



# ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

## Úloha č. 3:

## Digitální model terénu a jeho analýzy

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

14. února 2023

# 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 . . . . .	3
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 . . . . .	8
6.2	Načtení bodů ze csv souboru . . . . .	8
<b>7</b>	<b>Výstupní data</b>	<b>8</b>
<b>8</b>	<b>Problematické situace a jejich rozbor</b>	<b>9</b>
<b>9</b>	<b>Analýza polyedrického modelu</b>	<b>11</b>
9.1	Nekonvexní tvar . . . . .	11
<b>10</b>	<b>Dokumentace</b>	<b>12</b>
10.1	Třída Algorithms . . . . .	12
10.2	Třída Draw . . . . .	14
10.3	Třída CSV . . . . .	14
10.4	Třída Mainform . . . . .	15
<b>11</b>	<b>Závěr</b>	<b>15</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

Nebyly zpracovány žádné bonusové úlohy.

## 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 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. 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 orientované úsečky. Poloměr kružnice opsané trojúhelníku musí být minimální se středem v pravé polorovině. Nelze-li takovýto bod nalézt, je otočena orientace této úsečky.

Algoritmus:

1.  $p_1 = \text{rand}(P)$ ,  $\|p_1 - p_i\|_2 // \min$ , náhodný bod, nejbližší bod
2. vytvoř hranu  $e = (p_1, p_2)$
3.  $\underline{p} = \arg \min_{p_i \in L(e)} r'(k_i)$ ,  $k_i = (a, b, p_i)$ ,  $e(a, b)$
4. Pokud  $\nexists \underline{p}$ , prohodí se orientace  $e \leftarrow (p_2, p_1)$ , jdi na 3)
5.  $e_2 = (p_2, \underline{p})$ ,  $e_3 = (\underline{p}, p_1)$  - zbývající hrany trojúhelníku
6.  $AEL \leftarrow e$ ,  $AEL \leftarrow e_2$ ,  $AEL \leftarrow e_3$ , přidání 3 hran do AEL
7.  $DT \leftarrow e$ ,  $DT \leftarrow e_2$ ,  $DT \leftarrow e_3$ , přidání 3 hran do DT

8. while AEL not empty:
9.      $AEL \rightarrow e, e = (p_1, p_2)$  vezmi první hranu z AEL
10.     $e = (p_2, p_1)$ , prohod' její orientaci
11.     $p = \arg \min_{\forall p_i \in L(e)} r'(k_i), k_i = (a, b, p_i), e(a, b)$
12.    if  $\exists p$ :
13.        $e_2 = (p_2, p), e_3 = (p, p_1)$  - zbývající hrany trojúhelníku
14.       DTlongleftarrow e, přidej hranu do DT, ale ne do AEL
15.       add( $e_2, AEL, DT$ ), add( $e_3, AEL, DT$ ) přidej DT i do AEL

### 4.1.1 Implementace *add*

1. vytvoř hranu  $e' = (b, a)$
  2. if ( $e' \in AEL$ )
  3.      $AEL \rightarrow e'$ , odstraň z AEL
  4. else
  5.      $AEL \leftarrow e$ , přidej do AE
  6.      $DT \leftarrow (a, b)$ , přidej do DT
- DT jak v aplikaci vypadají v aplikaci ukazuje obrázek 7.1

## 4.2 Generování vrstevnic a jejich popisů

Po vytvoření Delaunayovy triangulace nad daty je možné v dalším zpracování generovat vrstevnice. Vrstevnice (izohypsy) jsou linie spojující místa se stejnou předem určenou nadmořskou výškou.

### 4.2.1 Generování vrstevnic

Předpokladem je již mít vytvořenou trojúhelníkovou síť. Všechny vrstevnice jsou generovány funkcí *Create contour lines*.

Algoritmus pro získání vrstevnic:

Vstupem je  $DT = \{e_1, e_2, \dots\}$ , kde  $e_i = \{s_i, e_i\}$ ,  $s_i = \{x_{is}, y_{is}, z_{is}\}$ ,  $e_i = \{x_{ie}, y_{ie}, z_{ie}\}$  a  $z$  je výška vrstevnice

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.  $dz_{i3} = z_{j3} - z$ , vyskove rozdily
5.  $dz_{i12} = z_{j1} * z_{j2}$ ,
6.  $dz_{i23} = z_{j2} * z_{j3}$ ,
7.  $dz_{i31} = z_{j3} * z_{j1}$ , souciny vyskovych rozdilu
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. for  $e \in Contours$   $z = getZ(e)$ , ziskejvyskuhrany

