

České vysoké učení technické v Praze
Fakulta stavební

155ADKG: Digitální model terénu

Michael Kala
Anna Zemánková

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.

Zhodnot'te výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabunami 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 tři 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 |
| Max celkem: | 65b |

Čas zpracování: 3 týdny

2 Údaje o bonusových úlohách

Z bonusových úloh byly zpracovány:

- výběr barevné stupnice při vizualizaci expozice (sklon byl vizualizován již na hodině)
- automatický popis vrstevnic
- barevná hypsometrie

3 Popis a rozbor problému

Mějme množinu bodů $P\{p_i\}$, $p_i = \{x_i, y_i, z_i\}$. Nad touto množinou chceme vytvořit síť trojúhelníků t_j pomocí Delaunay triangulace DT , následně vytvořit vrstevnice a pro vizualizaci DMT určit sklon a expozici jednotlivých trojúhelníků.

3.1 Delaunay triangulace

Vlastnosti:

- Uvnitř kružnice opsané trojúhelníku $t_j \in DT$ neleží žádný jiný bod množiny P .
- DT maximalizuje minimální úhel v $\forall t_j$, avšak DT neminimalizuje maximální úhel v t_j .
- DT je lokálně optimální i globálně optimální vůči kritériu minimálního úhlu.
- DT je jednoznačná, pokud žádné čtyři body neleží na kružnici.

3.2 Vrstevnice

Vrstevnice byly určeny za využití lineární interpolace, při které se předpokládá, že spád terénu je mezi podrobnými body p_i , mezi nimiž se provádí interpolace, konstantní.

Mějme trojúhelník t_j , tvořený hranami e_1, e_2, e_3 a rovinu vrstevnice ρ o dané výšce. Vztah hrany trojúhelníku a roviny vrstevnice:

1. $(z - z_i) * (z - z_{i+1}) < 0 \longrightarrow e_i \cap \rho$
2. $(z - z_i) * (z - z_{i+1}) > 0 \longrightarrow e_i \notin \rho$
3. $(z - z_i) * (z - z_{i+1}) = 0 \longrightarrow e_i \in \rho$

Pokud $e_1, e_2, e_3 \in \rho$, jedná se o trojúhelník náležící rovině ρ a není nutné vrstevnici pro tento trojúhelník řešit.

Jestliže $e_i \cap \rho$, je vypočten průsečík hrany $e_i = (p_1, p_2)$ a roviny vrstevnice ρ o výšce z :

$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1,$$
$$y = \frac{(y_2 - y_1)}{(z_2 - z_1)}(z - z_1) + y_1.$$

3.3 Sklon

Sklon je úhel φ mezi svislicí n a normálou trojúhelníku n_t . Rovina trojúhelníku t_j je určena vektory u, v .

$$\begin{aligned}n &= (0, 0, 1) \\n_t &= \vec{u} \times \vec{v} \\ \varphi &= \arccos\left(\frac{n_t n}{|n_t| |n|}\right)\end{aligned}$$

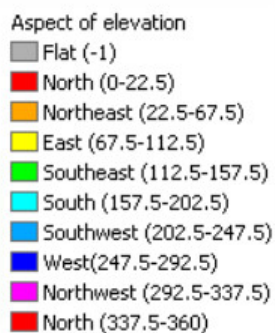
3.4 Expozice

Expozice je orientace trojúhelníku vůči světovým stranám.

$$A = \arctan 2\left(\frac{n_x}{n_y}\right);$$

kde n_x, n_y jsou vektorové součiny u a v .

Expozice je vizualizována pomocí barevného spektra, barvy byly vybrány stejně jako v SW ArcMAP - ESRI:



Obrázek 1: Vizualizace expozice

4 Popis algoritmů

4.1 Delaunayova triangulace

Triangulace byla realizována metodou inkrementální konstrukce, body jsou tedy do triangulace přidávány postupně a to tak, aby vybraný bod ležel v levé polorovině od orientované hrany, poloměr opsané kružnice byl minimální a zároveň jsou preferovány body, jejichž střed opsané kružnice leží v pravé polorovině. Pokud žádný bod těmito kritériím nevyhovuje, je orientace hrany obrácena a bod je vybírán znovu. Jakmile je bod nalezen, jsou k němu vytvořeny orientované hrany a vše je uloženo do triangulace.

