středem v pravé polorovině hran. Nelze-li takový bod nalézt, je otočena orientace této úsečky.

Algoritmus:

1.  $p_1 = \text{rand}(P)$  // nahodný bod
2.  $p_2 = \arg \min_{p_i \in P} \|p_1 - p_i\|_2$  // nejbližší bod
3. vytvoř hrany  $e = (p_1, p_2)$ ,  $e' = (p_2, p_1)$
4.  $AEL \leftarrow e$ ,  $AEL \leftarrow e'$  // přidání 2 hran do AEL
5. while AEL not empty:
6.      $e_1 = AEL.\text{pop}()$ ,  $e_1 = (p_1, p_2)$  // vezmi první hranu z AEL
7.      $e'_1 = (p_2, p_1)$  // prohoď její orientaci
8.      $\bar{p} = \arg \max_{p_i \in \sigma_L(e'_1)} \angle p_1, p_i, p_2$  // najdi Delanayovský bod
9.     if  $\exists \bar{p}$  : // takový bod existuje
10.          $e_2 = (p_2, \bar{p})$ ,  $e_3 = (\bar{p}, p_1)$  // vytvoř zbývající hrany trojúhelníku
11.          $DT \leftarrow e'_1$ ,  $DT \leftarrow e_2$ ,  $DT \leftarrow e_3$  // přidat hrany do DT
12.         update AEL( $e_2, AEL$ ), add( $e_3, AEL$ ) // Update AEL

Při přidání  $e = (a, b)$  do AEL. Kontrola, zda neobsahuje hranu s opačnou orientací  $e' = (b, a)$ , pokud ano, tak je taková  $e'$  odstraněna z AEL. Pokud tam ještě není, je  $e$  přidána do AEL. Triangulace je ukládána po trojúhelnících.

1. vytvoř hranu  $e' = (b, a)$
2. if ( $e' \in AEL$ )  $AEL \rightarrow e'$ , odstran z AEL
3. else
5.  $AEL \leftarrow e$ , přidej do AE
6.  $DT \leftarrow (a, b)$ , přidej do DT

Implementování bylo provedeno tak, že v prvním kroce bylo vygenerováno  $n$  bodů o náhodných souřadnicích polohy  $x$ ,  $y$ , aby se vešly do vykreslovacího okna Canvas. Dále byly spočteny souřadnice těžiště. Byl počítán cyklus FOR pro  $n$  bodů, kde souřadnice  $Z$  byly vypočteny pomocí výše uvedených rovnic pro kupu, hřbet a sedlo.

## 4.2 Generování vrstevnic a jejich popisů

Po vytvoření DT nad daty, je možné v dalším zpracování - tedy generování vrstevnic. Vrstevnice (izohypsy) jsou linie spojující místa se stejnou předem určenou nadmořskou výškou.

### 4.2.1 Generování vrstevnic

Předpokladem je již mít trojúhelníkovou síť. Hlavní vrstevnice jsou zobrazeny výraznější linií, než jsou vedlejší vrstevnice.

Algoritmus pro získání vrstevnic: Vstupem je DT.