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

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

Vrstevnice jak v aplikaci vypadají v aplikaci ukazuje obrázek 7.2

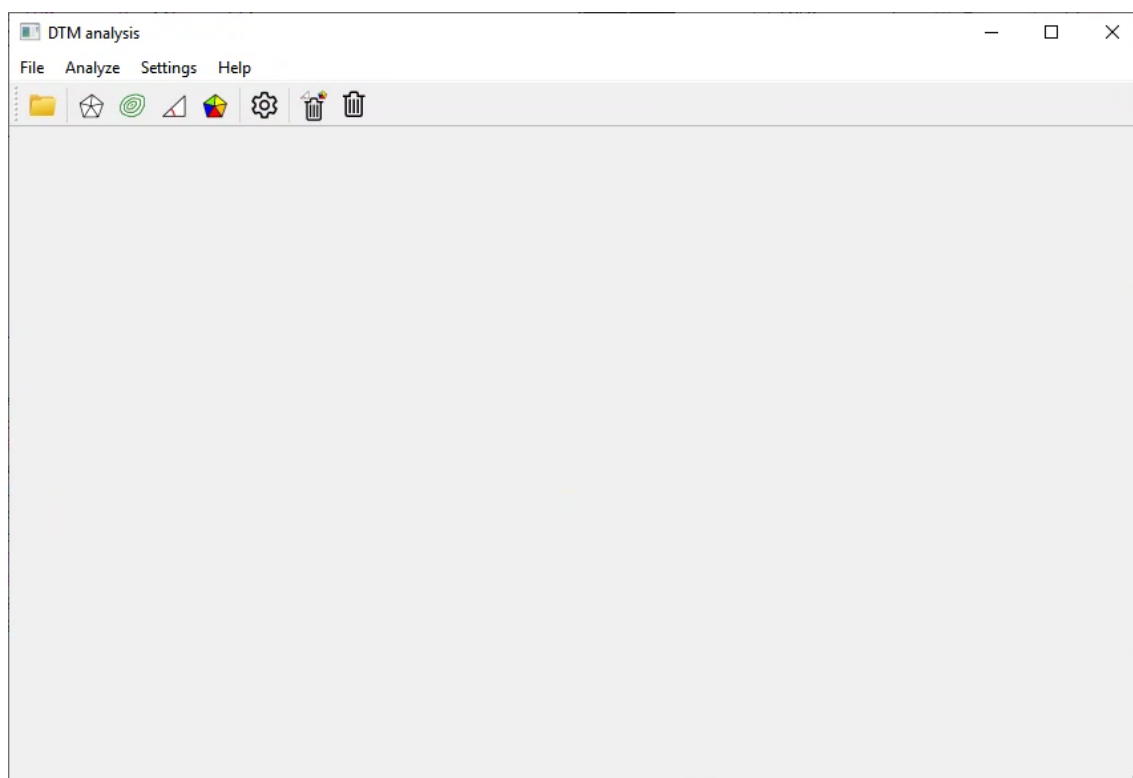
## 5 Vzhled aplikace

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, analýzu sklonu nebo expozice. Dále je zde nastavení vrstevnic (interval, minimální a maximální vrstevnice), tlačítko na vymazání analýzy sklonu nebo expozice a tlačítko na vymazání celého obsahu.

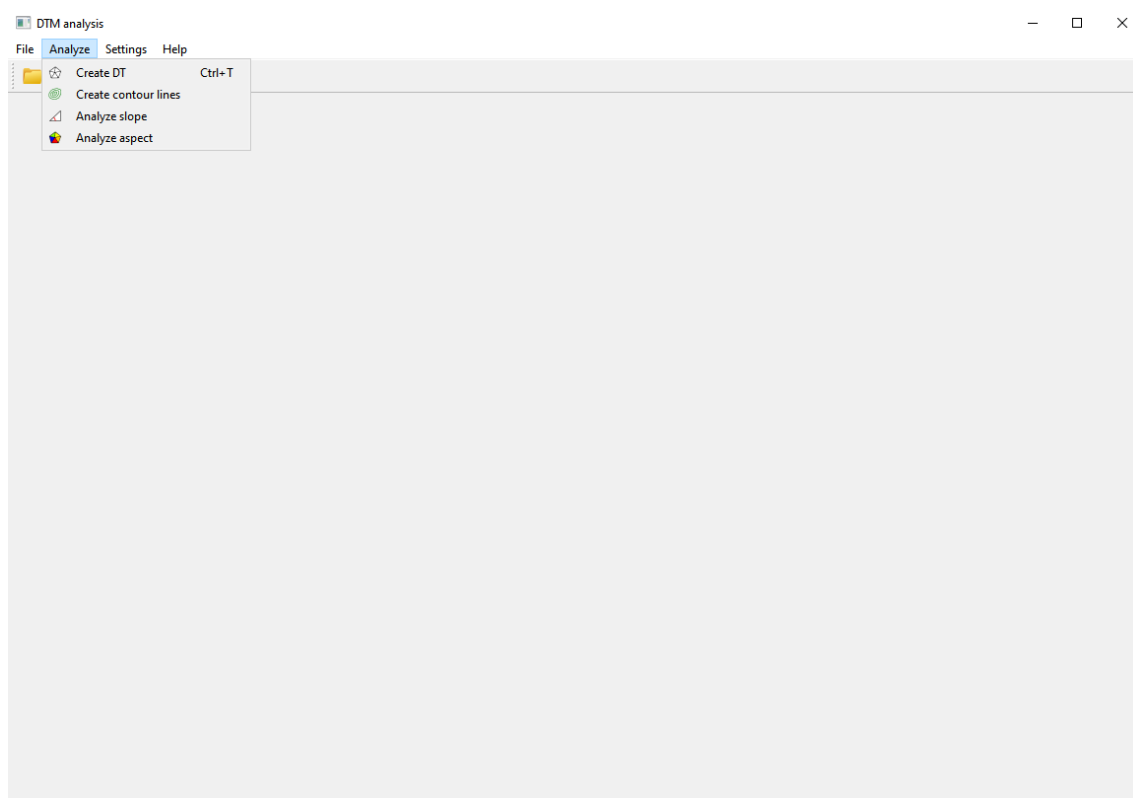
## 6 Vstupní data

Vstupní data je možno vložit 2 způsoby.

