

České vysoké učení technické v Praze

Fakulta stavební



Algoritmy v digitální kartografii

### **Úloha 3: Digitální model terénu**

Skupina

Lucie Děkanová

Josef Pudil

Zimní semestr 2020/2021

## Obsah

1. Zadání .....	3
2. Údaje o bonusových úlohách .....	3
3. Popis a rozbor problémů .....	4
4. Popis algoritmů .....	4
4.1 Delaunayho triangulace .....	4
4.2 Konstrukce vrstevnic .....	5
4.3 Sklon DMT .....	6
4.4 Expozice DMT .....	6
5. Popsání bonusových úloh .....	6
5.1 Výběr barevných stupnic .....	6
5.2 Automatický popis vrstevnic .....	7
5.3 Algoritmy pro generování terénních tvarů .....	7
5.3.1 Kupa .....	7
5.3.2 Hřbet .....	8
5.3.3 Údolí .....	9
5.3.4 Spočinek .....	10
6. Vstupní data .....	11
7. Výstupní data .....	11
8. Dokumentace .....	11
8.1 Třída Algorithms .....	11
8.2 Třída Draw .....	12
9. Ukázka aplikace .....	14
10. Zhodnocení metod .....	17
10.1 Delaunayho triangulace .....	17
10.2 Vrstevnice .....	17
10.3 Sklon a expozice .....	17
Závěr .....	18
Seznam obrázků .....	18

## 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.

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.

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 nekonverzní 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>

## 2. Údaje o bonusových úlohách

Tyto 3 bonusové úlohy byly zpracovány:

- ❖ Automatický popis vrstevnic
- ❖ Výběr barevných stupnic při vizualizaci sklonu a expozice
- ❖ Algoritmus pro automatické generování terénních tvarů – kupa, údolí, spočinek, hřbet

### 3. Popis a rozbor problémů

Cílem úlohy bylo vytvoření DMT pomocí Delaunayho triangulace. Aplikace byla vytvořena v sw QT Creator. Nad množinou bodů byl vytvořen DMT, kdy lze body zadávat pomocí tlačítka myši, nebo automaticky vygenerovat jeden ze 4 terénních tvarů. DMT lze vizualizovat pomocí sklonu, expozice nebo pomocí vrstevnic.

### 4. Popis algoritmů

#### 4.1 Delaunayho triangulace

V této úloze byla naprogramována Delaunayho triangulace pomocí metody inkrementální konstrukce.

Pomocí této metody lze vytvořit trojúhelníkovou síť, DMT a je možné ji využít v rovině i prostoru. U této metody se hledá minimální opsaná kružnice, která vede nejvhodnějším bodem k již vytvořené orientované hraně a to v její polorovině. Když takový bod existuje, tak se změní orientace hrany a pokračuje se dále v hledání bodů.

Tento algoritmus má svůj seznam hran (AEL = Active Edge List), ke kterým nebyl ještě nalezen 3. bod. Pokud je nalezena nová hrana, je třeba zjistit, jestli není hrana s opačnou orientací již v seznamu. Pokud ne, nově nalezená hrana se vloží do seznamu a hledá se další. Pokud už seznam hranu obsahuje, tak se hrana nevloží do seznamu. Pokud se k nějaké hraně z AEL nalezne 3. bod, tak se hrana ze seznamu odstraní. Algoritmus pokračuje, dokud není AEL prázdný.

Uvnitř opsané kružnice libovolného trojúhelníku triangulace nesmí ležet žádný bod. Delaunayho triangulace maximalizuje minimální úhel v trojúhelníku, vůči kritériu minimálního úhlu je lokálně i globálně optimální a je jednoznačná, pokud 4 body neleží na kružnici.

**Algoritmus Delaunayho triangulace vypadá následovně:**

1. Nalezení pivota **p1**, **p1 = min(x)**, nalezení bodu **p2** nejbližšího k pivotu
2. Vytvoření hrany **e = (p1, p2)**
3. Nalezení optimálního Delaunayho bodu **p**
  - Pokud **p** nenalezen, prohod' orientaci **e**, **e = (p2, p1)**, znovu krok 3
5. Vytvoř zbývající hrany trojúhelníku, **e2 = (p2, p)** a **e3 = (p, p1)**
6. **AEL ← e**, **AEL ← e2**, **AEL ← e3**

7.  $DT \leftarrow e, DT \leftarrow e2, DT \leftarrow e3$

8. dokud AEL není prázdná:

- $AEL \rightarrow e, e = (p1, p2)$  // vezmi první hranu z AEL
- Prohození orientace  $e, e = (p2, p1)$
- Nalezení optimálního Delaunayho bodu  $p$
- Pokud  $p$  existuje:

$e2 = (p2, p), e3 = (p, p1)$

$DT \leftarrow e, DT \leftarrow e2, DT \leftarrow e3$

Prohození orientace  $e2$  a  $e3, e2' = (p, p2), e3' = (p1, p)$

Pokud  $e2'$  je v AEL

▪  $AEL \rightarrow e2'$

Jinak  $AEL \leftarrow e2$

Pokud  $e3'$  je v AEL

▪  $AEL \rightarrow e3'$

Jinak  $AEL \leftarrow e3$

#### 4.2 Konstrukce vrstevnic

Vrstevnice je křivka, která spojuje místa se stejnou nadmořskou výškou. Vrstevnice lze dělit na základní, hlavní nebo zvýrazněné, které jsou značeny většinou jako každá 5. základní vrstevnice. Dále jsou vrstevnice doplňkové a pomocné.

Základní vrstevnice jsou zobrazovány tenkou čarou, hlavní silnější a doplňkové čárkovanou čarou. Jejich popis by měl být orientován po směru (pata písma směřována z kopce). Popis by měl být rovnoměrný po celé zobrazované oblasti. Nejčastěji se popisují pouze vrstevnice hlavní. V této úloze byla použita metoda zobrazení vrstevnic pomocí lineární interpolace, kdy je rozestup mezi dvěma body konstantní.

