

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

IDENTIFIKÁCIA STALAKTITOV V 3D MODELI
JASKYNE

Študentská vedecká konferencia 2018

**UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA**

**IDENTIFIKÁCIA STALAKTITOV V 3D MODELI
JASKYNE**

Študentská vedecká konferencia 2018

Študiálny program:	Informatika
Študiálny odbor:	9.2.1. Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci práce:	RNDr. Ondrej Krídlo, PhD.
Konzultant:	doc. Mgr. Michal Gallay, PhD.

Abstrakt

Hlavnou výzvou tejto práce bola analýza, identifikácia a následná separácia jaskynných útvarov – stalaktitov v 3D modeli stropu jaskyne, ktorý bol vytvorený z mračna bodov naskenovaných laserovým skenerom odborníkmi z Ústavu geografie. Ako vo väčšine oblastí dnešného sveta aj pri analyzovaní spomínaného 3D modelu je potrebné vysporiadať sa s veľkým množstvom dát.

Prvý implementovaný algoritmus pochádza z článku [2] a ide o vyhľadávanie lokálnych miním v 3D modeli. Tento algoritmus nachádza tie vrcholy 3D modelu, ktoré by teoreticky mali byť minimami hľadaných stalaktitov.

Ďalším z cieľov práce bolo navrhnúť postup pre separáciu jednotlivých stalaktitov. Navrhli sme preto algoritmus, ktorý ale dokáže vyseparovať stalaktit len po jeho prvé rozvetvenie. Ďalší algoritmus oddelí z 3D modelu tú časť, o ktorej predpokladáme, že patrí stropu jaskyne, takže zostávajúca časť 3D modelu by mala predstavovať stalaktity na zadanom strope. Spojením myšlienok z viacerých algoritmov sme nakoniec navrhli algoritmus, ktorý sa snaží čo najpresnejšie určiť časť 3D modelu, ktorá obsahuje len stalaktity. Všetky navrhnuté algoritmy sme implementovali v jazyku Java a testovali na poskytnutom 3D modeli. Pri každom algoritme sme následne vyhodnocovali jeho výstupy.

Kľúčové slová: *3D model, graf, stalaktit, lokálne minimum, súbor PLY.*

Abstract

The main challenge for this thesis was the analysis, identification and ensuing separation of cave's structures - stalactites in a 3D model of ceiling of this cave, which was created from point cloud created by laser scanner by experts from the Institute of Geography. As in most areas of today's world, even in the analysis of the above-mentioned 3D model, it is necessary to deal with a great abundance of data.

The very first implemented algorithm was from the article [2] and deals with searching of local minima in the 3D model. This algorithm finds those vertices of the 3D model, which, in theory, would be minimums of sought stalactites.

Another aim of the thesis was to propose a procedure for the separation of individual stalactites. Therefore, we have proposed an algorithm that can only separate the stalactite after its first branching. Another algorithm separates from the 3D model these part, that we suppose it belongs to the ceiling of the cave, so that the remaining part of the 3D model should represent stalactites at the assigned ceiling. By combining ideas from multiple algorithms, we have finally designed an algorithm, that attempts to determine as precisely as possible this part of the 3D model, that contains only stalactites. All proposed algorithms were implemented in Java language and tested on the provided 3D model. For each algorithm, we subsequently evaluated its given outputs.

Keywords: *3D model, graph, stalactite, local minima, PLY file.*

Obsah

Úvod	5
1 Vysvetlenie základných pojmov	6
1.1 3D model a jeho uloženie v súbore formátu PLY	6
2 Súčasný stav problematiky	7
2.1 Lokálne minimá v 3D modeli	7
2.2 Identifikácia stalagmitov v 3D modeli	7
3 Návrh, implementácia a vyhodnotenie algoritmov	8
3.1 Lokálne minimá v 3D modeli podľa článku	9
3.2 Vyseparovanie stalaktitu po prvé rozvetvenie	10
3.3 Oddelenie stropu od stalaktitov	12
3.4 Lokálne minimá, ktoré nepatria stropu	15
3.5 Spojenie predchádzajúcich prístupov	16
3.6 Objem stalaktitov	17
Záver	19
Zoznam použitej literatúry	20

Úvod

Rozsiahlosť jaskýň a podmienky v nich sťažujú ich skúmanie. Jedným zo spôsobov ako skúmať jaskynné útvary mimo prostredia jaskyne je napríklad naskenovanie jaskyne laserovým skenerom a z týchto naskenovaných dát vyrobiť 3D model, ktorý je možné ďalej analyzovať.