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



Obrázek 5.1: Vzhled aplikace



Obrázek 5.2: Menu aplikace

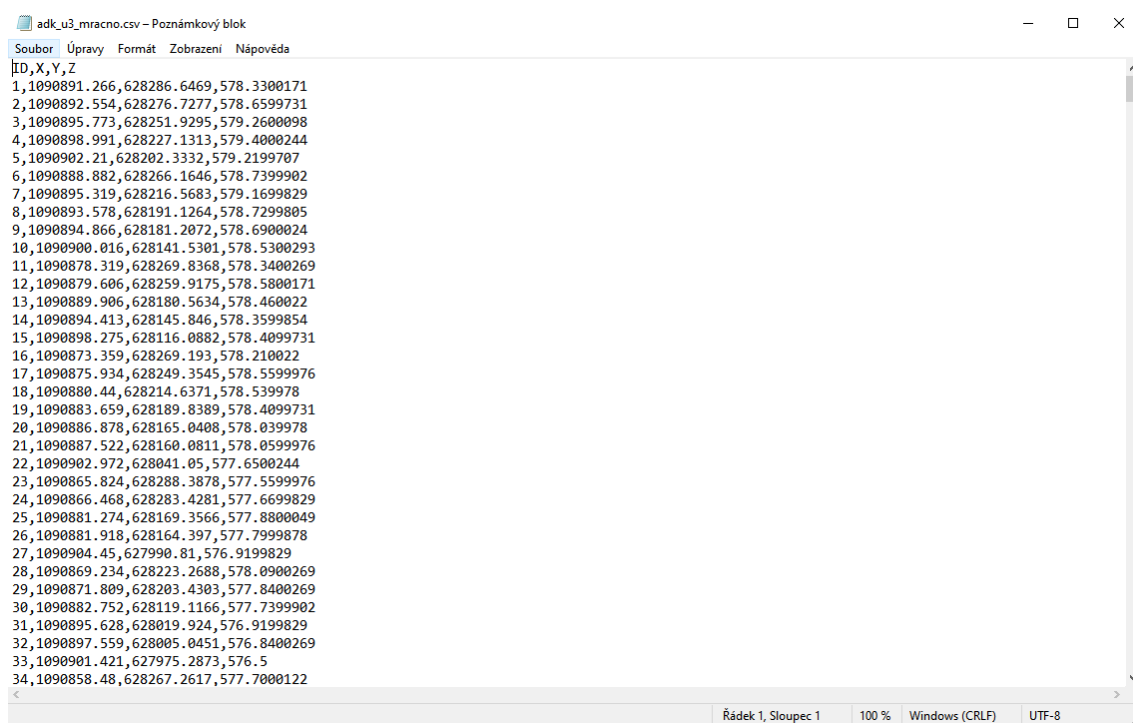


## 6.1 Vložení bodů pomocí kliknutí myši

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 mračnem bodů. Soubor csv vypadá následovně podle obrázku 6.1, struktura každého řádku je ID, X, Y, Z bodu. Důležité je, aby soubor měl hlavičku.

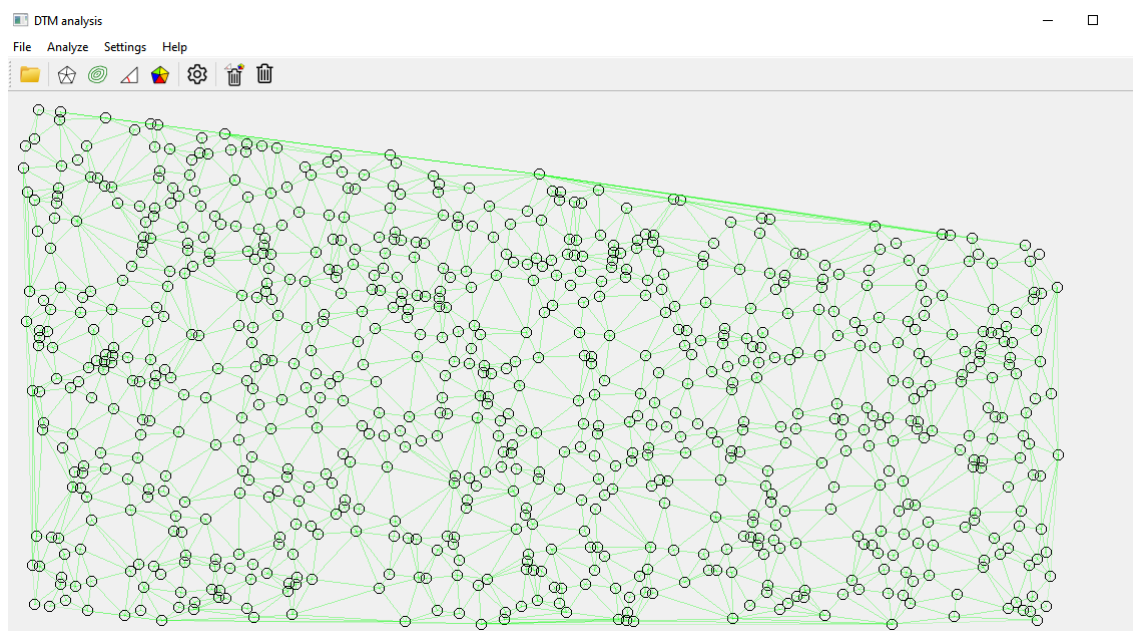


```
adk_u3_mracno.csv - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
ID,X,Y,Z
1,1090891.266,628286.6469,578.3300171
2,1090892.554,628276.7277,578.6599731
3,1090895.773,628251.9295,579.2600098
4,1090898.991,628227.1313,579.4000244
5,1090902.21,628202.3332,579.2199707
6,1090888.882,628266.1646,578.7399902
7,1090895.319,628216.5683,579.1699829
8,1090893.578,628191.1264,578.7299805
9,1090894.866,628181.2072,578.6900024
10,1090900.016,628141.5301,578.5300293
11,1090878.319,628269.8368,578.3400269
12,1090879.606,628259.9175,578.5800171
13,1090889.906,628180.5634,578.460022
14,1090894.413,628145.846,578.3599854
15,1090898.275,628116.0882,578.4099731
16,1090873.359,628269.193,578.210022
17,1090875.934,628249.3545,578.5599976
18,1090880.44,628214.6371,578.539978
19,1090883.659,628189.8389,578.4099731
20,1090886.878,628165.0408,578.039978
21,1090887.522,628160.0811,578.0599976
22,1090902.972,628041.05,577.6500244
23,1090865.824,628288.3878,577.5599976
24,1090866.468,628283.4281,577.6699829
25,1090881.274,628169.3566,577.8800049
26,1090881.918,628164.397,577.7999878
27,1090904.45,627990.81,576.9199829
28,1090869.234,628223.2688,578.0900269
29,1090871.809,628203.4303,577.8400269
30,1090882.752,628119.1166,577.7399902
31,1090895.628,628019.924,576.9199829
32,1090897.559,628005.0451,576.8400269
33,1090901.421,627975.2873,576.5
34,1090858.48,628267.2617,577.7000122
< >
Řádek 1, Sloupec 1 100 % Windows (CRLF) UTF-8
```

