

České vysoké učení technické v Praze

Fakulta stavební



Algoritmy v digitální kartografii

Úloha č. 3: Digitální model terénu a jeho analýzy

Skupina:

Sabina Kličková

Martin Vajner

Zimní semestr 2021/2022

## I. Obsah

1. Zadání .....	3
2. Bonusové úlohy .....	3
3. Popis a rozbor problémů.....	3
4. Popisy algoritmů .....	4
6. Vstupní data, formát vstupních dat, popis .....	7
7. Výstupní data, formát výstupních dat, popis .....	7
8. Dokumentaci: popis tříd, datových položek a jednotlivých metod .....	9
9. Zhodnocení algoritmů.....	11
11. Citovaná literatura .....	12
12. Seznam obrázků.....	12

## 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. 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. Zhodnoťte 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.

Delaunay triangulace, polyedrický model terénu.	+10b
Konstrukce vrstevnic, analýza sklonu a expozice.	+10b

## 2. Bonusové úlohy

V této úloze byly zpracovány následující bonusové úlohy:

Krok	Hodnocení
Automatický popis vrstevnic.	+3b
Výběr barevných stupnic při vizualizaci sklonu a expozice.	+3b

## 3. Popis a rozbor problémů

Zadáním úlohy je vytvořit aplikaci pomocí sw QT Creator. Cílem této aplikace je vytvořit DMT nad množinou bodů pomocí Delaunay triangulace. Množina bodů je vytvořena pomocí „klikání“ v grafickém okně či nahráním bodů ze souboru txt. DMT je vizualizováno pomocí vrstevnic s popisy a sklonu.

## 4. Popisy algoritmů

### Delaunay triangulace

Pro Delaunay triangulaci byla v této úloze použita metoda Inkrementální konstrukce.

Tato metoda vytváří trojúhelníkovou síť. V počátku se hledá minimální opsaná kružnice vedoucí bodem, k již vytvořené a orientované hraně. Existuje-li takový bod, změní se orientace hrany a hledání pokračuje dál.

Aktivním seznamem hran (AEL) se v této úloze rozumí ty hrany, ke kterým ještě nebyl nalezen třetí bod. Pokud je nalezena nová hrana, provede se kontrola, jestli již v seznamu není stejná hrana pouze s opačnou orientací. Pokud není, přidá se hrana do seznamu a hledání pokračuje. Když se k nějaké hraně v AEL nalezne třetí bod, je hrana ze seznamu odstraněna. „Výpočet“ končí, když je AEL prázdný. Delaunay triangulace maximalizuje minimální úhel v trojúhelníku a je jednoznačná neleží-li čtyři body na kružnici.

### Algoritmus:

- a. Nalezení pivota  $p1 = \min(x)$  a nalezení nejbližšího bodu k  $p1$
- b. Vytvoření hrany  $e(p1, p2)$
- c. Nalezení optimálního Delaunay bodu  $p$ 
  - a. Pokud  $p$  neexistuje, prohodí se orientace hrany  $e$  a hledání se opakuje
- d. Vytvoření hran trojúhelníku  $e2, e3$
- e.  $AEL \leftarrow e \_ AEL \leftarrow e2 \_ AEL \leftarrow e3$
- f.  $DT \leftarrow e \_ DT \leftarrow e2 \_ DT \leftarrow e3$
- g. Pokud AEL není prázdná:
  - a.  $AEL \rightarrow e$  //První hrana z AEL
  - b. Prohození orientace hrany  $e$
  - c. Nalezení optimálního Delaunay bodu  $p$
  - d. Pokud existuje  $p$ :
    - i. Vytvoření  $e2, e3$
    - ii.  $DT \leftarrow e, DT \leftarrow e2, DT \leftarrow e3$
    - iii. Prohození orientace hran  $e2, e3$
    - iv. Pokud je  $e2$  v AEL
      1.  $AEL \rightarrow e2$
    - v. Jinak  $AEL \leftarrow e2$
    - vi. Pokud je  $e3$  v AEL
      1.  $AEL \rightarrow e3$
    - vii. Jinak  $AEL \leftarrow e3$

### Tvorba vrstevnic

Vrstevnice je křivka spojující místa se stejnou nadmořskou výškou. Dělí se na základní a hlavní, kdy za hlavní je standartně považována každá pátá vrstevnice. Dalšími typy vrstevnic jsou pomocné a doplňkové.