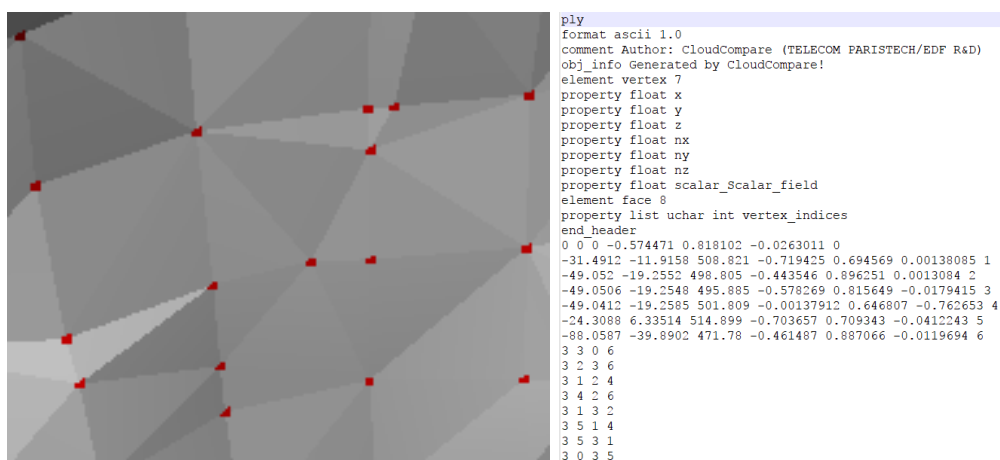
V našej práci sa zaoberáme algoritmami na analýzu, identifikáciu a následnú separáciu jaskynných útvarov – stalaktitov v 3D modeli stropu jaskyne. Na začiatku práce je krátke oboznámenie so súčasným stavom tejto problematiky, kde spomíname dva články [2] a [1], v ktorých sa venovali podobnej problematike.

V nasledujúcej časti sa venujeme jednotlivým algoritmom, ktoré sme navrhli a po implementácii testovali na poskytnutom 3D modeli a vyhodnocovali sme výstupy. Prvý implementovaný algoritmus pochádza z článku [2] a ide o vyhľadávanie lokálnych miním v 3D modeli. Tento algoritmus nachádza tie vrcholy 3D modelu, ktoré by teoreticky mali byť minimami hľadaných stalaktitov. Druhým implementovaným algoritmom je algoritmus, ktorý sme navrhli na vyseparovanie stalaktitu z 3D modelu. Tento algoritmus ale dokáže vyseparovať stalaktit len po jeho prvé rozvetvenie. Ďalším navrhnutým algoritmom, ktorý popisujeme v tejto práci je algoritmus na oddelenie tej časti 3D modelu, o ktorej predpokladáme, že patrí stropu jaskyne, takže zostávajúca časť 3D modelu by mala predstavovať stalaktity na zadanom strope. Ďalej popisujeme návrh algoritmu, ktorý využíva informáciu o predpokladanom strope jaskyne zo spomínaného algoritmu na vyfiltrovanie tých lokálnych miním celého 3D modelu, ktoré zároveň patria stropu. Teda ako výstup vráti vrcholy lokálnych miním, ktoré neboli označené, že patria stropu, keďže je podľa nás pochopiteľné, že vrcholy, ktoré boli označené, že patria stropu by nemali predstavovať minimá niektorého stalaktitu. Predposledná podkapitola tretej kapitoly sa venuje návrhu algoritmu na vyseparovanie výsledných stalaktitov z 3D modelu. Tento algoritmus je spojením prístupov popisovaných v predošlej časti tretej kapitoly. Posledným popisovaným je návrh algoritmu na vypočítanie objemu spomínaných výsledných stalaktitov.

Pri každom spomínanom algoritme sme následne vyhodnocovali výstupy, ktoré sme dostali po implementácii.

1 Vysvetlenie základných pojmov

1.1 3D model a jeho uloženie v súbore formátu PLY



Obr. 1: Vľavo zobrazená zväčšená časť 3D modelu. Vpravo štruktúra súboru PLY.

3D model, s ktorým pracujeme je reprezentovaný ako graf, množina vrcholov a trojuholníkových stien medzi týmito vrcholmi. Táto štruktúra sa dá uložiť do súboru formátu PLY, ktorý v rámci práce pre tento účel používame. V tomto súbore je táto štruktúra uložená tak, že sú tu zapísané informácie o vrcholoch a stenách grafu. Ako je vidieť na obrázku 1 za hlavičkou súboru kde sú uložené informácie o poradí stĺpcov z jednotlivými parametrami vrcholov a tiež informácie o počte vrcholov a stien nasledujú riadky s hodnotami jednotlivých vrcholov a stien grafu. Pri riadkoch so zápisom vrcholov je v každom riadku zapísaná informácia o jednom uloženom vrchole. Po riadkoch s vrcholmi nasledujú riadky so stenami daného 3D modelu. Tie sú zapísané tak, že prvá hodnota charakterizuje počet vrcholov, ktoré ju tvoria a nasledujúce hodnoty sú indexy týchto vrcholov v predchádzajúcom zozname vrcholov.

2 Súčasný stav problematiky

2.1 Lokálne minimá v 3D modeli

V článku [2] je popísaný algoritmus na hľadanie lokálnych miním v 3D modeli. Tento algoritmus využíva to, že 3D model je vlastne graf a teda vieme zistiť susedov vybraného vrcholu. Za lokálne minimum tento algoritmus označí vrchol, ktorý má menšiu súradnicu z (zodpovedajúca vertikálnej osi) než všetci jeho susedia a súradnica z normálového vektora v tomto vrchole je záporná. Tieto lokálne minimá sú potom považované za konce stalaktitov.