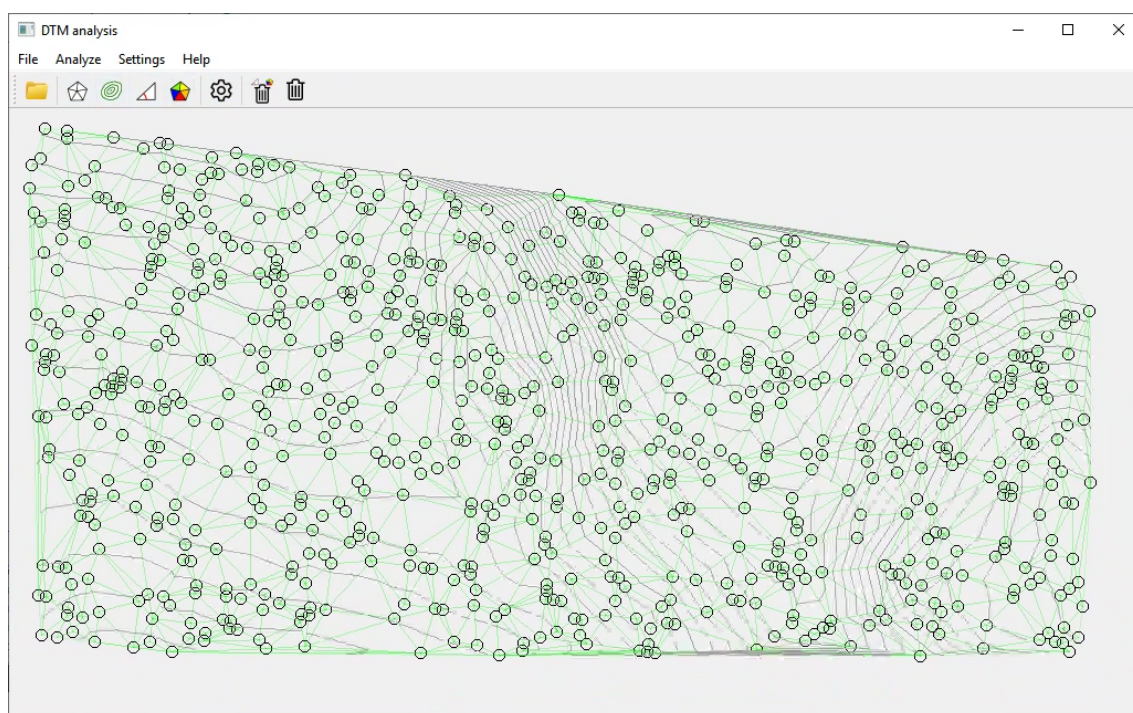
Obrázek 6.1: csv soubor

## 7 Výstupní data

Po načtení dat nebo vytvoření dat, je možno vytvořit DT a následně i vytvořit vrstevnice podle za-