Zobrazování vrstevnic závisí na jejich typu, kde základní jsou kresleny čarou tenkou, hlavní zvýrazněnou a doplňkové čerchovanou. Popis je přítomen standartně u vrstevnic hlavních, a to ve směru z kopce. Popis je rovnoměrně rozdělený po zobrazované oblasti, nejčastěji je popisována pouze hlavní vrstevnice.

Pro tuto úlohu byla použita metoda zobrazení pomocí lineární interpolace. Jejím základem je hledání průsečnice roviny trojúhelníku a vodorovné roviny ve výšce  $h$ . Body A, B této průsečnice jsou určeny z podobnosti trojúhelníků:

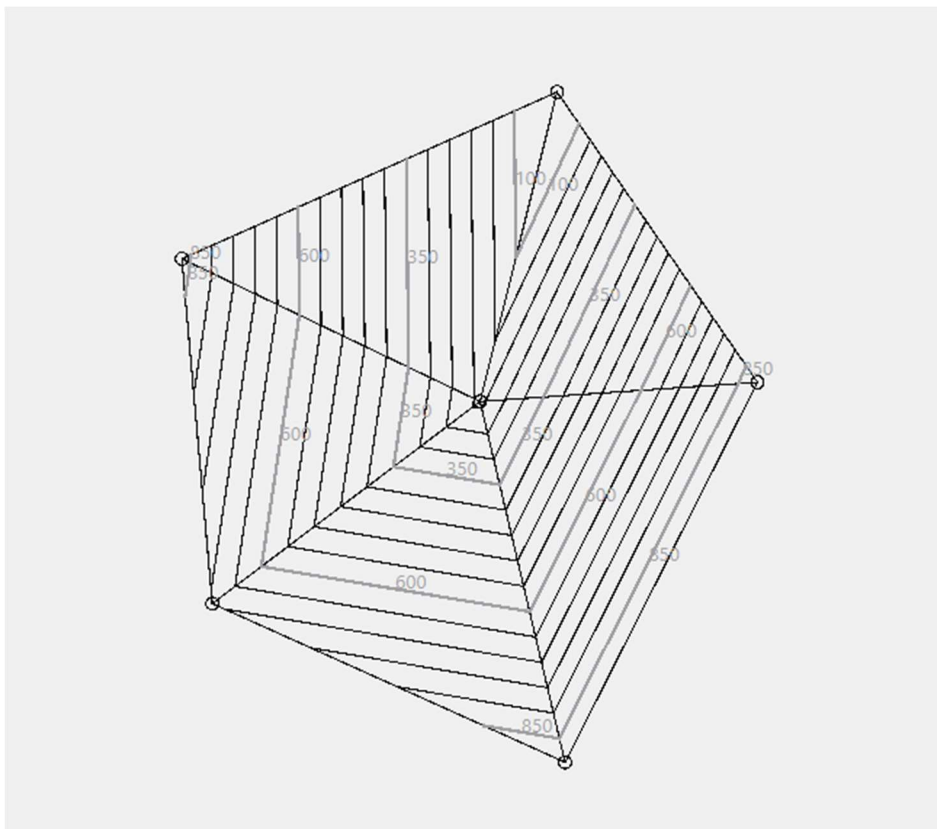
$$x_a = \frac{x_3 - x_1}{z_3 - z_1} * (z - z_1) + x_1$$

$$y_a = \frac{y_3 - y_1}{z_3 - z_1} * (z - z_1) + y_1$$

$$x_b = \frac{x_2 - x_1}{z_2 - z_1} * (z - z_1) + x_1$$

$$y_b = \frac{y_2 - x_1}{z_2 - z_1} * (z - z_1) + y_1$$

Po spojení těchto hran vznikne hrana tvořící v daném trojúhelníku vrstevnici o dané výšce.



*Příklad vypočtených vrstevnic*

### Expozice (Orientace)

Expozice označuje orientaci ke světové straně.