Algoritmus 1 Nájdenie lokálnych miním 3D modelu.

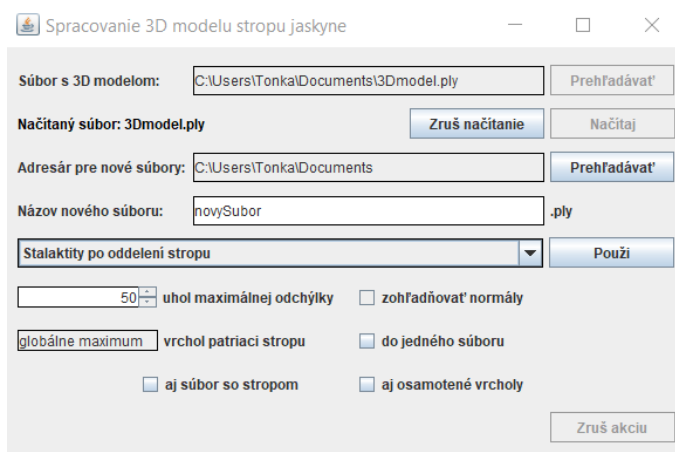
```
1: procedure LOKALNEMINIMA( $G = (V, E)$ )
2:    $L_m \leftarrow$  prázdny zoznam
3:   for  $v \in V$  do
4:      $N(v) \leftarrow$  zoznam susedov vrcholu  $v$ 
5:     if  $(z_v \leq z_{v_i}) \wedge (n_z < 0)$  pre všetky  $v_i \in N(v)$  then
6:        $L_m \leftarrow v$ 
7:   return  $L_m$  ▷ zoznam lokálnych miním 3D modelu
```

2.2 Identifikácia stalagmitov v 3D modeli

Iný prístup je popisovný v článku [1] kde sa zaoberali identifikáciou stalagmitov pomocou porovnávania súradníc stredov elíps, ktoré vznikli z 3D modelu po jeho rozčlenení vodorovnými plochami. Za jeden stalagmit sú potom považované tie elipsy, ktorých súradnice stredov sú od seba bližšie než zadaná vzdialenosť. Tento postup sme sa rozhodli v našej práci nepoužiť.

3 Návrh, implementácia a vyhodnotenie algoritmov

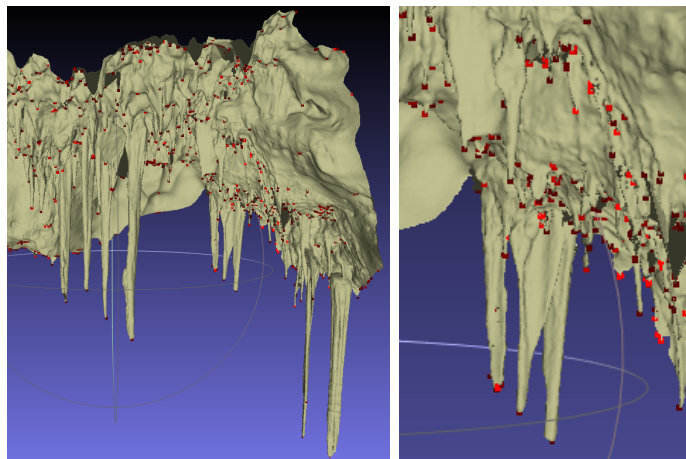
Všetky navrhnuté algoritmy sme implementovali v jazyku Java a testovali na poskytnutom 3D modeli. 3D model, s ktorým pracujeme je uložený v súbore formátu PLY a výstupy implementovaných algoritmov taktiež zapisujeme do výstupného súboru / súborov PLY. 3D model uložený v súbore PLY si zobrazujeme pomocou programu MeshLab alebo CloudCompare. Na to, aby sa dali implementované algoritmy používať sme naprogramovali jednoduché používateľské rozhranie využitím Swing (Java).



Obr. 2: Používateľské rozhranie využitím Swing (Java).

3.1 Lokálne minimá v 3D modeli podľa článku

Ako východiskový sme si zvolili algoritmus na nájdenie lokálnych miním popísaný v článku [2].

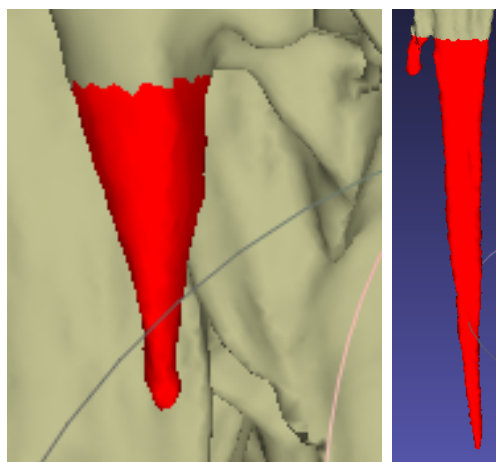


Obr. 3: Lokálne minimá celého 3D modelu.