1. Pro  $t_j \in DT$ , kde  $t_j = (e_{j,i}, e_{(j,i)+1}, e_{(j,i)+2})$
2.  $dz_{i1} = z_{j1} - z$ ,
3.  $dz_{i2} = z_{j2} - z$ ,
4.  $dz_{i3} = z_{j3} - z$ , vysokové rozdíly
5.  $dz_{i12} = z_{j1} * z_{j2}$ ,
6.  $dz_{i23} = z_{j2} * z_{j3}$ ,
7.  $dz_{i31} = z_{j3} * z_{j1}$ , součiny vysokových rozdílů
8. if ( $dz_{i1} = 0$ ) & (& ( $dz_{i2} = 0$ ) & (& ( $dz_{i3} = 0$ )), komplanarita rovin
9. Continue

```

10. else if ( $dz_{j1} == 0$ ) && ( $dz_{j2} == 0$ )
11.    $Contours \leftarrow e_{ji}$ , pridej do vrstevnic
12. else if ( $dz_{j2} == 0$ ) && ( $dz_{j3} == 0$ )
13.    $Contours \leftarrow e_{(ji)+1}$ , pridej do vrstevnic
14. else if ( $dz_{j3} == 0$ ) && ( $dz_{j1} == 0$ )
15.    $Contours \leftarrow e_{(ji)+2}$ , pridej do vrstevnic
16. else if ( $((dz_{j12} <= 0)) \&\& (dz_{j23} < 0) \parallel ((dz_{j12} < 0) \&\& (dz_{j23} <= 0))$ )
17.    $A = getContourPoint(p_{j1}, p_{j2}, z)$ 
18.    $B = getContourPoint(p_{j2}, p_{j3}, z)$ 
19.    $Contours \leftarrow e(A, B)$ , pridej do vrstevnic
20. else if ( $((dz_{j23} <= 0)) \&\& (dz_{j31} < 0) \parallel ((dz_{j23} < 0) \&\& (dz_{j31} <= 0))$ )
21.    $A = getContourPoint(p_{j2}, p_{j3}, z)$ 
22.    $B = getContourPoint(p_{j3}, p_{j1}, z)$ 
23.    $Contours \leftarrow e(A, B)$ , pridej do vrstevnic
24. else if ( $((dz_{j31} <= 0)) \&\& (dz_{j12} < 0) \parallel ((dz_{j31} < 0) \&\& (dz_{j12} <= 0))$ )
25.    $A = getContourPoint(p_{j3}, p_{j1}, z)$ 
26.    $B = getContourPoint(p_{j1}, p_{j2}, z)$ 
27.    $Contours \leftarrow e(A, B)$ , pridej do vrstevnic

```

Algoritmus pro získání bodů vrstevnic: Vstupem dva body se souřadnicemi x a y.

$$1. \ x = \frac{(x_2 - x_1)(z - z_1)}{(z_2 - z_1)} + x_1$$

$$2. \ y = \frac{(y_2 - y_1)(z - z_1)}{(z_2 - z_1)} + y_1$$

Algoritmus pro získání hlavních vrstevnic: Vstupem jsou vrstevnice, kde každá k-tá vrstevnice je hlavní s výškovým intervalem dz.

$$1. \ dh = dz * k$$

$$2. \ \text{for } e \in Contours \quad z = getZ(e), \text{ ziskejvyskuhrany}$$

3. if  $z \% dh == 0$ , zbytek po deleni

4.  $ContoursMain \leftarrow pair < z, e >$ , pridej hranu s vyskou jako par do struktury