Výpočet x a y části normálového vektoru:

$$\begin{aligned} u_x &= \Delta x_{2,1}; & u_y &= \Delta y_{2,1}; & u_z &= \Delta z_{2,1}; \\ v_x &= \Delta x_{2,3}; & v_y &= \Delta y_{2,3}; & v_z &= \Delta z_{2,3}; \end{aligned}$$

$$\begin{aligned} n_x &= (u_y * v_z - u_z * v_y) \\ n_y &= -(u_x * v_z - u_z * v_x) \end{aligned}$$

$$\text{Expozice: } A = \text{atan2} \left( \frac{n_x}{n_y} \right)$$

### Sklon

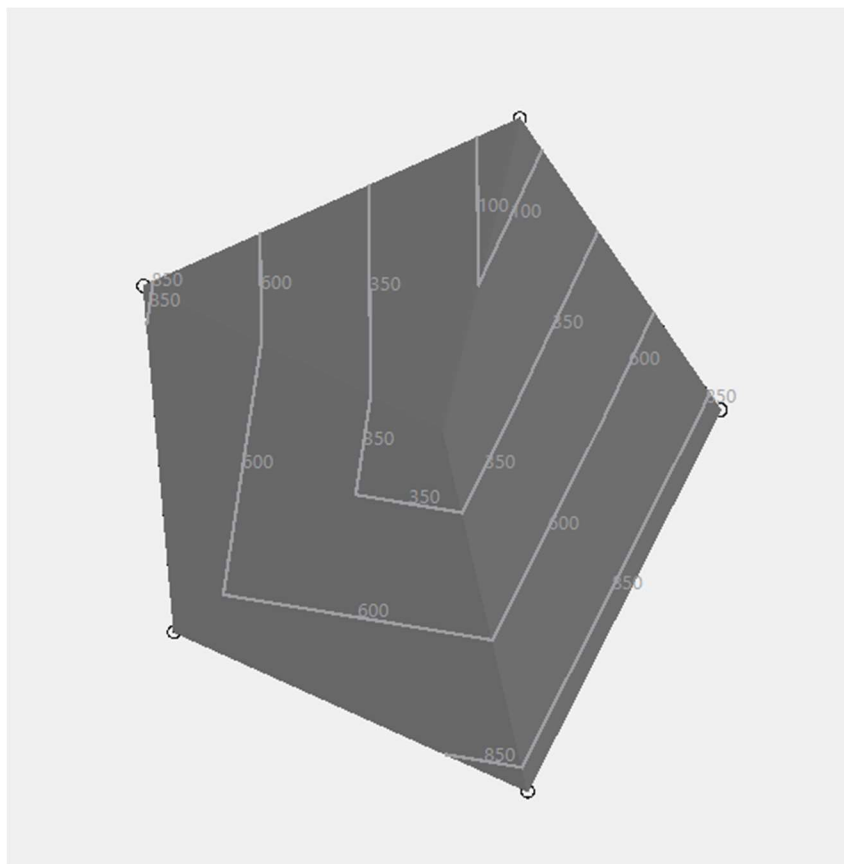
Sklon terénu definuje odchylka normálového vektoru roviny trojúhelníku od normálového vektoru vodorovné roviny.

Výpočet normálového vektoru roviny trojúhelníku:

$$\begin{aligned} u_x &= \Delta x_{2,1}; & u_y &= \Delta y_{2,1}; & u_z &= \Delta z_{2,1}; \\ v_x &= \Delta x_{2,3}; & v_y &= \Delta y_{2,3}; & v_z &= \Delta z_{2,3}; \end{aligned}$$

$$n_t = \sqrt{(u_y * v_z - u_z * v_y)^2 - (u_x * v_z - u_z * v_x)^2 + (u_x * v_y - u_y * v_x)^2}$$

$$\text{Sklon: } \varphi = \arccos \frac{n_z}{|n_t|}$$



Příklad barevného vykreslení sklonu

## 5. Popsání bonusových úloh

### Automatický popis vrstevnic

Popis vrstevnic byl generován pouze pro vrstevnice hlavní (každá 5. vrstevnice). Vrstevnice byly vykresleny funkcí `contourLines`. Z bodů hran o souřadnicích  $x$ ,  $y$ ,  $z$  byla získána hodnota  $z$  a vykreslena na vrstevnici. Umístění na vrstevnici realizováno průměrem souřadnic  $x$ ,  $y$  koncových bodů hran.