Po implementácii tohto algoritmu a jeho testovaní na poskytnutom 3D modeli sme prišli k nasledujúcim zisteniam.

1. Tento algoritmus pracuje korektne, len ak vieme pre každý vrchol 3D modelu súradnice normálového vektora v tomto bode. 3D model nemá vo vrcholoch súradnice normálového vektora napríklad preto, lebo scanner ktorý zaznamenával mračno bodov, z ktorého bol vyrobený 3D model tieto údaje nezaznamenával. Dodatočné dorátanie normál pre vrcholy pomocou programu, v ktorom bol 3D model vytvorený neprinieslo dobré výsledky. V takomto 3D modeli nebolo nájdených veľa lokálnych miním, ktoré isto boli minimom nejakého stalaktitu. Pre takéto 3D modeli treba zohľadňovanie normál pri algoritme vynechať, lebo inak ich nájde oveľa menej než ich má reálne nájsť.
2. Na relatívne rovnom strope jaskyne môže nájsť lokálne minimá, ktoré možno vôbec nie sú minimami stalaktitov, ale môžu byť spôsobené len miernym zvlnením stropu alebo štruktúrou 3D modelu, ktorý sa skladá z trojuholníkových stien, takže aj napohľad rovný povrch sa skladá z nerovností, ktoré môžu byť označené za minimá stalaktitov.
3. Viac lokálnych miním môže prislúchať jednému stalaktitu, ktorý má na sebe nerovnosti (mierne sa rozvetvuje) , napríklad kvôli nedokonalosti 3D modelu.

3.2 Vyseparovanie stalaktitu po prvé rozvetvenie



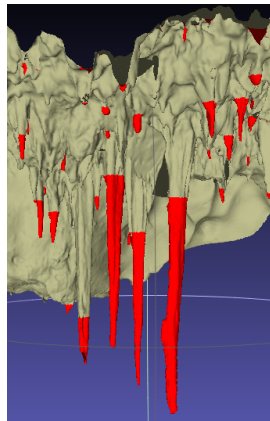
Obr. 4: Vpravo ohyb, na ktorom sa algoritmus zastavil. Vľavo rozvetvenie stalaktitu na dve časti medzi ktorými je ohyb.

Keďže vieme nájsť minimá stalaktitovov podľa algoritmu na hľadanie lokálnych miním tak sme skúsili navrhnúť algoritmus na ich vyseparovanie z 3D modelu. V tomto algoritme sme si zadefinovali stalaktit ako časť 3D modelu, ktorá začína v nejakom lokálnom minime a končí vo výške, kde sa nájde na tomto stalaktite prvé rozvetvenie.

Rozvetvenie tento algoritmus nájde tak, že si do zoznamu vrcholov usporiadaného od najmensej súradnice z najprv pridá vrchol lokálneho minima. A potom táto metóda do toho usporiadaného zoznamu pridáva susedov vrcholu, ktorý bol vybratý zo začiatku zoznamu (má najmenšiu súradnicu z), ktorý v tom zozname ešte predtým neboli. Celé to skončí keď sa nájde prvý ohyb (obrázok 4) – vybratý vrchol bude mať suseda s nižšou súradnicou z než má on a zároveň tento vrchol ešte nebol v zozname. Ak by to nebol ohyb tak by táto situácia nemohla nastať, lebo ten sused by bol už spracovaný predtým keďže ich spracúvame od najmensej súradnice z a zaznamenávame si, že ktorý vrchol už v zozname bol.

Algoritmus 2 Nájdenie stalaktitu po prvé rozvetvenie.

```
1:                                     ▷  $v_m$  je vrchol minima stalaktitu
2: procedure STALAKTITPOPRVEROZVETVENIE( $G = (V, E), v_m$ )
3:    $L_{\text{stalaktit}} \leftarrow$  prázdny zoznam                                     ▷ nájdené vrcholy stalaktitu
4:    $Rad \leftarrow$  prázdny zoznam       ▷ vrcholy usporiadané od najmensej súradnice Z
5:    $Bol \leftarrow$  prázdny zoznam       ▷ vrcholy, ktoré už boli v rade
6:    $NastalOhyb \leftarrow$  false
7:    $Rad \leftarrow v_m$ 
8:    $Bol \leftarrow v_m$ 
9:   while ( $NastalOhyb = \text{false}$ )  $\wedge$  ( $Rad \neq \emptyset$ ) do
10:     $v \leftarrow Rad$                                                          ▷ vyberieme z radu prvý vrchol
11:     $L_{\text{stalaktit}} \leftarrow v$ 
12:     $N(v) \leftarrow$  zoznam susedov vrcholu  $v$ 
13:    for  $s \in N(v)$  do
14:      if  $NastalOhyb = \text{true}$  then
15:        break
16:      if  $s \notin Bol$  then
17:         $Rad \leftarrow s$ 
18:         $Bol \leftarrow s$ 
19:        if  $z_s < z_v$  then                                                         ▷ porovnanie súradníc Z
20:           $NastalOhyb \leftarrow \text{true}$ 
21:  return  $L_{\text{stalaktit}}$                                                          ▷ zoznam vrcholov stalaktitu
```



Obr. 5: Stalaktity po prvé rozvetvenie.