#### Konstrukce vrstevnic:

Lineární interpolace je založena na analytické geometrii, kdy hledáme průsečnici roviny tvořené trojúhelníkem a vodorovné roviny s výškou  $h$ . Koncové body A a B této průsečnice se určují z podobnosti trojúhelníků.

$$xA = (x3 - x1) / (z3 - z1) \cdot (z - z1) + x1$$

$$yA = (y3 - y1) / (z3 - z1) \cdot (z - z1) + y1$$

$$xB = (x2 - x1) / (z2 - z1) \cdot (z - z1) + x1$$

$$yB = (y2 - y1) / (z2 - z1) \cdot (z - z1) + y1$$

Spojením těchto bodů vznikne hrana, která tvoří v daném trojúhelníku vrstevnici o dané výšce. Vrstevnice byly nakresleny pro případy, když strana trojúhelníku leží ve vodorovné rovině, vodorovná rovina protíná trojúhelník ve vrcholu a protilehlé straně a když vodorovná rovina protíná trojúhelník ve dvou stranách.

### 4.3 Sklon DMT

Sklon terénu je definován jako odchylka normálového vektoru roviny trojúhelníku od normálového vektoru vodorovné roviny.

**Algoritmus sklonu DMT:**

$$nx = u_y \cdot v_z - v_y \cdot u_z$$

$$ny = -(u_x \cdot v_z - v_x \cdot u_z)$$

$$nz = u_x \cdot v_y - v_x \cdot u_y$$

$$nt = \sqrt{nx^2 + ny^2 + nz^2}$$

$$\varphi = \arccos(nz/nt)$$

### 4.4 Expozice DMT

Expozice označuje orientaci terénu vůči slunci. Značí, na jakou část terénu dopadá kolik světla.

**Expozice DMT:**

$$nx = u_y \cdot v_z - v_y \cdot u_z$$

$$ny = -(u_x \cdot v_z - v_x \cdot u_z)$$

$$A = \arctan2(nx/ny)$$

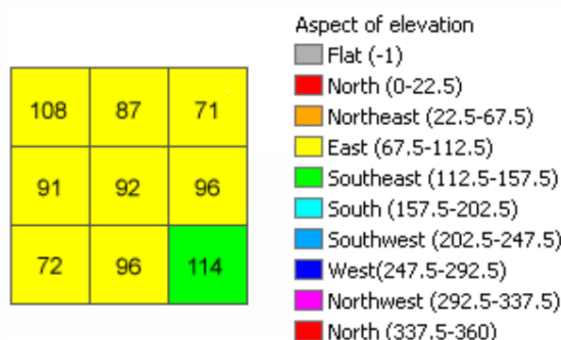
## 5. Popsání bonusových úloh

### 5.1 Výběr barevných stupnic

V úloze byla jako první bonusová vybrán výběr barevných stupnic při vizualizaci sklonu a expozice.

V aplikaci byl vytvořen ComboBox, kde bylo možné si vybrat mezi odstíny šedi a barevnou stupnicí při zobrazení sklonu i expozice. U výběru v odstínech šedi zobrazuje aplikace nejtmavší barva (černá) místa s největším sklonem, s nejmenším jsou zobrazena téměř bíle. U barevné škály byla vybrána červená barva jako místa s největším sklonem, světle zelená jako místa s nejmenším. Barevná škála byla vybrána v odstínech hnědé.

Barevná škála (stejně jako intervaly) pro expozici byla inspirována prostředím ArcGIS, ale namíchání barev bylo zvoleno trochu odlišné (více pastelových barev).



Obrázek 1: Intervaly expozice z ArcGIS

Vytvoření barevných stupnic bylo provedeno v třídě Draw, kde byly jednotlivé trojúhelníky procházeny a ke každému vybrána jedna barva podle analýzy. Barevné stupnice jsou nejlépe vidět při analýze nad konkrétním generovaným terénním tvarem.

## 5.2 Automatický popis vrstevnic

Automatický popis byl vytvořen pro každou 5.vrstevnici (hlavní). Pomocí funkce contourLines byly vrstevnice vykresleny. Výšky byly vykresleny pomocí funkce labelContours. Hrany vrstevnic jsou tvořeny dvěma body o souřadnicích X, Y, Z. Odtud je zobrazená hodnota získána a je umístěna na střed dané vrstevnice. Toho je docíleno zprůměrováním souřadnic X, Y počátečního a koncového bodu.

## 5.3 Algoritmy pro generování terénních tvarů

### 5.3.1 Kupa

Kupa je generována s pomocí náhodných bodů, kdy je spočítán bod umístěný v těžišti všech bodů a přidělí se mu výška Z (X,Y jsou převzaty z náhodných bodů). Výšky jsou přiděleny tak, že čím dále jsou od středu, tím je jejich výška menší.

Výpočet těžiště:

$$xT = \sum x_{i=1} / n ,$$

$$yT = \sum y_{i=1} / n ,$$

**n.....celkový počet bodů**

## **generateStack**

1. Generování náhodných bodů
2. Výpočet těžiště
3. Přidělení výšky těžišti bodů
4. Přidání těžiště do vektoru bodů
5. Pro každý bod ve vektoru bodů:
  - Určení vzdálenosti bodu od těžiště
  - Přiřazení výšky bodu nepřímo úměrné jeho vzdálenosti od těžiště

### **5.3.2 Hřbet**

Tento generátor určí náhodně, jestli bude hřbet ve směru osy X nebo osy Y. Podle toho setřídí body. Prvnímu a poslednímu bodu v množině se přidělí výšky.