### Barevné stupnice

Nejprve byl vytvořen `ComboBox` pro výběr typu vykreslení sklonu s možnostmi vykreslení barevné či šedé stupnice. Poté byl zvolen způsob vykreslování, kde podle vzorce  $\text{barva} = 255 - \text{sklon} * 255 / \text{PI}$ . Tato hodnota pak byla přiřazena každému trojúhelníku pomocí `setBrush (QColor (col, col, col))`. U stupnice barevné byl použit systém intervalů, kde každý interval má jiný odstín barvy. Aplikace zobrazuje nejtmavší barvou místo s největším sklonem.

## 6. Vstupní data, formát vstupních dat, popis.

Vstupní data jsou body datového typu `std::vector<QPoint3D>`. Data byla generována z datové sady ArcCR500 a dále exportována v ArcGIS Pro do formátu ASCII (.txt).

Formát dat textového souboru:

`y << x << z`

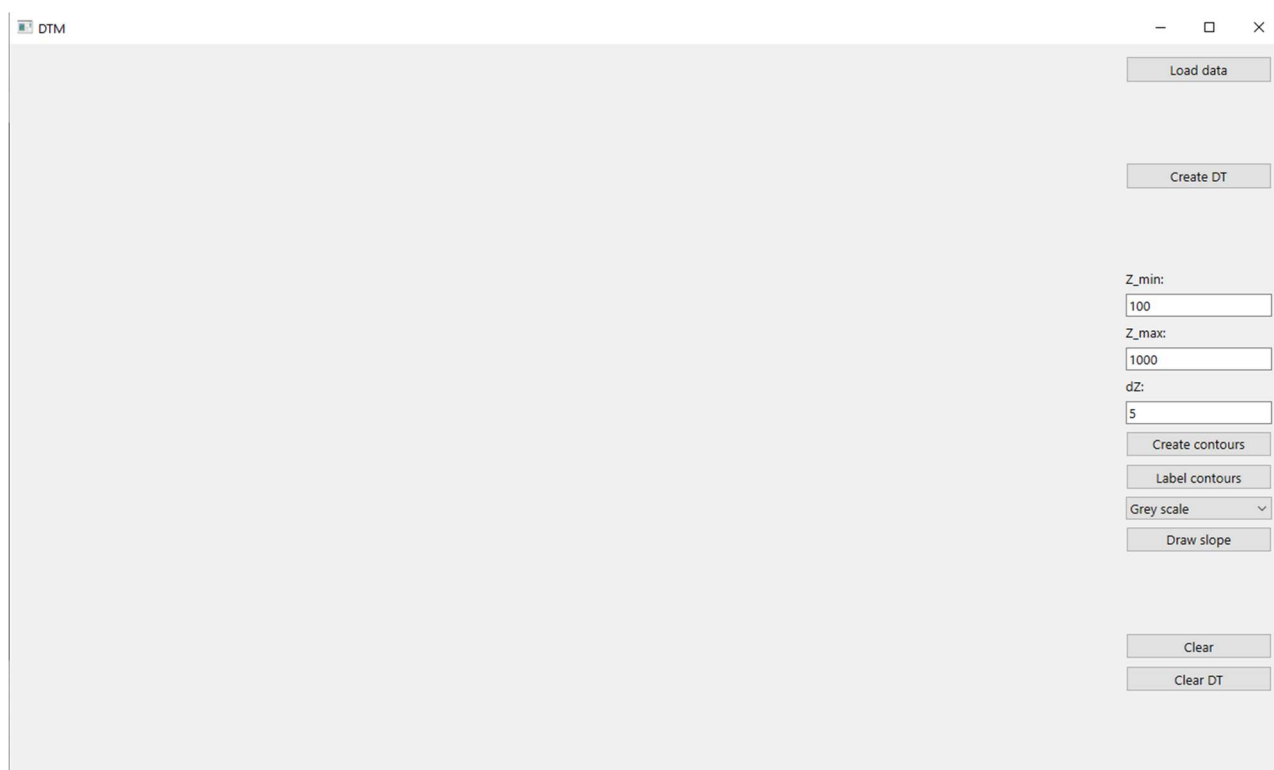
kde:

$x$  - souřadnice  $x$  bodu

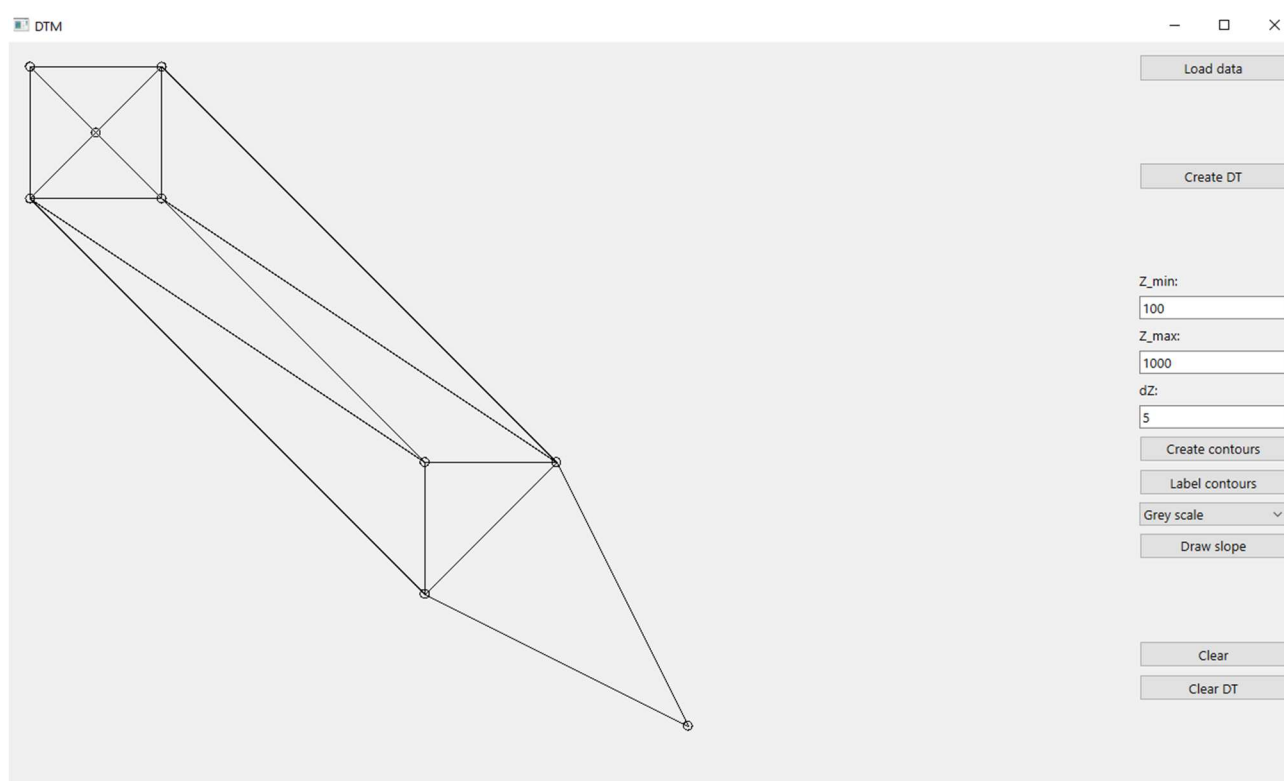
$y$  – souřadnice  $y$  bodu

## 7. Výstupní data, formát výstupních dat, popis

Výstupem je aplikace, která dokáže nahrát data a vytvořit nad nimi trojúhelníkovou síť. Rovněž dokáže vykreslit vrstevnice a jejich popis a sklon pomocí barevných škál.