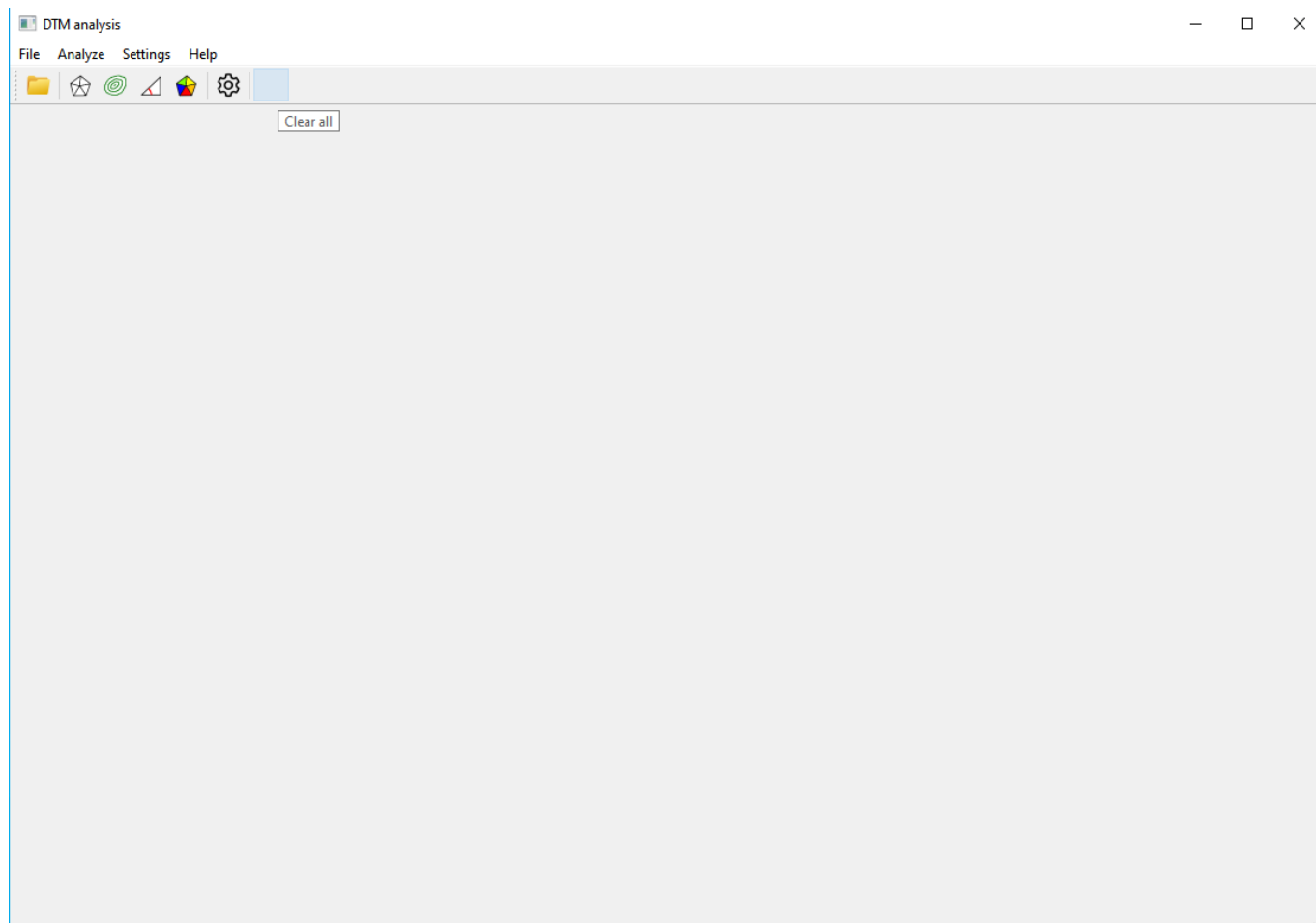
## 5 Vzhled aplikace

Po spuštění aplikace se zobrazí následující okno, zde je možnost si vybrat z menu. Menu obsahuje načtení csv souboru, konec aplikace, výpočet DT, vytvoření vrstevnic, analýza sklonu nebo orientace. Dále je zde nastavení vrstevnic (interval, minimální a maximální vrstevnici) a tlačítko na vymazání celého obsahu.

## 6 Vstupní data

Vstupní data je možno vložit 2 způsoby.