Nevýhodu tohto prístupu, na ktorú sme prišli po implementácii a testovaní na 3D modeli je, že aj malé rozvetvenie spôsobí zastavenie algoritmu, čo je pochopiteľné na základe jeho návrhu (napr. kvôli nedostatkom 3D modelu).

3.3 Oddelenie stropu od stalaktitov

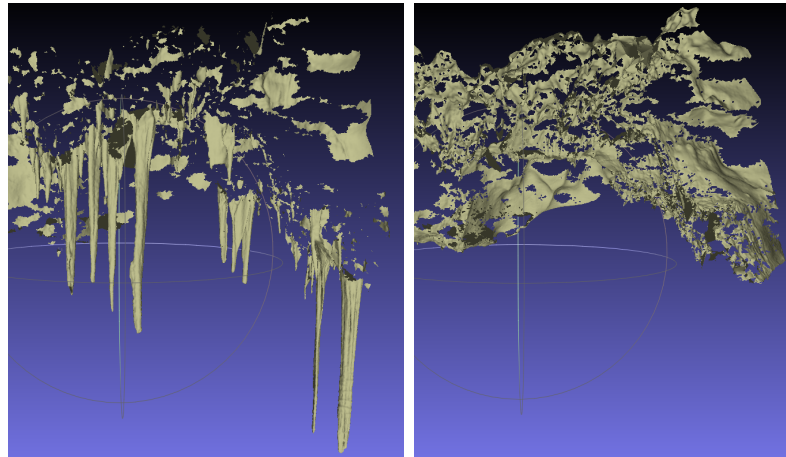
Ako iný prístup nám napadlo oddeliť z 3D modelu tú časť, ktorá by mala predstavovať strop, takže to čo ostalo by teoreticky mali byť stalaktity.

Za strop v tomto algoritme považujeme ucelenú plochu stien (trojuholníkov), kde každá má odchýlku od vodorovnej roviny menšiu než uhol zadaný v parametri metódy a obsahuje vrchol, ktorý predpokladáme, že patrí stropu. Takže vráti zoznam vrcholov 3D modelu, ktoré podľa tohto algoritmu patria stropu.

Metóda pracuje nasledovne. Uchováva si rad vrcholov, ktoré patria doteraz nájdenému stropu a trojuholníky, ktoré ho obsahujú ešte všetky neboli vyhodnotené či patria alebo nepatria stropu. Do tohto zoznamu sa na začiatku vloží vrchol stropu zadaný v parametri metódy. Po vybratí vrcholu z radu sa zaznačí, že daný vrchol patrí stropu a pre každú stenu, ktorá ho obsahuje sa vyhodnotí podľa jej sklonu či patrí alebo nepatrí stropu. Ak patrí stropu tak sa všetky jej vrcholy, ktoré ešte neboli v rade do neho pridajú pre spracovanie a ak má väčší sklon než je zadané tak sa s vrcholmi danej steny nič nerobí. Keď sa rad vrcholov stropu pripravených na spracovanie minie tak predpokladáme, že vrcholy, ktoré neboli zaznačené ako strop sú vrcholy tvoriace stalaktity.

Algoritmus 3 Oddelenie stropu od stalaktitov.

```
1:                                     ▷  $v_s$  je vrchol stropu
2:                                     ▷  $u_m$  maximálny uhol stropu
3: procedure ODDELENIESTROPU( $G = (V, E), v_s, u_m$ )
4:    $L_s \leftarrow$  prázdny zoznam                                     ▷ vrcholy stropu
5:    $Rad \leftarrow$  prázdny zoznam
6:    $L_s \leftarrow v_s$ 
7:    $Rad \leftarrow v_s$ 
8:   while  $Rad \neq \emptyset$  do
9:      $v \leftarrow Rad$                                              ▷ vyberieme vrchol z radu
10:     $L_{steny} \leftarrow$  zoznam stien obsahujúcich  $v$ 
11:    for  $s \in L_{steny}$  do
12:      if  $v_i \in L_s$  pre všetky  $v_i \in s$  then
13:        continue
14:       $uhol \leftarrow$  uhol odchýlky  $s$  od vodorovnej roviny
15:      if  $uhol \leq u_m$  then                                     ▷ test či stena patrí stropu
16:        for  $v_i \in s$  do
17:          if  $v_i \notin L_s$  then
18:             $L_s \leftarrow v_i$                                      ▷ zaznačíme si vrcholy vyhovujúcej steny
19:             $Rad \leftarrow v_i$ 
20:  return  $L_s$                                                      ▷ zoznam vrcholov stropu
```



Obr. 6: Vľavo stalaktity čo ostali po oddelení stropu a vpravo strop čo bol oddelený.