Poté se spočítá bod v těžišti a přidělí se mu výška prvního bodu. U zbylých náhodně vygenerovaných bodů se zjistí jejich poloha vůči počátečnímu a koncovému bodu hřbetu (jejich vzdálenost) a podle toho, k jakému bodu jsou blíže, se jim přidělí náhodná výška (menší, než je výška počátečního bodu hřbetu).

## **generateRidge**

1. Vygenerování náhodných bodů
2. Náhodné určení orientace hřbetu (horizontálně či vertikálně vůči kreslicímu oknu)
3. Pokud horizontálně
  - Setřídění bodů podle Xjinak
  - Setřídění bodů podle Y
4. Přidělení výšky prvnímu a poslednímu bodu ze setříděných bodů
5. Výpočet těžiště bodů + přidělení výšky
6. Přidání těžiště do vektoru bodů



7. Pro všechny body:  $i = 1, i < \text{velikost vektoru bodů} - 1$

- $d1$  //vzdálenost bodu od prvního bodu
- $d2$  //vzdálenost bodu od předposledního bodu
- pokud  $d1 < d2$ 
  - přidělení výšky nepřímo úměrné vzdálenosti bodu od prvního bodu
- jinak
  - přidělení výšky nepřímo úměrné vzdálenosti bodu od posledního bodu

### 5.3.3 Údolí

Údolí se tvoří stejně jako hřbet, jen s tím rozdílem, že prvnímu a poslednímu bodu setříděné množiny bodů je přiřazena nízká výška a zbylým bodům se úměrně na vzdálenosti přiřazuje náhodná vyšší výška.

#### **generateValley**

1. Vygenerování náhodných bodů

2. Náhodné určení orientace údolí (horizontálně či vertikálně vůči kreslicímu oknu)

3. Pokud horizontálně

- Setřídění bodů podle X

jinak

- Setřídění bodů podle Y

4. Přidělení výšky prvnímu a poslednímu bodu ze setříděných bodů

5. Výpočet těžiště bodů + přidělení výšky

6. Přidání těžiště do vektoru bodů

7. Pro všechny body:  $i = 1, i < \text{velikost vektoru bodů} - 1$

- $d1$  //vzdálenost bodu od prvního bodu
- $d2$  //vzdálenost bodu od předposledního bodu
- pokud  $d1 < d2$ 
  - přidělení výšky přímo úměrné vzdálenosti bodu od prvního bodu
- jinak
  - přidělení výšky přímo úměrné vzdálenosti bodu od posledního bodu

#### 5.3.4 Spočinek

Primární výška byla přiřazena dvou krajním bodům. Zbytku byla udělena výška podle vzdálenosti od dvou krajních bodů. Poté byla zjištěna výška bodu v polovině datasetu a přidělena dalším  $n/4$  bodům datasetu.

#### **generateSettling**

1. Vygenerování náhodných bodů

2. Náhodné určení orientace spočinku vůči kreslicímu oknu

3. Pokud horizontálně

- Setřídění bodů podle X

jinak

- Setřídění bodů podle Y

4. Nastavení výšky spočinku, tj. přidělení výšky prvnímu a druhému bodu

5. Pro všechny body:  $//i = 2$

- Vypočtení vzdálenosti bodu od prvního bodu
- Přidělení výšky bodu nepřímo úměrně vzdálenosti bodu od spojnice prvních dvou bodů

6. Pro čtvrtinu bodů z druhé poloviny datasetu:

- $\text{points}[i].z() = \text{points}[\text{points.size()}/2].z()$

Pro zbývající body:

- 14)  $\text{points}[i].z() = \text{points}[i].z() + dz // dz = z1 - z2$

$// z1 = \text{points}[\text{points.size()}/2].getZ();$

$// z2 = \text{points}[\text{points.size()}/2 + n/4].getZ();$

## 6. Vstupní data

Výstupní data jsou body datového typu `std::vector<QPoint3D>`. Tato množina bodů je generována s předem určeným počtem uživatelem při stisknutí tlačítka **Generate Points**. Dále je možné za vstupní data považovat body nahrané z textového souboru ve formátu X,Y,Z, lze vložit body pomocí myši.

## 7. Výstupní data

Výstupem je aplikace, která na vstupních datech vytvoří Delaunayho triangulaci. Dále je možné vykreslit vrstevnice, přidat jim automatický popis, analyzovat sklon nebo expozici terénu.

## 8. Dokumentace

### 8.1 Třída Algorithms

```
int getPointLinePosition(QPoint3D &q, QPoint3D &p1, QPoint3D &p2);
```

Funkce, která určuje, jestli se bod nachází v práve, nebo levé části od přímky. Vstupem je bod a 2 body linie.

```
void circleCenterAndRadius(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3,  
double &r, QPoint3D &s);
```

Funkce, která získá střed kružnice a zároveň i poloměr. Vstupem jsou 3 body.

```
int findDelaunayPoint(QPoint3D &pi, QPoint3D &pj,  
std::vector<QPoint3D> &points);
```

Funkce, určující Delaunayského bodu ideálního pro Delaunayho triangulaci.

```
double dist(QPoint3D &p1, QPoint3D &p2);
```

Funkce určující vzdálenost dvou bodů. Vstupem jsou 2 body.

```
double getPointLineDistance(QPoint3D &q, QPoint3D &p1, QPoint3D &p2);
```

Funkce určující vzdálenost linie a bodu. Vstupem je bod a 2 body linie.

```
int getNearestpoint(QPoint3D &p, std::vector<QPoint3D> &points);
```

Funkce, která nalezne nejbližší bod. Vstupem je bod a množina bodů

```
std::vector<Edge> DT(std::vector<QPoint3D> &points);
```

Funkce, která vytvoří Delaunayho triangulaci. Vstupem je množina bodů.

```
void updateAEL(Edge &e, std::list<Edge> &ael);
```