1. Naklikat body v programu
2. Pomocí tlačítka Open otevřít csv soubor.



Obrázek 5.1: Vzhled aplikace

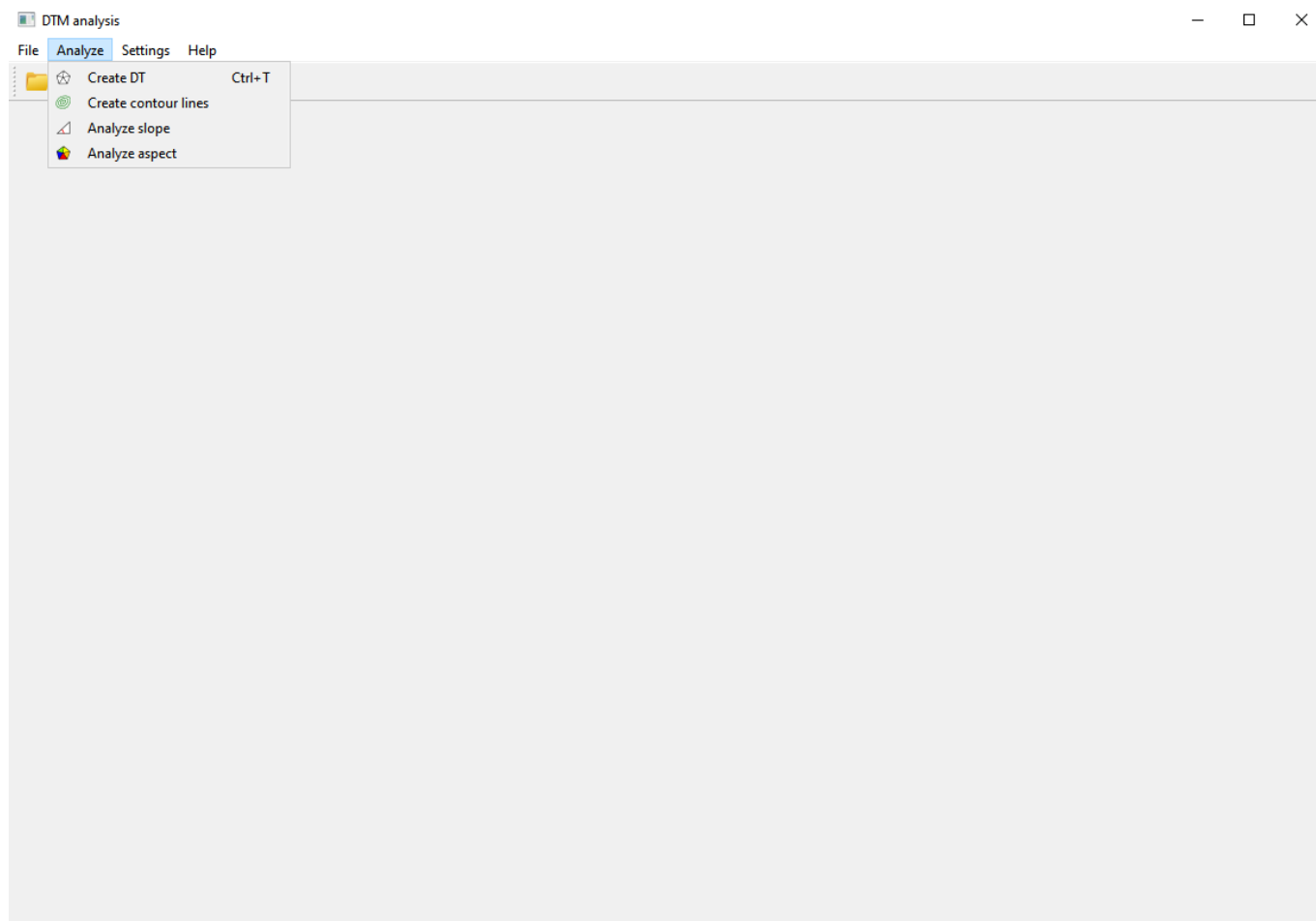
## 6.1 Vložení bodů pomocí kliknutí myši

Prvním způsobem, jak vygenerovat body do aplikace, je pomocí kliknutí, kdy je vygenerována souřadnice s náhodnou Z souřadnicí v rozmezí  $[0,1000]$ .

## 6.2 Načtení bodů ze csv souboru

Načtení bodů pomocí csv souboru je možno po kliknutí na tlačítko open. Otevře se vyhledávací okno a je možno si načíst libovolný csv soubor s mračenem bodu. Soubor csv vypadá následovně podle obrázku