Nevýhody tohto prístupu sú nasledovné:

1. Zadávanie maximálneho uhla stropu. Pri malom zadanom uhle, sa môže stať, že niektoré časti stropu s väčším sklonom nemusia byť nájdené. Prí veľký uhol by zase mohol označiť za strop veľa stalaktitov.
2. Ďalším nedostatkom tohto prístupu je aj to, že treba zadať vrchol 3D modelu, ktorý predpokladáme, že patrí stropu. Ako tento štartovací vrchol sme sa rozhodli použiť vrchol globálneho maxima 3D modelu. Použitie globálneho maxima má výhodu v tom, že ho vieme pomerne jednoznačne nájsť a teoreticky by isto nemalo patriť stalaktitu za predpokladu, že načítaný 3D model obsahuje nejaký strop. Nevýhodou použitia globálneho maxima na tento účel je, že strop bude vyhľadávať, len od takého uhla ako je uhol sklonu stien, ktoré bezprostredne susedia z týmto vrcholom, lebo ak bude zadaný uhol menší tak sa tento algoritmus zastaví priskoro a nájde len veľmi malú časť stropu, lebo akoby sa nebude vedieť dostať dole zo „strmého kopca“ na ktorého vrchole je globálne maximum aj keď pod týmto akoby „strmým kopcom“ môže byť časť stropu z malým sklonom, ktorú by tento algoritmus inak našiel, ak by mal nejaký vrchol, ktorý tomuto úseku stropu patrí.

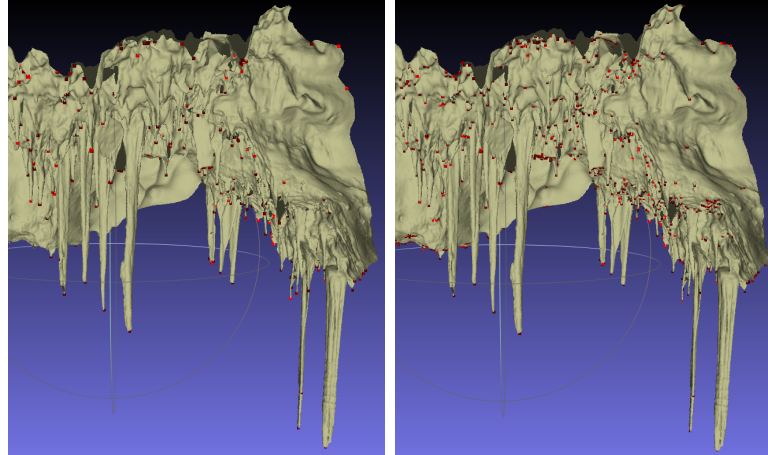
3.4 Lokálne minimá, ktoré nepatria stropu

Keďže máme spomínaný algoritmus na oddelenie stropu od stalaktitov tak nám napadlo, či by sa ním nedal vylepšiť algoritmus na nájdenie lokálnych miním tak aby nájdené lokálne minimá s väčšou pravdepodobnosťou patrili nejakému minimu stalaktitu a nie napríklad miernej nerovnosti na strope jaskyne. Hlavnou ideou je, že vrcholy stropu by zároveň nemali byť minimom niektorého stalaktitu.

Tento algoritmus zoberieme lokálne minimá pôvodného 3D modelu aj zo stropom a spomedzi nich odstráni tie, ktoré boli označené algoritmom na oddelenie stropu ako vrcholy stropu.

Algoritmus 4 Nájdenie lokálnych miním, ktoré nepatria stropu.

```
1: ▷  $L_{\text{strop}}$  vrcholy stropu  
2: ▷  $L_m$  lokálne minimá celého 3D modelu  
3: procedure MINIMANEPATRIACESTROPU( $L_{\text{strop}}, L_m$ )  
4:    $L_{\text{minima}} \leftarrow$  prázdny zoznam  
5:   for  $v \in L_m$  do  
6:     if  $v \notin L_{\text{strop}}$  then  
7:        $L_{\text{minima}} \leftarrow v$   
8:   return  $L_{\text{minima}}$  ▷ lokálne minimá nepatriace stropu
```



Obr. 7: Vľavo lokálne minimá nepatriace stropu, vpravo lokálne minimá celého 3D modelu.

Po implementácii sa ukázalo, že hľadať lokálne minimá v súbore s predpokladanými stalaktitmi (napr. súbor, kde sú uložené vrcholy, ktoré metóda na oddelenie stropu od stalaktitov neoznačila za vrcholy stropu) nie je dobrý postup, lebo 3D mo-

del v tomto súbore nie je ucelený a ako lokálne minimá to pridá vrcholy z okrajov orezaných oblastí, ktoré ostali akoby vrúbkované kvôli okrajovým trojuholníkom.

Pri spustení tohto algoritmu na 3D modeli so stropom sa ukázalo, že lokálnych miním, ktoré boli nájdené kvôli miernemu zvlneniu stropu je menej ako pri pôvodnom algoritme na hľadanie lokálnych miním.