Funkce, který mění orientaci hrany, případně odmazává.

```
QPoint3D getContourPoint(QPoint3D &p1, QPoint3D &p2, double z);
```

Funkce, která určuje bod ležící na vrstevnici.

```
std::vector<Edge> contourLines(std::vector<Edge> &dt, double z_min,  
double z_max, double dz);
```

Funkce generující vrstevnice. Vstupem je minimální z maximální z a po jakém intervalu se mají vytvářet. Vstupem je také množina hran..

```
double getSlope(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
```

Funkce, která vypočítá sklon trojúhelníku. Vstupem jsou 3 body.

```
double getAspect(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3);
```

Funkce, který vypočítá orientaci trojúhelníku. Vstupem jsou 3 body.

```
std::vector<Triangle> analyzeDTM(std::vector<Edge> &dt);
```

Funkce, která analyzuje DTM a vykreslí podle zadání sklonu, nebo orientace trojúhelníků.

```
std::vector<QPoint3D> generateStack(int n, int height, int width);
```

Funkce generující kupu.

```
std::vector<QPoint3D> generateRandomPoints(int n, int height, int width);
```

Funkce generující náhodné body.

```
std::vector<QPoint3D> generateRidge(int n, int height, int width);
```

Funkce generující hřbet

```
std::vector<QPoint3D> generateValley(int n, int height, int width);
```

Funkce generující údolí

```
std::vector<QPoint3D> generateSettling(int n, int height, int width);
```

Funkce generující spočinek.

## 8.2 Třída Draw

```
void setPoints(std::vector<QPoint3D> &points_){points=points_;}
```

Funkce pro nastavení bodů.

```
std::vector<QPoint3D> & getPoints(){return points;}
```

Funkce pro získání bodů.

```
void setDT(std::vector<Edge> &dt_){dt = dt_;}
```

Funkce pro nastavení Delaunayho triangulace

```
std::vector<Edge> & getDT(){return dt;}
```

Funkce pro získání Delaunayho triangulace.

```
void setContours(std::vector<Edge> &contours_){contours = contours_;}
```

Funkce pro nastavení vrstevnic.

```
std::vector<Edge> & getContours(){return contours;}
```

Funkce pro získání vrstevnic.

```
void setDTM(std::vector<Triangle> &dtm_){dtm = dtm_;}
```

Funkce pro nastavení DTM.

```
std::vector<Triangle> & getDTM(){return dtm;}
```

Funkce pro získání DTM.

```
void loadFile(std::string &path, std::vector<QPoint3D> &points, int height,  
int width, double &min_z, double &max_z);
```

Funkce, pro načtení souboru.

```
void setSlope(bool &slope_){slope = slope_;}
```

Funkce pro nastavení Sklonu.

```
void setAspect(bool &aspect_){aspect = aspect_;}
```

Funkce pro nastavení orientace svahu.

```
void setPanchromatic(bool &panchromatic_){panchromatic = panchromatic_;}
```

Funkce pro nastavení odstínů šedi.

```
void setColorful(bool &colorful_){colorful = colorful_;}
```

Funkce pro nastavení barevnosti.

```
void setLabelContours(std::vector<Edge> &label_contours_)  
{label_contours = label_contours_;}
```

Funkce pro nastavení popisu vrstevnic.

```
std::vector<Edge>& getLabelContours() {return label_contours;}
```

Funkce pro získání popisu vrstevnic.

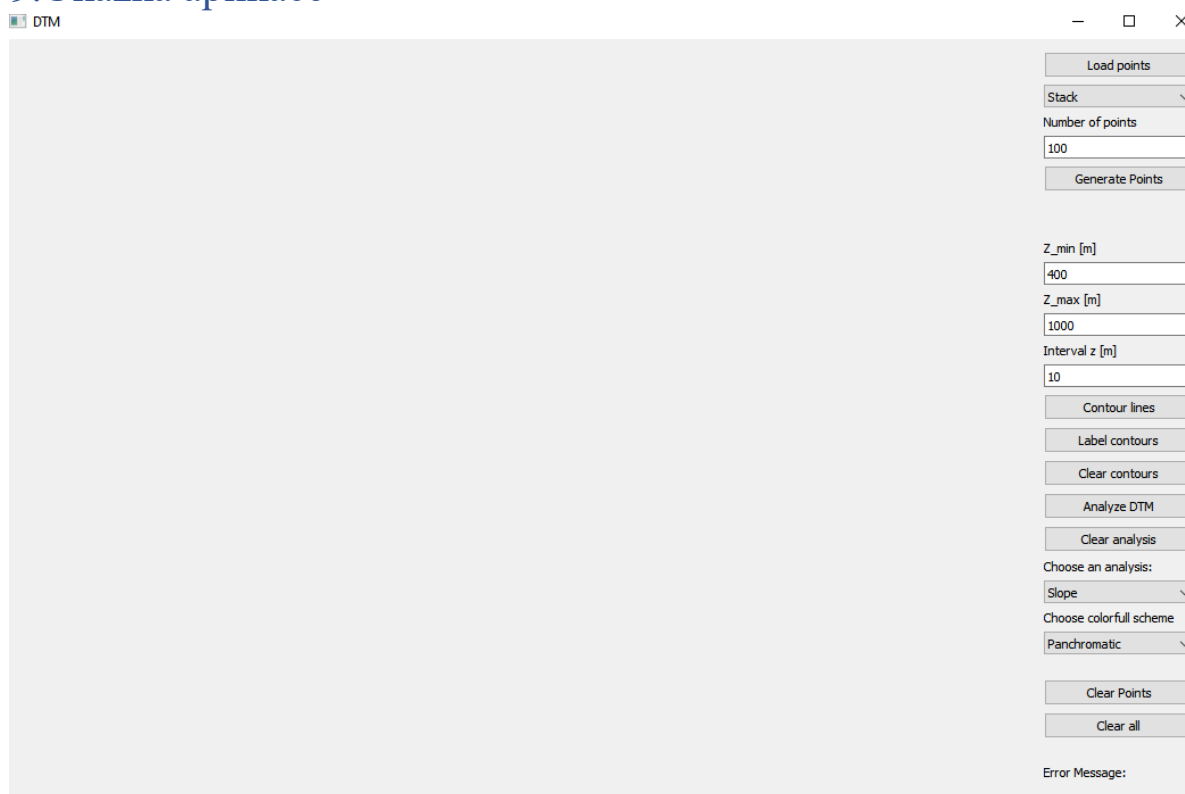
```
void setMainContours(std::vector<Edge> &main_contours_)  
{main_contours = main_contours_;}
```