Obrázek 5.2: Menu aplikace

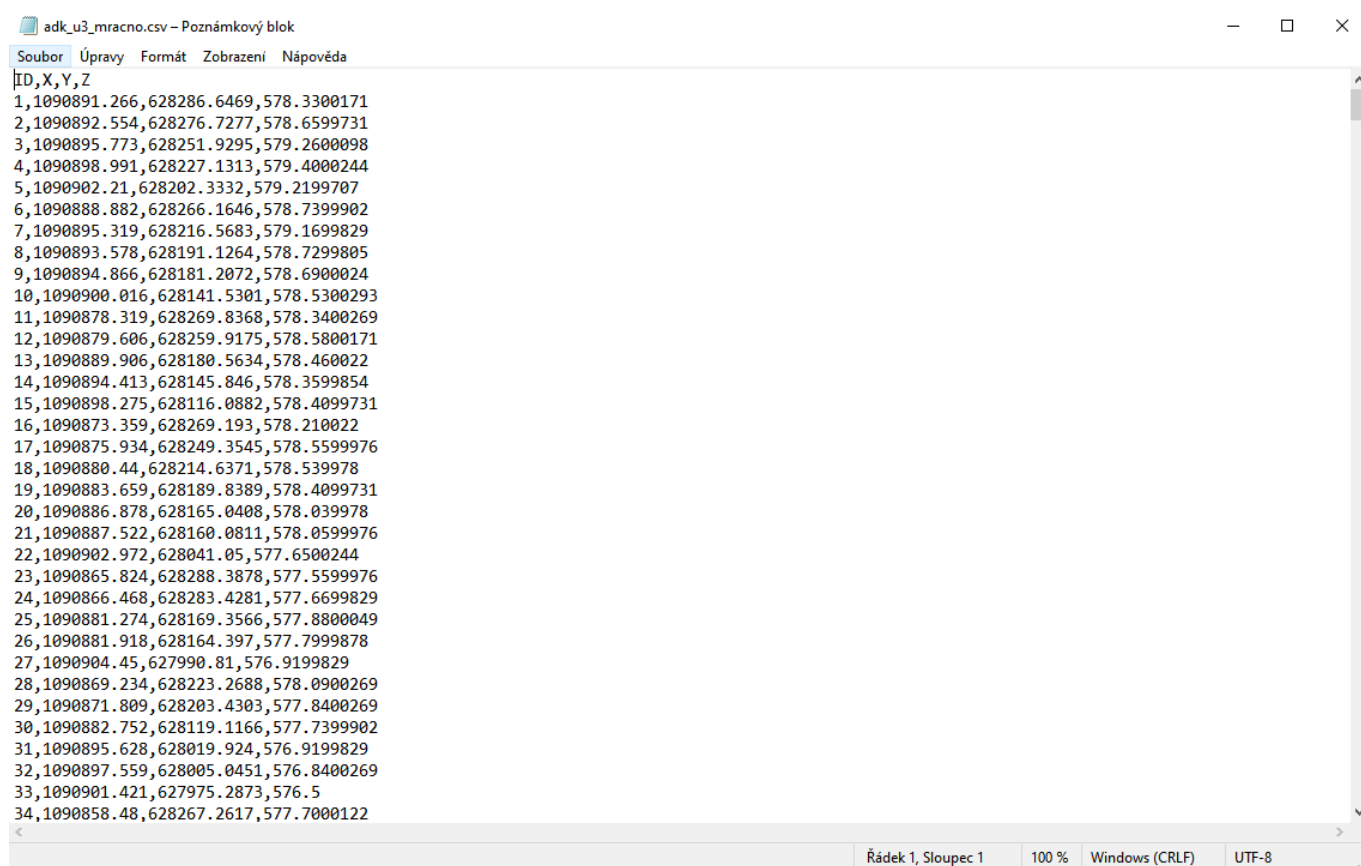
7.2. Kde jsou body vloženy v pořadí ID, X, Y, Z. Důležité je, aby soubor měl hlavičku a oddělovač byl pomocí čárky.

## 7 Výstupní data

Po načtení dat nebo vytvoření dat, je možno vytvořit DT a následně i vytvořit vrstevnice podle zadaného intervalu. Je možné vytvořit sklon či expozici z dat. Poté je možné aplikaci ukončit.