Nevýhodou je, že tento algoritmus potrebuje dostať ako jeden zo vstupov zoznam vrcholov, ktoré patria stropu. Tento zoznam síce vieme dostať ako výstup spomínaného algoritmu na oddelenie stropu, ale tento algoritmus má viacero nejednoznačných parametrov, ktoré treba vhodne zadať aby bol výsledný nájdený strop čo najpresnejší.

3.5 Spojenie predchádzajúcich prístupov

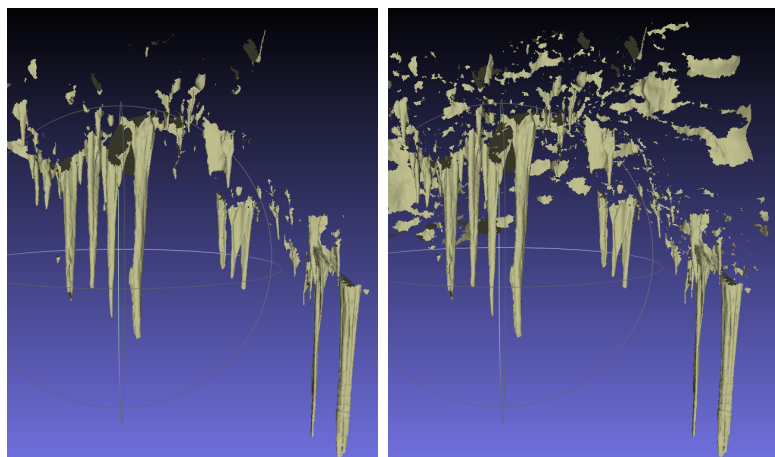
Nakoniec ako najlepší prístup na identifikáciu stalaktitov v 3D modeli sa nám zdá nasledujúci postup, ktorý využíva viacero algoritmov, ktoré som spomínala na predošlých stranách. Ako stalaktity je označená časť 3D modelu, ktorá začína v lokálnych minimách, ktoré nepatria stropu a pridávajú sa tam vždy susedia vybraného vrcholu, ktorý majú väčšiu súradnicu z ako vybraný vrchol ale daný sused nepatrí stropu.

Algoritmus 5 Vyseparovanie stalaktitov.

```

1: ▷  $L_{\text{strop}}$  vrcholy stropu
2: ▷  $L_{\text{minima}}$  lokálne minimá nepatriace stropu
3: procedure VYSEPAROVANIESTALAKTITOV( $G = (V, E)$ ,  $L_{\text{strop}}$ ,  $L_{\text{minima}}$ )
4:    $L_{\text{stalaktity}} \leftarrow$  prázdny zoznam
5:    $Rad \leftarrow$  prázdny zoznam
6:    $Bol \leftarrow$  prázdny zoznam ▷ vrcholy, ktoré už boli v rade
7:    $Rad \leftarrow L_{\text{minima}}$ 
8:    $Bol \leftarrow L_{\text{minima}}$ 
9:   while  $Rad \neq \emptyset$  do
10:     $v \leftarrow Rad$  ▷ vyberieme vrchol z radu
11:    if  $v \in L_{\text{strop}}$  then
12:      continue
13:     $L_{\text{stalaktity}} \leftarrow v$ 
14:     $N(v) \leftarrow$  zoznam susedov vrcholu  $v$ 
15:    for  $s \in N(v)$  do
16:      if  $(z_s \geq z_v) \wedge (s \notin Bol)$  then ▷ susedia s vyššou súradnicou  $Z$ 
17:         $Rad \leftarrow s$ 
18:         $Bol \leftarrow s$ 
19:  return  $L_{\text{stalaktity}}$  ▷ zoznam vrcholov vyseparovaných stalaktitov

```



Obr. 8: Vľavo vyseparované stalaktity, vpravo stalaktity čo ostali po oddelení stropu .

Výhodou oproti algoritmu na oddelenie stropu od stalaktitov je, že tento algoritmus odfiltruje viaceré časti stropu, ktoré neboli nájdené, kvôli ich väčšiemu sklonu. Budú odfiltrované celé ak neobsahujú lokálne minimum.

Nevýhodou sú spomínané parametre algoritmu na oddelenie stropu.

Tieto vyseparované stalaktity sa môžu ako výstup zapísať spolu do jedného súboru ply alebo pomocou algoritmu na hľadanie komponentov súvislosti grafu sa dajú jednotlivé časti 3D modelu uložiť do osobitných súborov.

3.6 Objem stalaktitov

Postup pre vyrátanie objemu sme si zvolili nasledovný:

Vyseparované stalaktity, ktoré sme spomínali v predošlej kapitole si najprv akoby rozkúsujeme na stalaktity bez rozvetvení. Každý stalaktit začína v niektorom lokálnom minime, ktoré nepatrí stropu zoradených od najmenej súradnice z . A z tohto minima si zaznačujeme vrcholy, ktoré patria jednému stalaktitu bez rozvetvení tak, že prechádzame len cez susedov, ktorí majú vyššiu súradnicu z a zaznačujeme si, ktoré vrcholy sme si už pridali do niektorého stalaktitu aby bol každý vrchol len v jednom stalaktite.