Funkce pro nastavení hlavních vrstevnic.

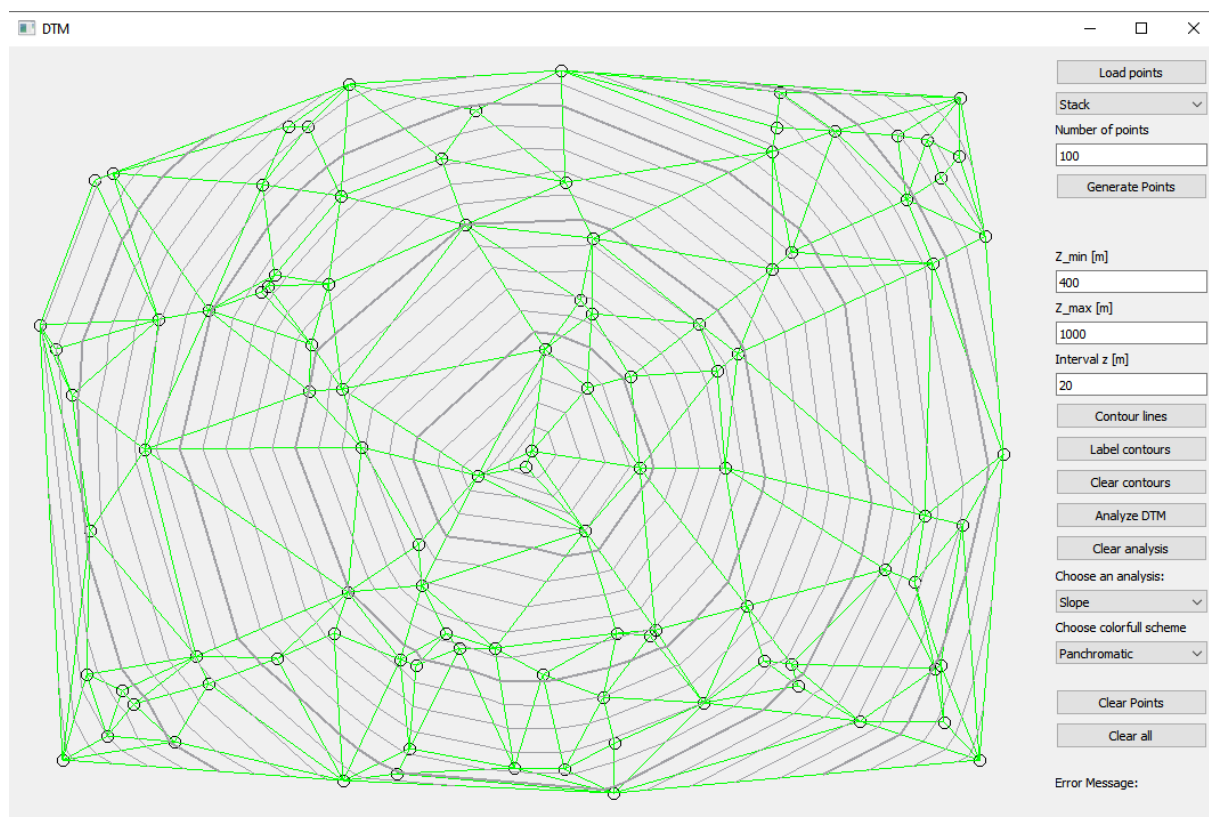
```
std::vector<Edge>& getMainContours() {return main_contours;}
```

Funkce pro získání hlavních vrstevnic.