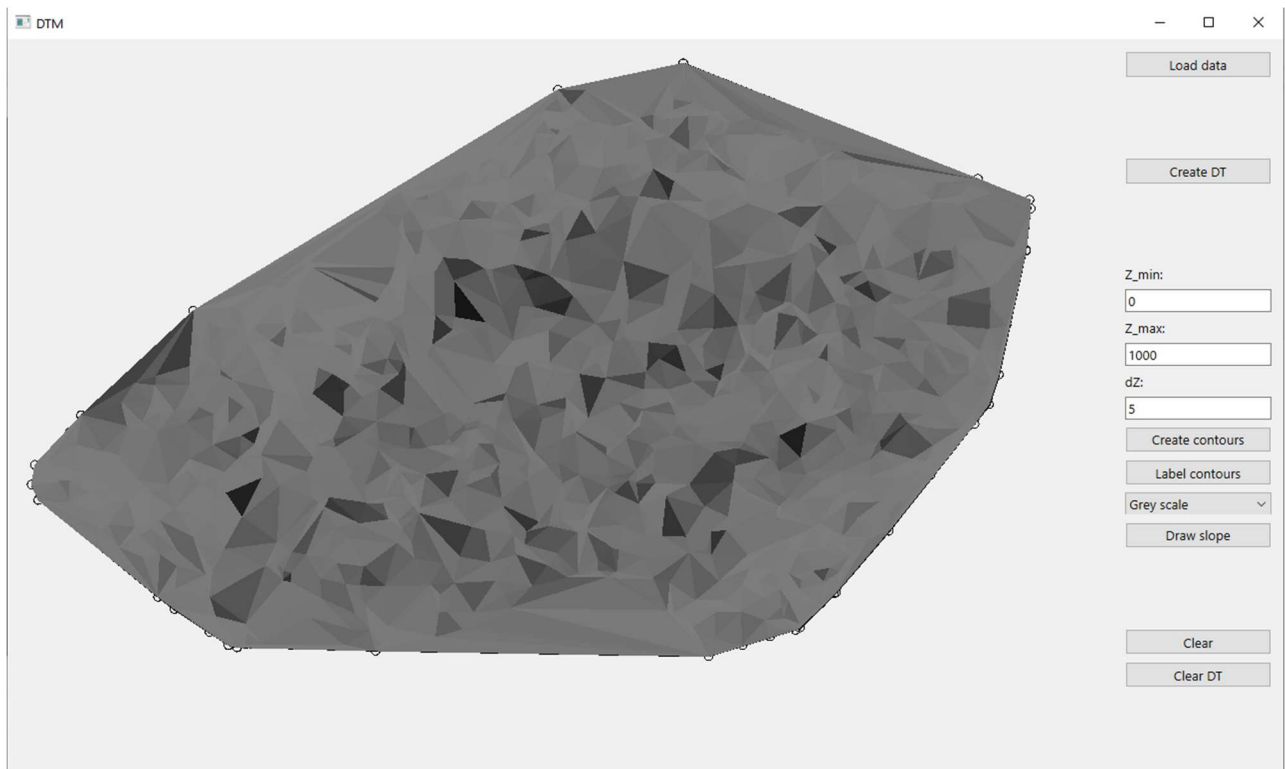


*Ukázka aplikace po spuštění*



*Ukázka aplikace po nahrání a vytvoření DT*





Ukázka aplikace po vykreslení sklonu

## 8. Dokumentaci: popis tříd, datových položek a jednotlivých metod

```
class Draw : public QWidget
private:
    std::vector<QPoint3D> points;
        - definice proměnné uložení jednotlivých bodů(x,y,z)
    std::vector<Edge> dt;
        - definice proměnné uložení Delaunay trojúhelníků
    std::vector<Edge> contours;
        - definice proměnné uložení vrstevnic
    std::vector<Triangle> triangles;
        - definice proměnné uložení trojúhelníků
    std::vector<Edge> contourslabel, contourslabel2, contoursmain;
        - definice proměnné uložení hlavních vrstevnic a jejich popisů
    boolean greycol, colors;
        - definice proměnné uložení nastaveného vykreslení barev
    int width,height;

public:
    explicit Draw(QWidget *parent = nullptr);
    void paintEvent(QPaintEvent *event);
        - vykreslení všech grafických prvků a jejich grafické zvýraznění
    void mousePressEvent(QMouseEvent *event);
        - přidání proměnné(bod) do seznamu pomocí grafického okna
    void clear();
        - mazání vykreslených proměnných
    std::vector<QPoint3D> getPoints(){return points;}
        - vrácení proměnných(bod)
    void setDT(std::vector<Edge> &dt_){dt = dt_;}
        - přiřazení vektoru Delaunay trojúhelníků
    std::vector<Edge> getDT(){return dt;};
```

```

- vracení vektoru Delaunay trojúhelníků
void setContours(std::vector<Edge> &contours_){contours = contours_;}
- přiřazování vrstevnic
std::vector<Edge> getContours(){return contours;};
- vracení vrstevnic
std::vector<Triangle> getTriangles(){return triangles;};
- vracení trojúhelníku pro funkci slope
void setTriangle(std::vector<Triangle> &triangles_){triangles = triangles_;}
- přiřazování trojúhelníku pro funkci slope
void clearDT();
- smazání vektoru Delaunay trojúhelníků
void setLabelContours(std::vector<Edge> &contourslabel_){contourslabel = contourslabel_;}
- přiřazení popisů vrstevnic
std::vector<Edge> & getLabelContours() {return contourslabel;};