Pro "manipulaci" s hranami se používá struktura Active Edges List *AEL*, do ní jsou ukládány hrany, ke kterým je třeba nalézt třetí bod a vytvořit trojúhelník. Jakmile je *AEL* prázdná, algoritmus končí.

4.1.1 Implementace metody

1. $p_1 = rand(P), ||p_2 - p_1|| = min \dots$ náhodný a nejbližší bod
2. Vytvoř hranu $e = (p_1, p_2)$
3. Inicializuj: $p_{min} = argmin_{\forall p_i \in \sigma_L(e)} r'(k_i), k_i = (a, b, p_i), e = (a, b)$
4. Pokud $\nexists p_{min}$, prohod' orientaci $e \leftarrow (b, a)$. Jdi na 3)
5. $e_2 = (p_1, p_{min}), e_3 = (p_{min}, p_1) \dots$ zbývající hrany trojúhelníku
6. $AEL \leftarrow e, AEL \leftarrow e_2, AEL \leftarrow e_3$
7. $DT \leftarrow e, DT \leftarrow e_2, DT \leftarrow e_3$
8. while *AEL* not empty:
9. $AEL \rightarrow e, e = (p_1, p_2) \dots$ vezme první hranu z *AEL*
10. $e = (p_2, p_1) \dots$ prohodí její orientaci
11. $p_{min} = argmin_{\forall p_i \in \sigma_L(e)} r'(k_i), k_i = (a, b, p_i), e = (a, b)$
12. if $\exists p_{min}$:
13. $e_2 = (p_1, p_{min}), e_2 = (p_{min}, p_1) \dots$ zbývající hrany trojúhelníku
14. $DT \leftarrow e$
15. $add(e_2, AEL, DT), add(e_3, AEL, DT)$

Dílčí algoritmus Add:

1. Vytvoř hranu $e' = (b, a)$
2. if $(e' \in AEL)$
3. $AEL \rightarrow e' \dots$ Odstraň z AEL
4. else:
5. $AEL \leftarrow e \dots$ Přidej do AEL
6. $DT \leftarrow (a, b)$ Přidej do DT

4.1.2 Problematické situace

Pokud je vstupními daty grid, dochází k nejednoznačnosti Delaunay triangulace - na opsané kružnici leží čtyři body. V tom případě nefunguje výpočet vrstevnic ani následných analýz.

5 Vstupní data

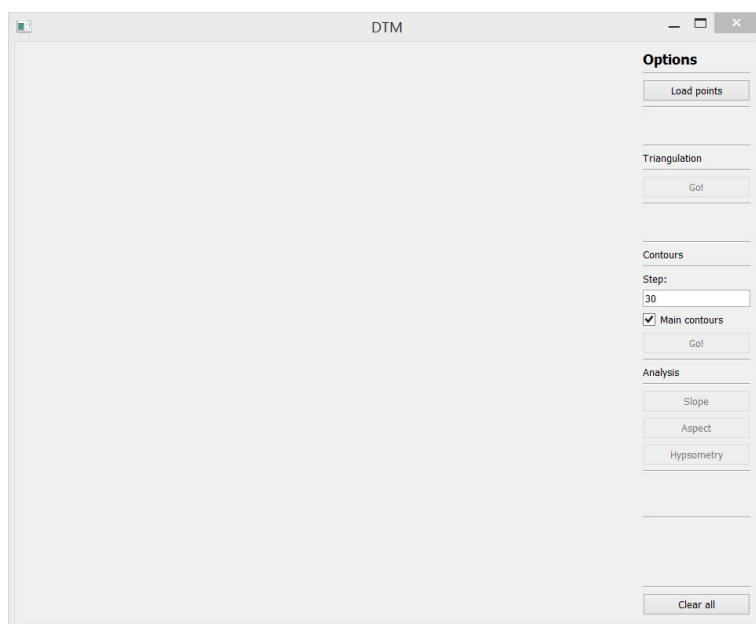
Vstupními daty je textový soubor **.txt* se souřadnicemi bodů ve tvaru $[X,Y,Z]$. Lze jej do aplikace nahrát pomocí tlačítka *Load points*.