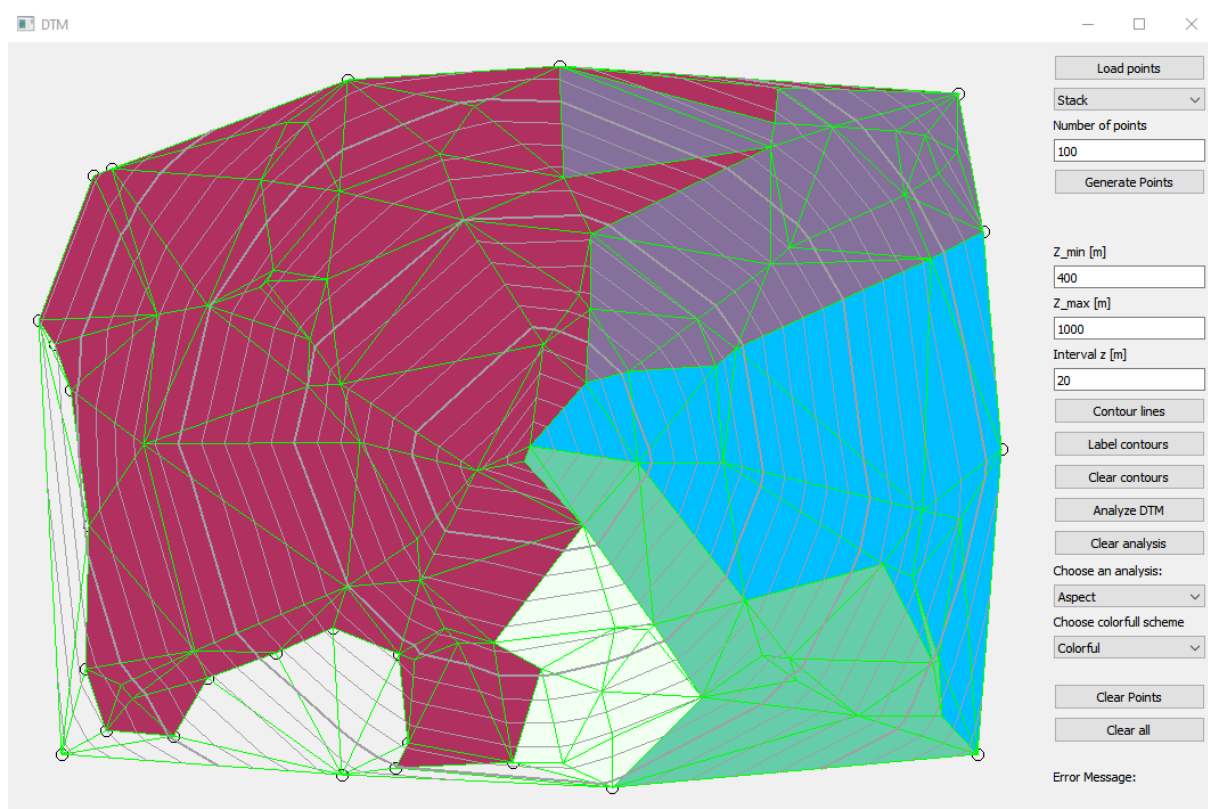
## 9. Ukázka aplikace



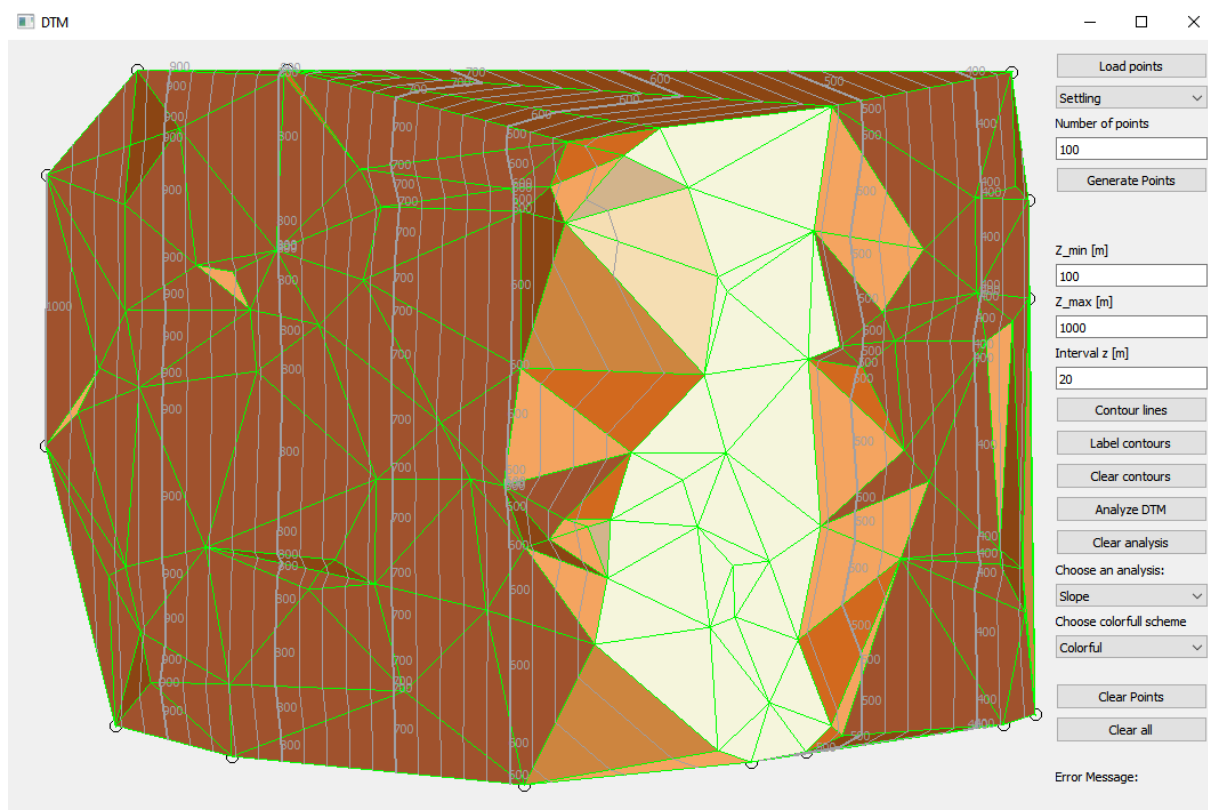
Obrázek 2: Ukázka základního tvaru aplikace