- vracení popisů vrstevnic
void setMainContours(std::vector<Edge> &contoursmain_){contoursmain = contoursmain_;}
- přiřazení hlavní vrstevnice (tzn. každá pátá)
std::vector<Edge> & getMainContours() {return contoursmain;};
- vracení hlavních vrstevnic (tzn. každá pátá)

void setGreycol(bool &greycol_){greycol = greycol_};
- nakreslení vykreslování stupňů šedi profunkci slope
void setColors(bool &colors_){colors = colors_};
- nastavení vykreslování barevné škály pro funkci slope

void loadFile(std::string &path);
- načtení dat formátu ASCII(.txt)

int getWidth(){return width;};
int getHeight(){return height;};

void setWidth(int width_){width=width_};
- nastavení šířky QWidgetu pro vykreslení
void setHeight(int height_){height=height_};
- nastavení výšky QWidgetu pro vykreslení

class Algorithms
{
public:
    Algorithms();
    int getPointLinePosition(QPoint3D &a,QPoint3D &p1,QPoint3D &p2);
- zjištění vzájemné polohy přímky a bodu (vlevo, vpravo, hrana)
- vstup: souřadnice bodu q a lomových bodů polygonu
    std::tuple<QPoint3D,double> getCircleCenterAndRadius(QPoint3D &p1,QPoint3D &p2,QPoint3D &p3);
- zjištění středu a poloměru kruhu, který je definován pomocí tří bodů
    int getDelaunayPoint(QPoint3D &s,QPoint3D &e,std::vector<QPoint3D> &points);
- vracení tzv. Delaunayova bodu
    int getNearestPoint(QPoint3D &p, std::vector<QPoint3D> &points);
- vracení nejbližšího bodu
    std::vector<Edge> dT(std::vector<QPoint3D> &points);
- definice vektoru hran tvořených 3D body
    void updateAEL(Edge &e, std::list<Edge> &ael);
- update seznamu aktivních trojúhelníků
    QPoint3D getContourPoint(QPoint3D &p1, QPoint3D &p2, double z);
- vracení bodu pro vykreslení vrstevnic
    std::vector<Edge> getContourLines(std::vector<Edge> &dt, double zmin, double zmax, double dz);
- vracení vrstevnic
    double getSlope(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
- vracení sklonu
    double getExposition(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
- vracení expozice
    std::vector<Triangle> analyzeDTM(std::vector<Edge> &dt);