Součástí příloh je textový soubor s testovacími daty *testovací_data.txt*.

6 Výstupní data

Výsledky aplikace jsou vizualizovány v kanvasu grafického rozhraní.

7 Ukázka vytvořené aplikace



Obrázek 2: Rozvržení aplikace

Nejprve je třeba nahrát vstupní data pomocí tlačítka *Load points* a následně je možné vypočítat Delaunay triangulaci, teprve poté jsou zpřístupněna i ostatní tlačítka aplikace.

Při výpočtu vrstevnic je třeba zadat krok vrstevnic a (ne)zaškrtnout zvýraznění hlavních vrstevnic.

V sekci analysis je možné vypočíst a vizualizovat sklon a expozici.

Vše je uvedeno do původního stavu (smazány body i vypočtené výsledky) kliknutím na tlačítko Clear.

8 Dokumentace

8.1 Algorithms

V třídě Algorithms jsou staticky implementovány algoritmy počítající Delaunay triangulaci, vrstevnice, analýzu DMT (sklon a expozici) - a barevnou hypsometrii.

- Výčtový typ **TPosition**

- Typ využitý jako návratová hodnota členské metody **getPointLinePosition**.
- **LEFT = 0**
- **RIGHT = 1**
- **ON = 2**

- Metoda **getPointLinePosition**

- Tato metoda slouží k určení polohy bodu vůči přímce. Návratovou hodnotou je výčtový typ **TPosition**.
- Vstup
 - * **QPointF &q** - určovaný bod
 - * **QPointF &a, &b** - body přímky
- Výstup
 - * **LEFT** - bod vlevo od přímky
 - * **RIGHT** - bod vpravo od přímky
 - * **ON** - bod na přímce

- Metoda **getCircleRadius**

- Tato metoda slouží k výpočtu poloměru opsané kružnice **double**.
- Vstup
 - * **QPoint3D &p1, &p2, &p3, &c** - body p_i jsou body, jimž je kružnice opsána, do bodu c jsou ukládány souřadnice středu kružnice.
- Výstup
 - * Vypočtený poloměr kružnice.

- Metoda **getNearestPoint**

- Tato metoda slouží k vyhledání nejbližšího bodu. **int**.
- Vstup
 - * **QPoint3D &p, std::vector<QPoint3D> &points** - od bodu p je hledán nejbližší bod z vektoru bodů *points*.
- Výstup
 - * index nejbližšího bodu.

- Metoda **distance**

- Tato metoda slouží k výpočtu vzdálenosti dvou bodů. **double**.
- Vstup
 - * **QPoint3D &p1, QPoint3D &p2**
- Výstup
 - * vzdálenost dvou bodů.
- Metoda **getDelaunayPoint**
 - Tato metoda slouží k vyhledání bodu, který vyhovuje kritériím Delaunay triangulace. **int**.
 - Vstup
 - * **QPoint3D &s, QPoint3D &e, std::vector<QPoint3D> &points**
 - Výstup
 - * index bodu.
- Metoda **DT**
 - Tato metoda slouží k vytvoření vektoru, v němž jsou hrany Delaunay triangulace. **std::vector<Edge>** .
 - Vstup
 - * **std::vector<QPoint3D> &points**
 - Výstup
 - * vektor hran Delaunay triangulace.
- Metoda **getContourPoint**
 - Tato metoda slouží k vypočtení souřadnic průsečíku vrstevnice a hrany DT. **QPoint3D**.
 - Vstup
 - * **QPoint3D &p1,&p2, double z** z je výška vrstevnice, bod p_1 resp. p_2 je počáteční resp. koncový bod hrany.
 - Výstup
 - * průsečík vrstevnice a trojúhelníku DT.
- Metoda **createContours**
 - Tato metoda slouží k vypočtení vrstevnic. **std::vector<Edge>**.
 - Vstup
 - * **std::vector<Edge> &dt, double z_min, double z_max, double dz, std::vector<double> &contour_heights, std::vector<QPolygonFZ> &hyps**
 - Výstup
 - * vektor hran vrstevnic.