## 8 Problematické situace a jejich rozbor

Delauného triangulace dává dobré výsledky pro terény, které jsou hladké a spojitě, takže u skalních převisů, liniových staveb a lidských úprav terénu nemusí aplikace dávat dobré výsledky. Samotná aplikace



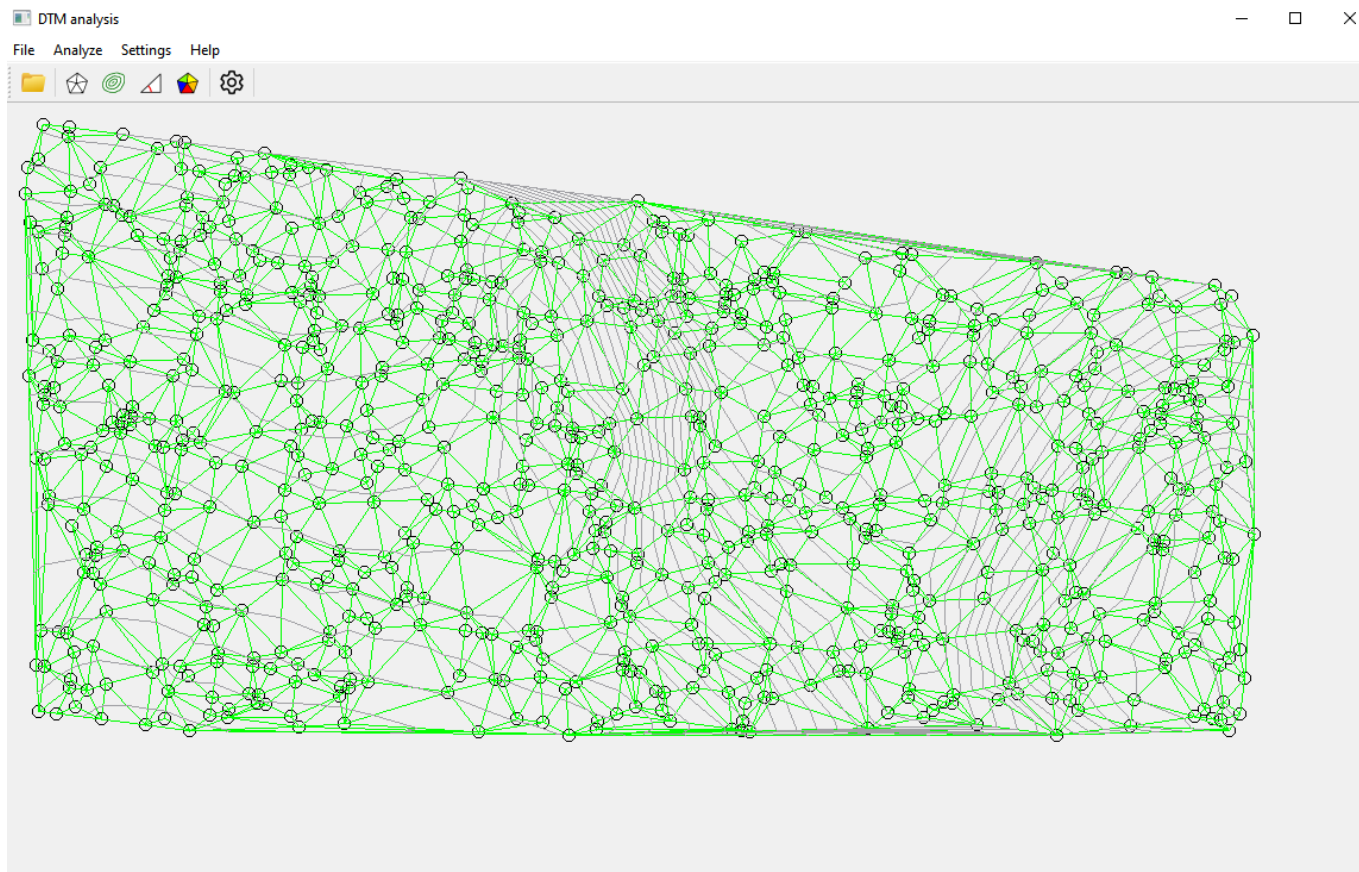
Obrázek 6.1: csv soubor

má v sobě mouchy, není vidět ikona pro vyčištění aplikace. Dále byla vytvořena funkce pro orientaci terénu, která funguje, nicméně se nechce vykreslit.