```

- výpočet analýzy DTM z Delaunay triangulace

## 9. Zhodnocení algoritmů

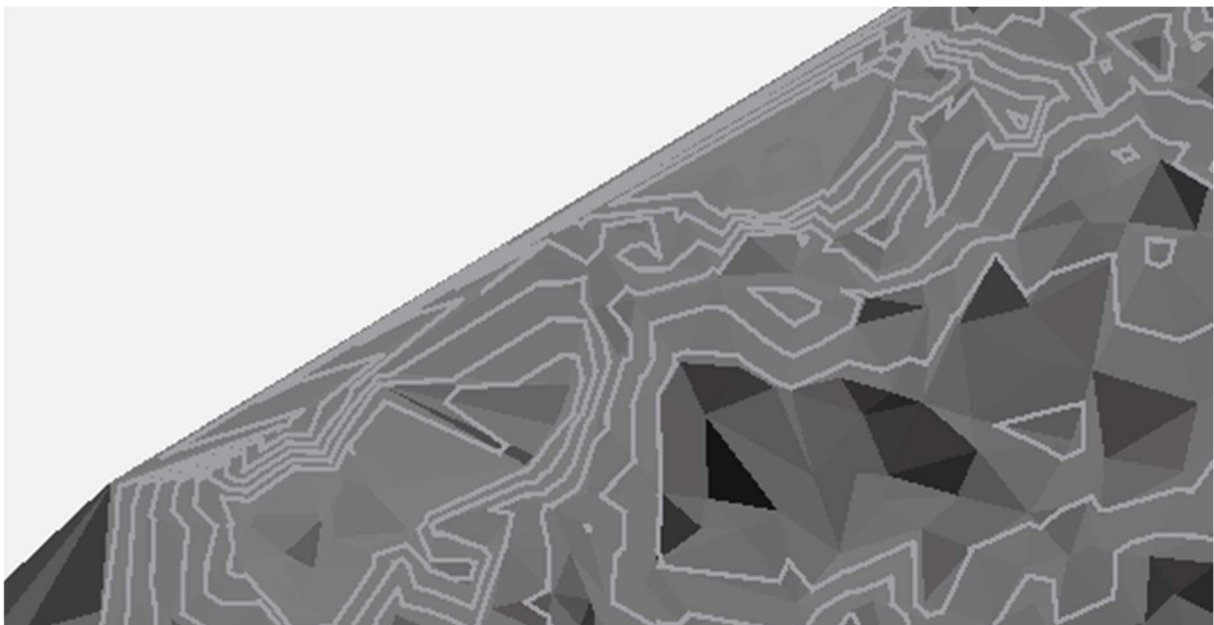
### Delaunay triangulace

Algoritmus selhává u dat ve formě gridu a pro některé terénní tvary. To by se případně řešilo pomocí přidání povinných hran. Z hlediska zobrazení reálného stavu by pak vizualizace více odpovídala skutečnosti. Aplikace je vhodná pro zpracování méně členitých území.

### Vrstevnice

Zvolená metoda Lineární interpolace nezobrazuje skutečný stav, jelikož předpokládá konstantní spád mezi dvěma body. Pro výsledek více realistický by tedy bylo vhodné užít jinou metodu interpolace. Při vykreslování je nutné znát nejmenší výšku v souboru.

Výpočet nedává reálné výsledky v okrajových částech Delaunayovy triangulace. Myslíme si, že tyto výsledky jsou způsobeny tím, že se vrstevnice vykreslují nad tupouhlým trojúhelníkem.



*Ukázka chyby při generování vrstevnic*

### Sklon a expozice

Analýza sklonu a expozice má tendenci selhávat pro nečlenité oblasti. Důvodem je zaokrouhlení. Rovinatý terén se pak zobrazuje jako nerovinný.

## 10. Závěr

Námi vytvořená aplikace dokáže načíst datovou sadu tvořenou body z formátu ASCII a vytvořit z ní trojúhelníkovou síť pomocí Delaunayovy triangulace. Z té pak dokáže vytvořit vrstevnice a jejich popisy a zároveň graficky zobrazit hodnoty sklonu pomocí barevných škál. Vstupní data byla generována z datové sady ArcCR500. Data byl exportována v ArcGIS Pro do formátu ASCII (.txt).

## 11. Citovaná literatura

1. **Tomáš, Bayer.** Perslonal page of Bayer Tomas. *Charles University of Prague*. [Online] [Citace: 05. 12 2021.] <https://web.natur.cuni.cz/~bayertom/index.php/teaching/algoritmy-v-digitalni-kartografii>.

## 12. Seznam obrázků

<i>Příklad vypočtených vrstevnic</i>	5
<i>Příklad barevného vykreslení sklonu</i>	6
<i>Ukázka aplikace po spuštění</i>	8
<i>Ukázka aplikace. po nahrání a vytvoření DT</i>	8
<i>Ukázka aplikace po vykreslení sklonu</i>	9
<i>Ukázka aplikace po vykreslení sklonu</i>	11