- Metoda **getSlope**
 - Tato metoda slouží k výpočtu sklonu trojúhelníku double.
 - Vstup
 - * **QPoint3D &p1, &p2, &p3** - vrcholy trojúhelníku.
 - Výstup
 - * sklon trojúhelníku [°].
- Metoda **getAspect**
 - Tato metoda slouží k výpočtu expozice trojúhelníku double.
 - Vstup
 - * **QPoint3D &p1, &p2, &p3** - vrcholy trojúhelníku.
 - Výstup
 - * expozice trojúhelníku [°].
- Metoda **analyzeDMT**
 - Tato metoda slouží k vytvoření trojúhelníků a výpočtů jejich sklonu a expozice `std::vector<Triangle>`.
 - Vstup
 - * **std::vector<Edge> &dt** - vektor hran Delaunay triangulace.
 - Výstup
 - * vektor trojúhelníků a jejich sklon a expozice.
- Metoda **sweepLineCH**
 - Tato metoda slouží k výpočtu konvexní obálky pomocí algoritmu Sweep Line. Jejím výstupním typem je **QPolygonF**.
 - Vstup
 - * **std::vector <QPointF> points** - vektor bodů, kolem nichž má být vytvořená konvexní obálka.
 - Výstup
 - * Polygon obsahující konvexní obálku.

8.2 Draw

Třída draw slouží k vykreslení načtených bodů, vypočtené Delaunay triangulace, vrstevnic, sklonu, expozice a hypsometrie. Třída dědí od třídy **QWidget**.

- Členské proměnné
 - **std::vector <QPoint3D> points** vektor načtených bodů
 - **std::vector <Edge> dt** - vektor hran Delaunay triangulace

- **std::vector <Edge> contours** - vektor hran vrstevnic
 - **std::vector <double> contours_heights** - vektor výšek vrstevnic
 - **std::vector <Triangle> dtm** - vektor trojúhelníků Delaunay triangulace
 - **std::vector <QPolygonFZ> hyps** - vektor polygonů hypsometrie
 - **bool draw_main** - nese informaci, zda chce uživatel vykreslit zvýraznění hlavních vrstevnic
 - **int step** - rozestup vrstevnic
 - **bool draw_slope** - nese informaci, má být vykreslen sklon
 - **bool draw_aspect** - nese informaci, má být vykreslena expozice
 - **bool draw_hyps** - nese informaci, má být vykreslena hypsometrie
 - **std::vector <QColor> ctable** - tabulka barev pro hypsometrii
- public metody
 - Metoda **paintEvent**
 - * Tato metoda slouží k vykreslení načtených bodů, vypočtené Delaunay triangulace, vrstevnic, sklonu, expozice a hypsometrie. Metoda se volá pomocí metody **repaint()**. Návrátovým typem je **void**.
 - * Vstup
 - **QPaintEvent *e**
 - Metoda **clearDT**
 - * Tato metoda slouží pro vymazání bodů, triangulace, vrstevnic a hypsometrie. Na vstupu není nic a je typu **void**.
 - Metoda **getPoints** Tato metoda slouží k zpřístupnění členské proměnné **points**.
 - Metoda **getDT** Tato metoda slouží k zpřístupnění členské proměnné **dt**.
 - Metoda **setDT** Tato metoda slouží k přiřazení hodnot členské proměnné **dt**.
 - Metoda **setContours** Metoda slouží k přiřazení hodnot členským proměnným **contours**, **sontours_heights**, **step**, **draw_main** a **hyps**.
 - Metoda **setDTM** Tato metoda slouží k přiřazení hodnot členské proměnné **dtm**.
 - Metoda **loadPoints**
 - * Tato metoda slouží pro načtení bodů z textového souboru do vektoru **points** a pro určení minimální a maximální výšky bodů v souboru. Zároveň je zvoleno "měřítko" kanvasu dle velikosti kanvasu a rozsahu vstupních dat. **void**.
 - * Vstup
 - **std::string path, QSizeF &canvas_size, double &min_z, &max_z**
 - Metoda **genAspColor** Metoda slouží k přiřazení hodnot do členské proměnné **ctable** - zde jsou přiřazeny barvy k hodnotám expozice.