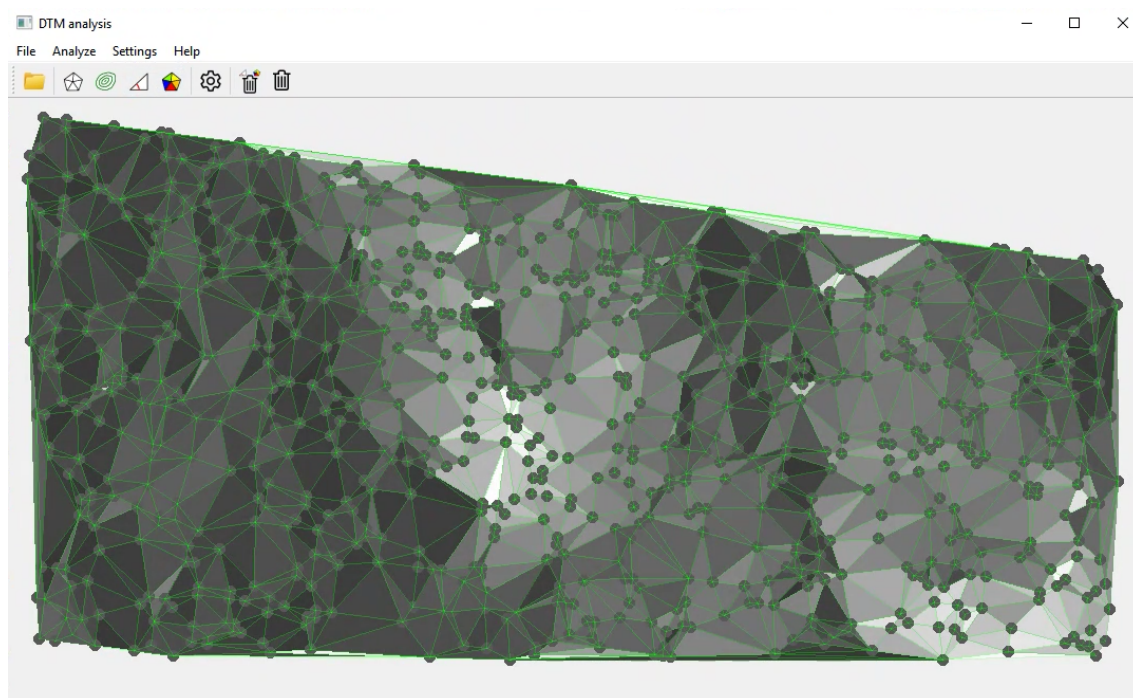
Obrázek 7.1: Načtení csv souřadnic a vytvořené DT.



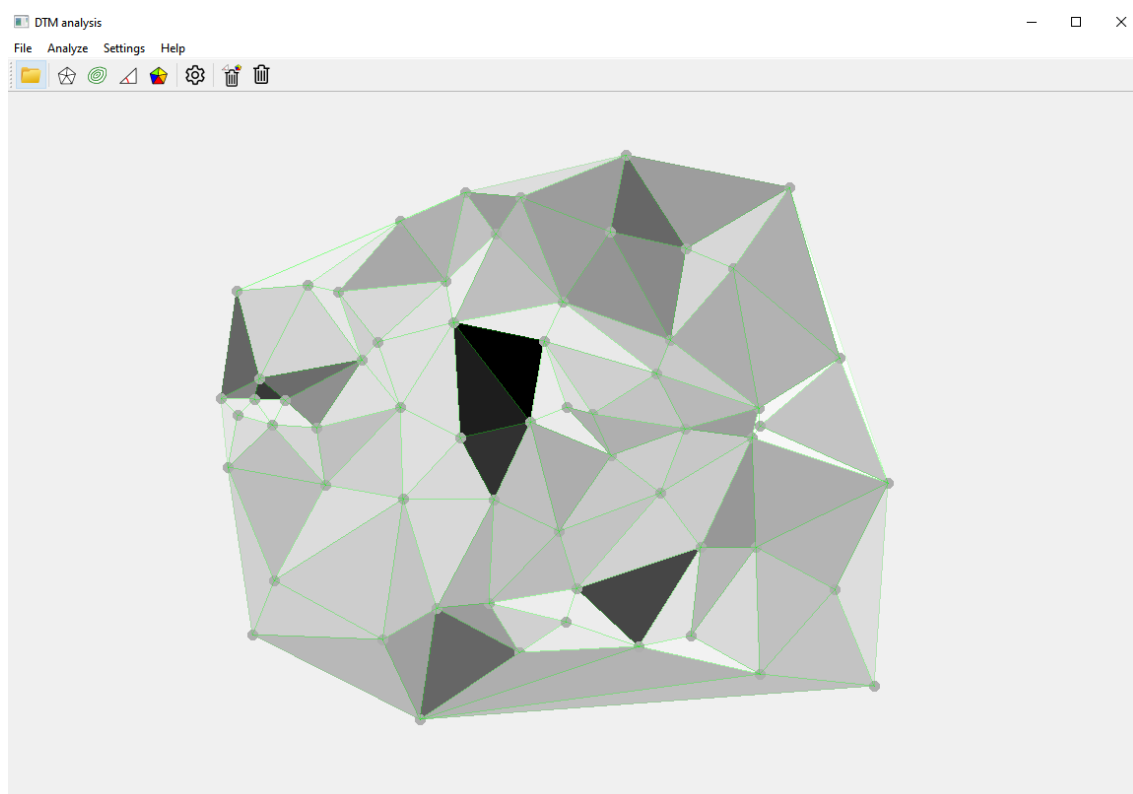
Obrázek 7.2: Načtení csv souboru, následně vytvořené DT společně s vrstevnicemi - rozestup 1 m

daného intervalu. Je možné vytvořit sklon či expozici z dat. Poté je možné aplikaci ukončit. Vytvoření expozice s načtených dat viz obrázek 7.2. Vytvoření sklonu a expozice z naklikaných bodů viz 7.4 7.3.