Obrázek 3: Vytvořený kopec s vygenerovanými vrstevnicemi

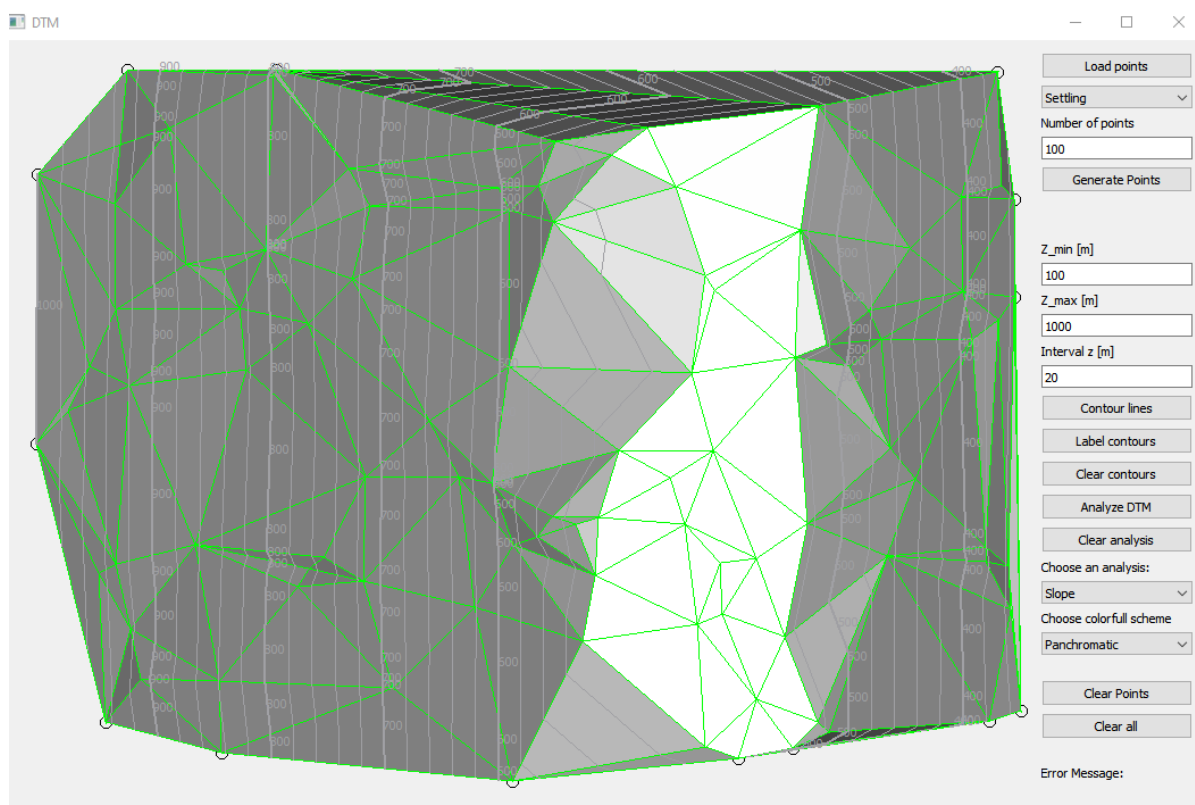


Obrázek 4: Analýza orientace svahu barevně

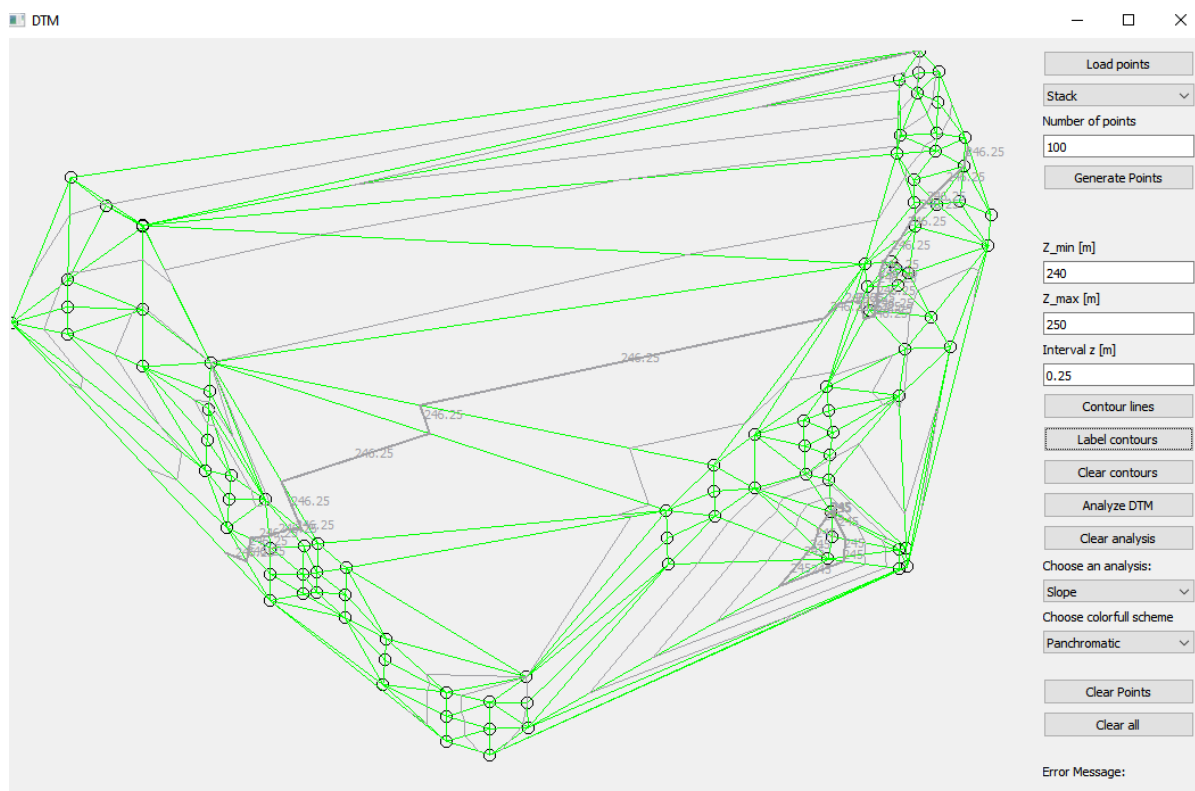


Obrázek 5: Ukázka aplikace s barevným vykreslením sklonu



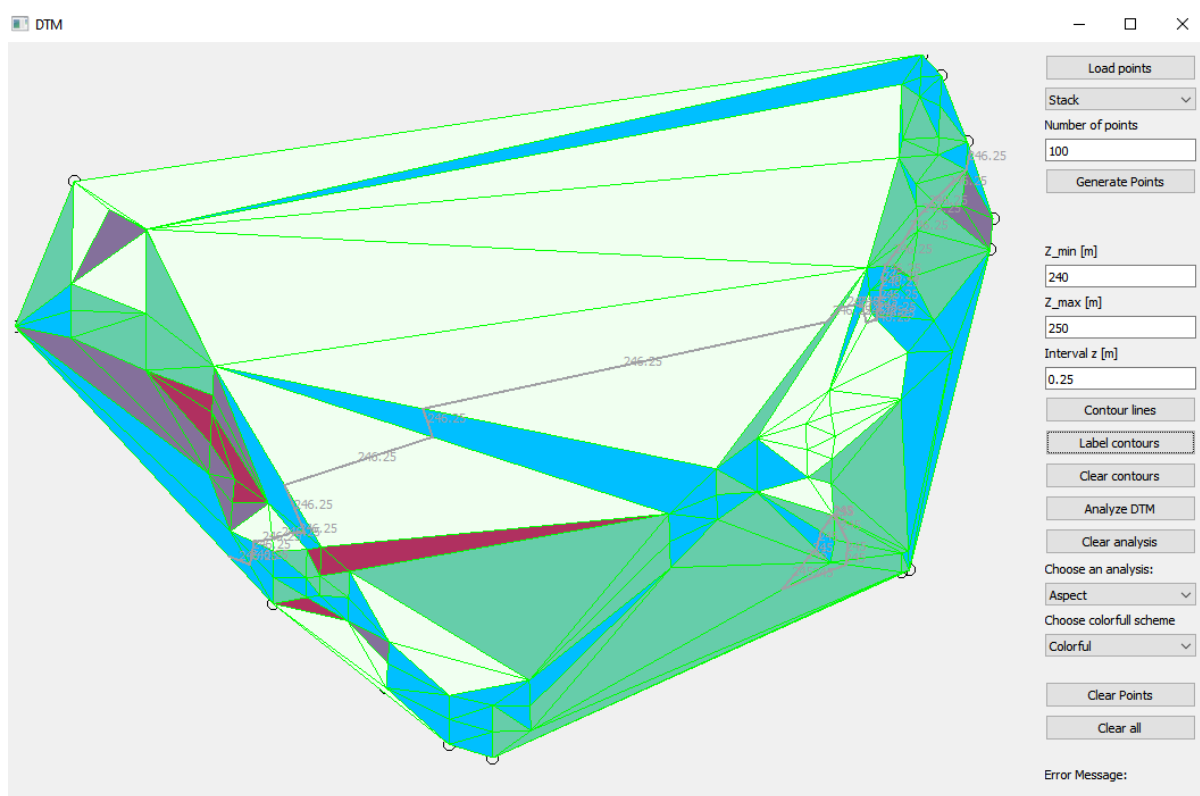


Obrázek 6: Vygenerování počinku se sklonem terénu v odstínech šedi



Obrázek 7: Vytvořený model z námi zadáných souřadnic





Obrázek 8: Vytvořený model s orientací terénu se známými souřadnicemi

## 10. Zhodnocení metod

### 10.1 Delaunayho triangulace

Tento algoritmus není jednoznačný pro body na gridu. Selhává pro některé terénní tvary. Tyto problémy by se daly vyřešit přidáním povinných hran. Výstupy by byly určitě lepší z hlediska reálné vizuální podoby terénu. Aplikace je dobrá pro málo členitá výškopisná data bez povinných hran.

### 10.2 Vrstevnice

Lineární interpolace vrstevnic nevystihuje příliš realitu. Tento algoritmus předpokládá konstantní spád mezi dvěma body. Pro lepší výsledek by bylo lepší použít geomorfologickou interpolaci. Tato tvorba vrstevnic je ovšem velmi náročná na výpočet.

Vrstevnice jsou občas špatně zobrazeny u tvaru sedla, kdy se vytváří kolem vrcholů malé polygony.