8.3 Edge

- členské proměnné **Qpoint3D s, e** - počáteční a koncový bod hrany
- public metody:
 - **getS resp. getE** - metoda slouží k zpřístupnění členských proměnných s resp. e. Návrátovým typem je **QPoint3D**.
 - **switchOr** - metoda slouží ke změně orientace hrany - prohodí počáteční a koncový bod. Návrátovým typem je **void**.

8.4 QPoint3D

Implicitní hodnoty souřadnic: [0, 0, 0]

- členské proměnné **double z** - výška bodu
- public metody:
 - **getZ** - metoda slouží k zpřístupnění členské proměnné z. Návrátovým typem je **double**.
 - **setZ** - metoda slouží k přiřazení hodnoty do členské proměnné z. Návrátovým typem je **void**.

8.5 SortByXAsc, SortByZAsc,

Třídy sloužící jako sortovací kritérium - podle rostoucí souřadnice x resp. souřadnice y (při stejných souřadnicích x resp. y je druhým kritériem druhá souřadnice) a podle rostoucího úhlu mezi body (při stejném úhlu je druhým kritériem vzdálenosti mezi body).

8.6 Triangle

- členské proměnné:
 - QPoint3D p1,p2,p3** - tři vrcholy trojúhelníku
 - double slope, aspect** - sklon a expozice trojúhelníku
- public metody:
 - Metodami je zpřístupněna hodnota členských proměnných.

8.7 Widget

Tato třída slouží ke komunikaci s GUI. Třída dědí od třídy `QWidget`. Všechny její metody slouží jako sloty k signálům z GUI, nemají žádné vstupní hodnoty a jejich návratovým typem je `void`.

- Metoda **`on_del_button_clicked`** - reaguje na zmáčknutí tlačítka pro vypočtení Dalaunay triangulace, volá metodu **`AnalyzeDTM`** z třídy **`Algorithms`**.
- Metoda **`on_clear_button_clicked`** - reaguje na zmáčknutí tlačítka pro vymazání obsahu kreslicího plátna, volá metodu **`clearDT`** z třídy **`Draw`**.
- Metoda **`on_cont_button_clicked`** - reaguje na zmáčknutí tlačítka pro generování vrstevnic bodů, volá metodu **`createContours`** z třídy **`Algorithms`** a **`setContours`** z třídy **`Draw`**.
- Metoda **`on_slope_button_clicked`** - reaguje na zmáčknutí tlačítka pro výpočet sklonu, volá metodu **`setDrawSlope`** z třídy **`Draw`**.
- Metoda **`on_load_points_button_clicked`** - reaguje na zmáčknutí tlačítka pro načtení bodů, volá metodu **`loadPoints`** z třídy **`Draw`**.
- Metoda **`on_aspect_button_clicked`** - reaguje na zmáčknutí tlačítka pro výpočet expozice, volá metodu **`setDrawAspect`** z třídy **`Draw`**.
- Metoda **`on_hyps_button_clicked`** - reaguje na zmáčknutí tlačítka pro vykreslení hypsometrie, volá metodu **`drawHypsometry`** z třídy **`Draw`**.

9 Přílohy

- Příloha č.1: Testovacídata - *testovací_data.txt*.

10 Závěr

10.1 Zhodnocení

10.2 Návrhy na vylepšení

11 Zdroje

1. BAYER, Tomáš. 2D triangulace, DMT [online][cit. 1.12.2018].
Dostupné z: <https://web.natur.cuni.cz/bayertom/images/courses/Adk/adk5.pdf>
2. BAYER, Tomáš. 2D triangulace, DMT [online][cit. 30.11.2018].
Dostupné z: <https://web.natur.cuni.cz/bayertom/images/courses/Adk/adkcv3.pdf>
3. Tools: Aspect. ArcMap—ArcGIS Desktop [online]. Environmental Systems Research Institute, 2016 [cit. 2018-12-02].
Dostupné z: <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/how-aspect-works.htm>