## 8 Problematické situace a jejich rozbor

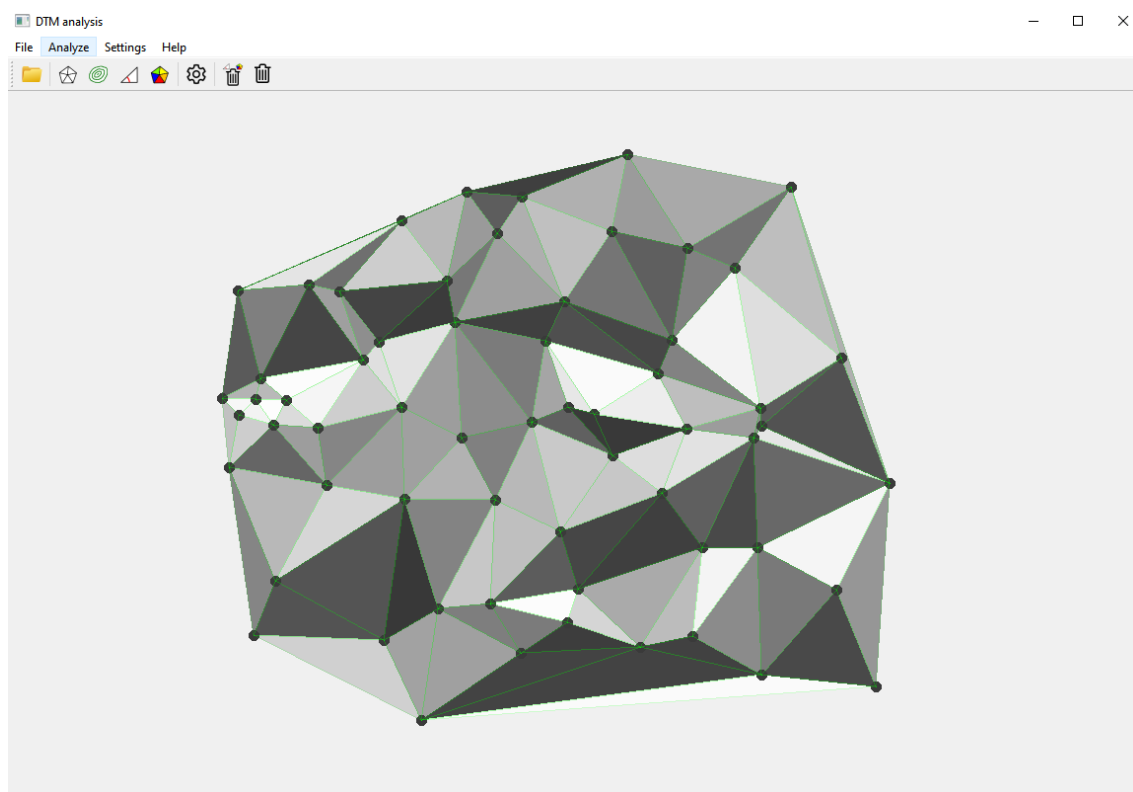


Obrázek 7.3: Vytvoření expozice bez vrstevnic



Obrázek 7.4: Vytvoření sklonu naklikanými body

Delaunayho 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. DT nevhodně



Obrázek 7.5: Vytvoření expozice naklikanými body

vystihuje terénní hrany. Proto bez ručního zadání terénních hran do modelu je terén aproximován hůře. Takže pokud se přidají terénní hrany, tak DT pracuje lépe a model není zdeformovaný.

## 9 Analýza polyedrického modelu

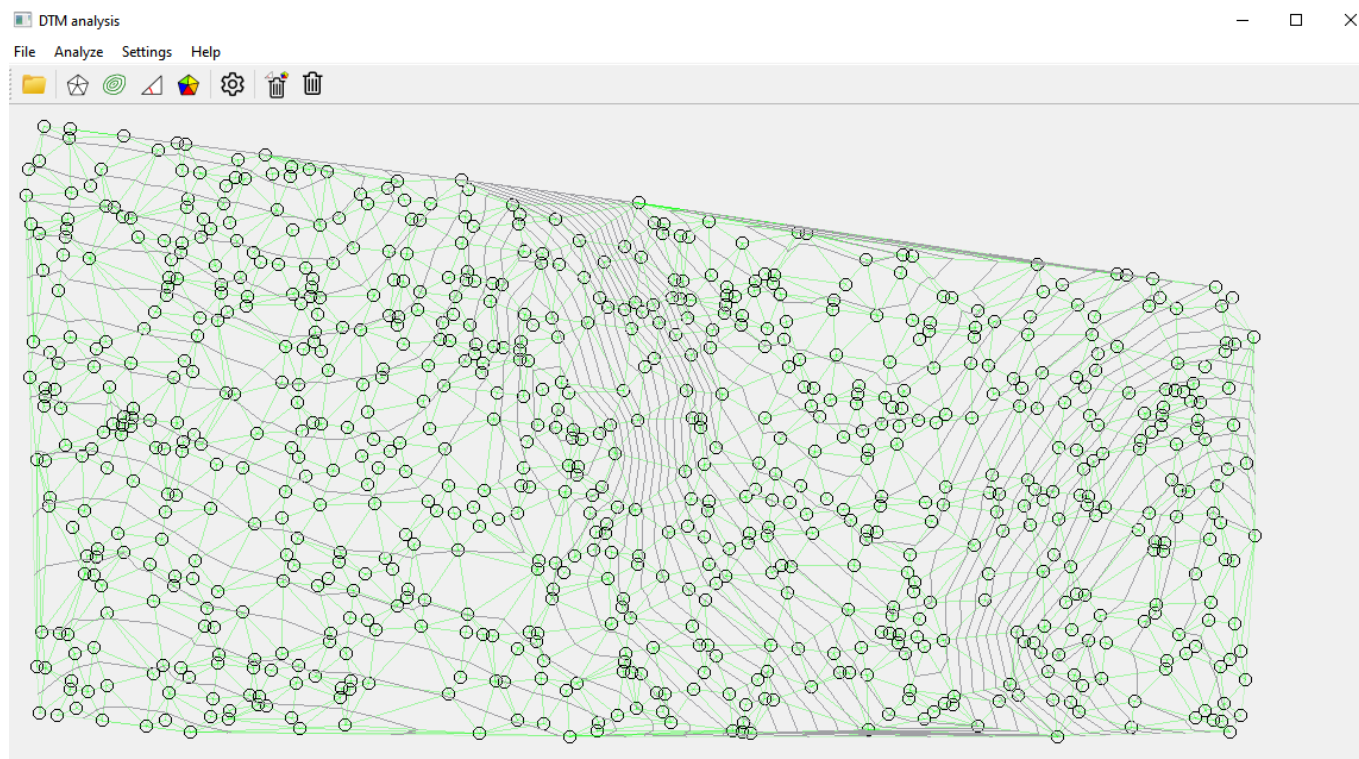
Polyedrickým modelem se rozumí terén, který je reprezentován nepravidelnými trojúhelníky, společná je nejvyšší hrana. Obecně Delaunayho triangulace dává dobré výsledky pro terény, které jsou hladké a spojité. Tvar může být konvexní či nekonvexní, s otvory či bez. Pokud lze zadat povinné spojnice (hřbetnice, údolnice, spádnice), zlepšují se aproximační vlastnosti. Problematické situace mohou nastat u děl vytvořených lidmi, jako jsou liniové stavby a jejich terénní úpravy kolem nich. Dále to mohou být koryta řek, skalní převisy.