### 10.3 Sklon a expozice

Analýza sklonu a expozice selhává pro rovinná území. Toto je způsobeno zaokrouhlením, kde takto způsobené chyby zkreslují výsledek – rovinatý terén se zobrazuje jako nerovinný.

## Závěr

Byla vytvořena aplikace, která umí vygenerovat, nebo načíst množinu bodů a následně z ní vytvořit Delaunayho triangulaci, následně vypočítat vrstevnice a sklon a orientaci terénu.

Vygenerovat množiny typu kupa, hřbet, údolí a spočinek. Dále také může popsat vrstevnice a vytvořit hlavní a vedlejší s popisem pouze hlavních (tedy každé páté).

Data byla použita z bakalářské práce Karla Nováka s jeho plným vědomím. Jedná se o bakalářskou práci Účelové komunikace v obci Suhrovice. Polohopisné a výškové zaměření.

## Seznam obrázků

Obrázek 1: Intervaly expozice z ArcGIS .....	7
Obrázek 2: Ukázka základního tvaru aplikace .....	14
Obrázek 3: Vytvořený kopec s vygenerovanými vrstevnicemi .....	14
Obrázek 4: Analýza orientace svahu barevně .....	15
Obrázek 5: Ukázka aplikace s barevným vykreslením sklonu.....	15
Obrázek 6: Vygenerování počinku se sklonem terénu v odstínech šedi .....	16
Obrázek 7: Vytvořený model z námi zadaných souřadnic.....	16
Obrázek 8: Vytvořený model s orientací terénu se známými souřadnicemi.....	17