## 9 Dokumentace

### 9.1 Třída Algorithms

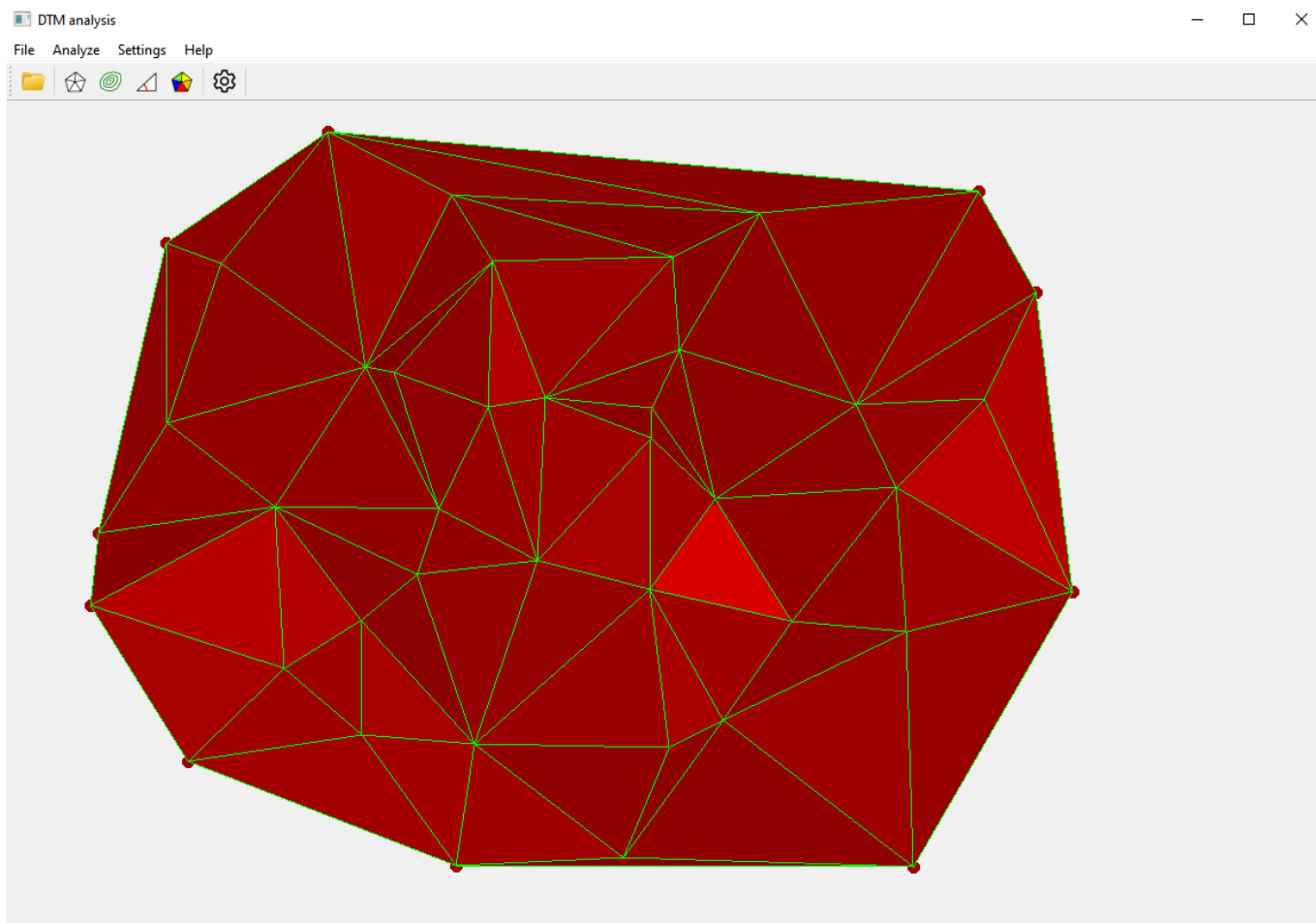
- *int getPointLinePosition(constQPoint3D &p1, constQPoint3D &p2, constQPoint3D &q)*  
– analyzuje vzájemný vztah bodu a přímky
- *double getTwoLinesAngle(constQPoint3D &p1, constQPoint3D &p2, constQPoint3D &p3, constQPoint3D &p4)*  
– vypočítá úhel mezi dvěma vektory
- *int getNearestPoint(constQPoint3D &p, conststd :: vector < QPoint3D > &points)*  
– vrací index nejbližšího bodu k bodu p
- *int getDelaunayPoint(constQPoint3D &p1, constQPoint3D &p2, conststd :: vector < QPoint3D > &points);*