Bohužel nemáme více reálných dat, na kterých by se dalo lépe analyzovat algoritmus. Protože naklikané body Canvasu nenahradí skutečný terén.

### 9.1 Nekonvexní tvar



V aplikaci není implementována možnost triangulovat nekonvexní oblasti. Viz obrázek 9.1. Kdy část měření algoritmus dobře aproximuje území, nicméně pokud se jedná o odlehlé body a nebo body na kraji území s dlouhými hranami trojúhelníku viz označeno červeně na obrázku 9.2, tak algoritmus reprezentuje skutečnost špatně. Ke zlepšení by pomohlo, pokud by byla možnost odebrat trojúhelníky, které nechceme zahrnout.

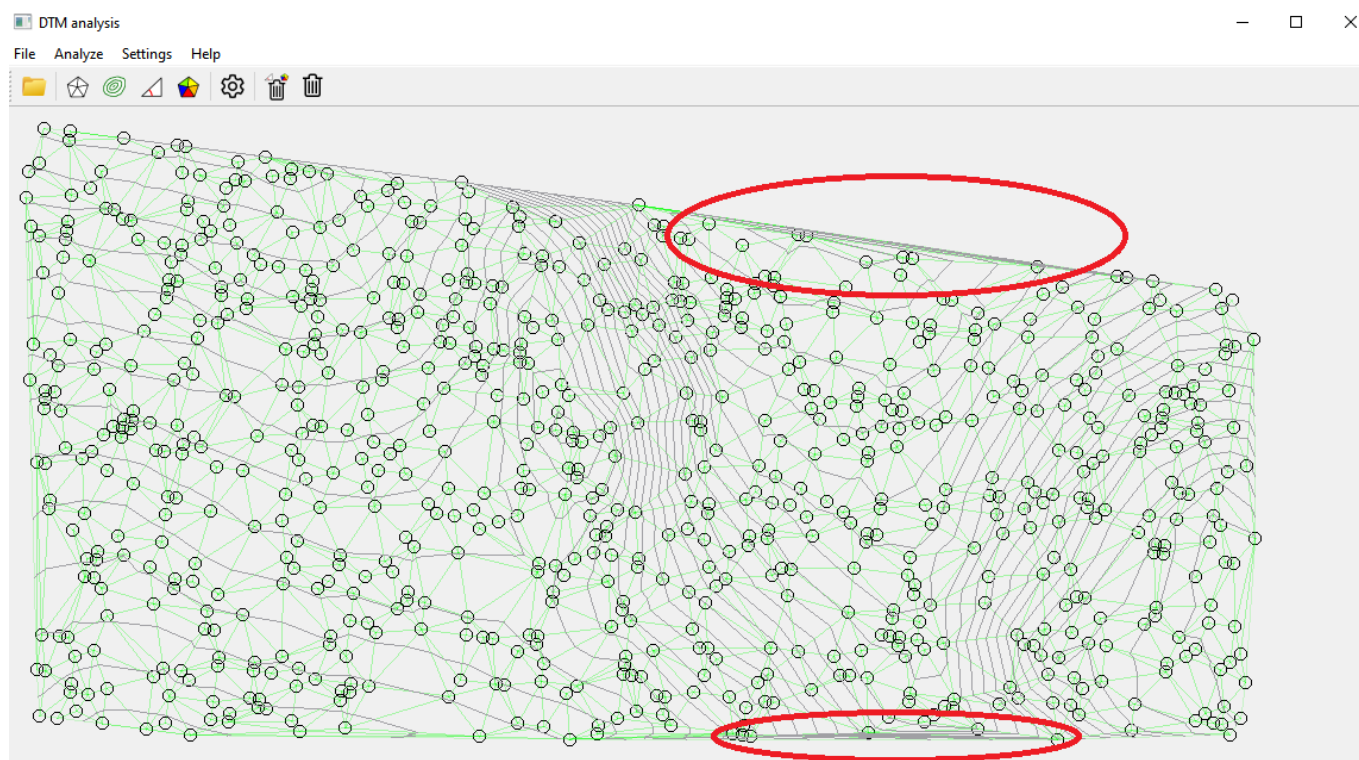


Obrázek 9.1: Polyedrický model na kterém se představí problémy algoritmu.