Výsledný objem stalaktitov potom vyrátame ako súčet objemov týchto jednotlivých stalaktitov bez rozvetvení. Každému stalaktitu vyrátame jeho objem nasledujúcim algoritmom, ktorý dostane ako vstup vrcholy, ktoré tvoria stalaktit, vrchol minima stalaktitu a výšku úsekov podľa ktorej si rozdelíme stalaktit na množiny vrcholov podľa ich súradnice z . Pre každú množinu vrcholov zistíme jej max a min súradnicu x a y . Z týchto hodnôt súradníc vypočítame obsah elipsy, ktorá by sa mala teoreticky nachádzať na vrchu daného úseku.

Následne vyrátame objem telesa medzi každými dvoma po sebe nasledujúcimi elipsami. Podľa Cavalieriho princípu sa dá rátanie objemu tohto telesa s elipsovými podstavami previesť na vyrátanie objemu zrezaného kužeľa s kruhovými podstavami rovnakých obsahov ako zadané elipsy. Následne účtom objemov týchto telies teoreticky dostaneme objem stalaktitu bez rozvetvení.

Výhodou a zároveň nevýhodou je zistenie približného objemu stalaktitov. Je to nepresné, hlavne kvôli tomu, že je ťažké určiť presné rozmery elipsy na vrchu nejakého úseku, keďže podľa tohto algoritmu sa presnosť rátania objemu so zmenšujúcou sa výškou úseku nezvyšuje.

1. Ak je priveľká výška úseku tak to málo zohľadňuje tvar rátaného stalaktitu.
2. Ak je primalá výška úseku tak vyrátaný objem elipsy môže byť výrazne menší než je v skutočnosti, lebo úsek môže obsahovať primálo bodov aby tam boli zahrnuté vrcholy, ktorých max a min súradnicu x a y je približne taká ako v tomto úseku reálne je.
3. Ak sa vrch stalaktitu zužuje (napr. lebo strop, ktorý bol z okolia oddelený sa zvažoval) tak aj tak dáva, že obsah elipsy zodpovedá vrchu zadaného úseku aj keď v takomto prípade to skôr zodpovedá spodnej časti úseku.

Záver

V práci sme sa venovali hlavne návrhu algoritmov pre analyzovanie 3D modelu jaskyne. Po ich implemetácii a testovaní na poskytnutom 3D modeli sme následne vyhodnotili ich výstupy a pri každom prístupe sme uviedli ich výhody a aj nedostatky.

Za hlavné výhody pri týchto algoritmoch považujeme, ich prínos pri automatickom separovaní stalaktitov zo zadaného 3D modelu. Vyseparované stalaktity pri algoritme, kde sme spojili viaceré navrhnuté prístupy vie pomerne dobre nájsť časť 3D modelu, ktorá by mala zodpovedať hľadaným stalaktitom. Algoritmus na vyrátanie objemu vráti síce približný odhad, pre predstavu o objeme tento údaj nie je zlý, ale je tu ešte dosť priestoru na vylepšovanie tohto prístupu.

Navrhnuté algoritmy majú prirodzene aj svoje nedostatky, ktoré sme bližšie popísali pri jednotlivých algoritmoch. Medzi hlavné z nich patria niektoré parametre (napr. uhol maximálneho uhla stropu), ktoré je potrebné dobre zadať aby bol výstup čo najpresnejší. Na výstup implementovaných algoritmov má prirodzene vplyv aj vstupný 3D model, ktorý môže mať viacero nedostatkov, ktoré môžu nepriaznivo ovplyvniť výstup.

V tejto práci sa dá ďalej pokračovať návrhom ďalších algoritmov a vylepšovaním tých doteraz navrhnutých aby výstupy čo najpresnejšie zodpovedali očakávaniam. Napríklad do budúcnosti by bolo dobré navrhnuť presnejší algoritmus na nájdenie stropu alebo zefektívniť implementáciu navrhnutých algoritmov.

Zoznam použitej literatúry

- [1] JAILLET, S., SADIÉ, B., HAJRI, S., PLOYON, E., AND DELANNOY, J.-J. Une analyse 3d de l'endokarst : applications lasergrammétriques sur l'aven d'orgnac (ardèche, france). *Géomorphology* 442, 4 (2011), 379–394. ISBN: 2-913282-54-4
<http://journals.openedition.org/geomorphologie/9594>.
- [2] SILVESTRE, I., RODRIGUES, J. I., FIGUEIREDO, M., AND VEIGA-PIRES, C. High-resolution digital 3d models of algar do penico chamber: limitations, challenges, and potential. *International Journal of Speleology* 44, 1 (2014), 25–35. Tampa, FL (USA) ISSN: 0392-6672
<http://dx.doi.org/10.5038/1827-806X.44.1.3>.