Obrázek 7.1: Načtení csv souboru, následně vytvořené DT společně s vrstevnicemi.

- vrací index Delaunayho bodu v rámci vektoru bodů
- *std :: vector < Edge > createDT(const std :: vector < QPoint3D > points);*
  - vytvoří DT ze vstupních bodů
- *void updateAEL(const Edge &e, std :: list < Edge > &ael);*
  - aktualizuje hranu v listu hran
- *std :: vector < Edge > createContourLines(const std :: vector < Edge > dt, double zmin, double zmax,*  
  - vrací hrany vrstevnic
- *double computeSlope(const QPoint3D &p1, const QPoint3D &p2, const QPoint3D &p3);*
  - pomocný výpočet pro sklon
- *std :: vector < Triangle > analyzeSlope(const std :: vector < Edge > &dt)*
  - vrací hodnotu sklonu trojúhelníku
- *double computeAspect(const QPoint3D &p1, const QPoint3D &p2, const QPoint3D &p3)*
  - pomocný výpočet pro orientaci
- *std :: vector < Triangle > analyzeAspect(const std :: vector < Edge > &dt)*
  - vrátí hodnotu orientace trojúhelníku

## 9.2 Třída Draw



Obrázek 7.2: Vytvoření sklonu bez vrstevnic.

- *void mousePressEvent(QMouseEvent \* event)*  
– vrátí souřadnice kurzoru po kliknutí na Canvas
- *void paintEvent(QPaintEvent \* event)*  
– vykreslí body na Canvas
- *void clearContours()*  
– vyčistí vrstevnice
- *void clearPoints()*  
– vyčistí body
- *void clearDT()*  
– vyčistí DT

## 9.3 Třída CSV

- *static std::vector< std::vector< std::string >> read\_csv(std::string &filename);*  
– načte vstupní csv soubor

- *static std :: vector < QPoint3D > getPoints3D(std :: vector < std :: vector < std :: string >> &csv\_content, double &x\_min, double &x\_max, double &y\_min, double &y\_max);*  
– vrací vektor QPoint3D z csv souboru

## 9.4 Třída MainForm

- *void on\_actionOpen\_triggered()*  
– otevře průzkumníka souborů
- *void on\_actionCreate\_DT\_triggered()*  
– vytvoří DT
- *void on\_actionClear\_all\_triggered()*  
– vyčistí Canvas
- *void on\_actionExit\_triggered()*  
– ukončí soubor
- *void on\_actionCreate\_contour\_lines\_triggered()*  
– vytvoří vrstevnice
- *void on\_actionCreate\_slope\_triggered()*  
– vytvoří sklon svahu
- *void on\_actionCreate\_aspect\_triggered()*  
– vytvoří orientaci svahu

## 10 Závěr

Byla vytvořena aplikace pro zpracování množiny bodů, kde výstupem je vytvoření digitálního modelu terénu a analýzy tohoto modelu.

## Literatura

- [1] Tomáš Bayer, *DMT*, [https://agony.natur.cuni.cz/~bayertom/images/courses/Adk/adk5\\_new.pdf](https://agony.natur.cuni.cz/~bayertom/images/courses/Adk/adk5_new.pdf), [27.11.2022].