## 10 Dokumentace

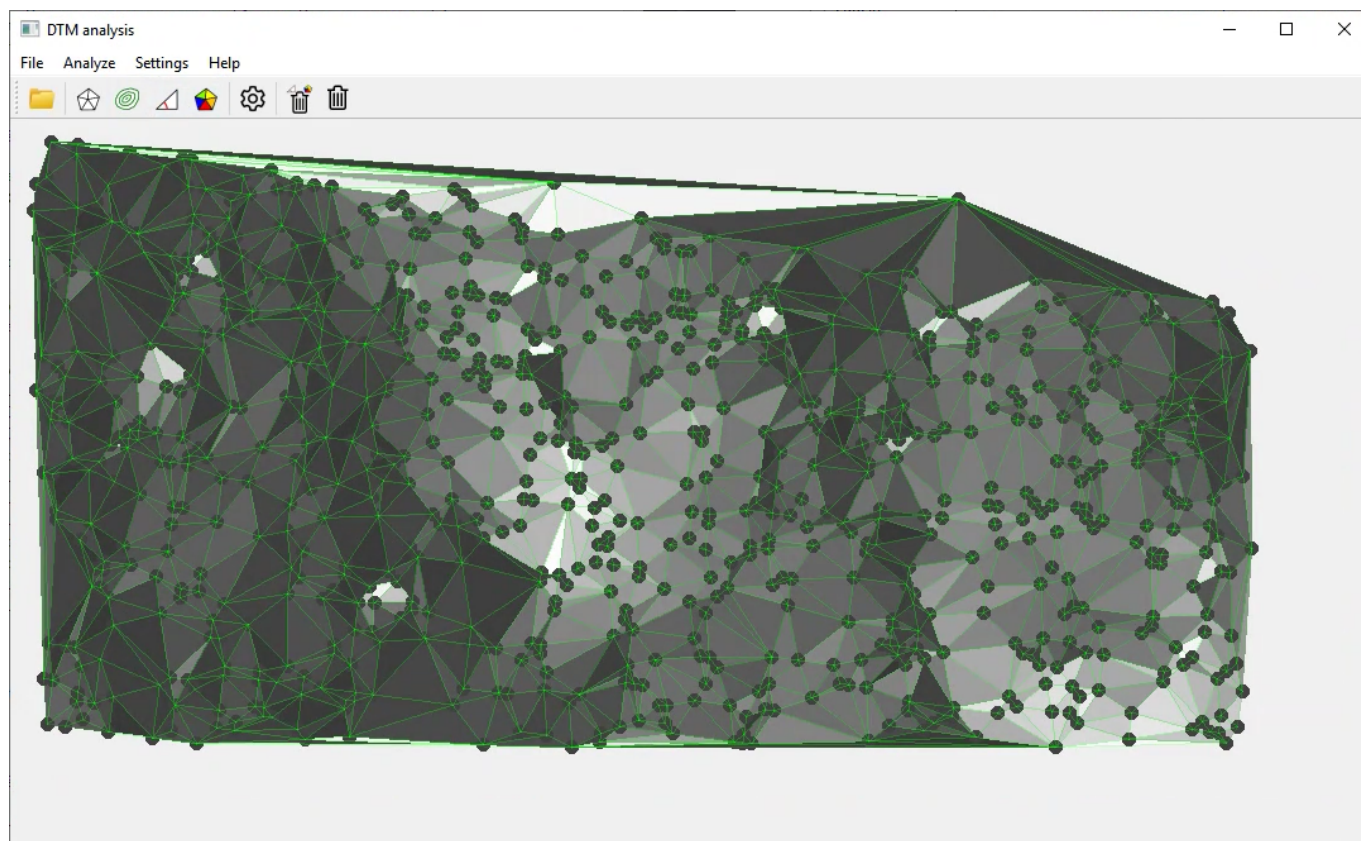
### 10.1 Třída Algorithms

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



Obrázek 9.2: Polyedrický model s červeným označením míst, kde se jedná o trojúhelníky, které špatně reprezentují skutečnost. Z důvodu dlouhých hran, odlehklých bodů Nejlepší možností je vytvořit funkci, která při výpočtu nebude zahrnovat tyto trojúhelníky.

- `int getDelaunayPoint(const QPoint3D &p1, const QPoint3D &p2, const std::vector< QPoint3D > &points);`  
– 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 > &acl);`  
– aktualizuje hranu v listu hran
- `std::vector< Edge > createContourLines(const std::vector< Edge > dt, double zmin, double zmax, double dz);`  
– 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
- `static double getMinSlope(std::vector< Triangle > &triangles);`  
– vrátí nejmenší hodnotu sklonu
- `static double getMaxSlope(std::vector< Triangle > &triangles);`  
– vrátí největší hodnotu sklonu



Obrázek 9.3: Polyedrický model - nahrané + klikané, analýza expozice (1)

## 10.2 Třída Draw

- *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

## 10.3 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

## 10.4 Třída MainForm

- `void on_actionOpen_triggered()`  
– otevře průzkumníka souborů
- `void on_actionCreate_DT_triggered()`  
– vytvoří DT
- `void on_actionExit_triggered()`  
– ukončí soubor
- `void on_actionCreate_contour_lines_triggered()`  
– vytvoří vrstevnice
- `void on_actionAnalyze_slope_triggered()`  
– vytvoří sklon svahu
- `void on_actionAnalyze_aspect_triggered()`  
– vytvoří orientaci svahu
- `void on_actionClear_slope_aspect_triggered()`  
– vymaže z Canvas vytvořenou analýzu sklonu nebo expozice
- `void on_actionClear_all_triggered()`  
– vyčistí Canvas

## 11 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].