

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

IDENTIFIKÁCIA STALAKTITOV V 3D MODELI
JASKYNE DOMICA

Bakalárska práca

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA V KOŠICIACH
PRÍRODOVEDECKÁ FAKULTA

IDENTIFIKÁCIA STALAKTITOV V 3D MODELI
JASKYNE DOMICA

Bakalárska práca

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



Univerzita P. J. Šafárika v Košiciach
Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Antónia Matisová
Študijný program: Informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.1. informatika
Typ záverečnej práce: Bakalárska práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Identifikácia stalaktitov v 3D modeli jaskyne Domica

Názov EN: Stalactites identification in Domica cave 3D model

Cieľ:

1. Prieskum základov a doterajších riešení danej problematiky.
2. Implementácia, resp. aplikácia známych prístupov.
3. Testovanie na 3D modeli laserového skenu jaskyne Domica.
4. Vyhodnotenie testovaných prístupov.

Literatúra:

[1] Bella, P., Hlaváčová, I., and Holúbek, P. Zoznam jaskýň v Slovenskej republike. SMOPaJ – SSJ – SSS, Liptovský Mikuláš (2007), 364 s.

[2] Hochmuth, Z. Krasové územia a jaskyne slovenska. Geographia Cassoviensis 210, 2 (2008), 1–209. <https://geo.ics.upjs.sk/index.php/13-geographia-cassoviensis/239-gc2009-3-3>.

[3] Jaillet, S., Sadier, B., Hajri, S., Ployon, E., and Delannoy, J.-J. Une analyse 3d de l'endokarst : applications lasergrammétriques sur l'aven d'orgnac (ardèche, france). G'geomorphology 442, 4 (2011), 379–394. ISBN: 2-913282-54-4 <http://journals.openedition.org/geomorphologie/9594>.

[4] 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>.

Vedúci: RNDr. Ondrej Krídlo, PhD.
Ústav : ÚINF - Ústav informatiky
Riaditeľ ústavu: prof. RNDr. Viliam Geffert, DrSc.

Dátum schválenia: 11.05.2018


Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta
Ústav informatiky

Abstrakt

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

Prvý implementovaný algoritmus pochádzal z článku [4] 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 identifikáciu jednotlivých stalaktitov. Navrhli sme preto algoritmus, ktorý ale dokáže identifikovať 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 Domica cave, which was created from point cloud created by laser scanner by experts from the Institute of Geography University of Pavol Jozef Safarik in Košice. 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 [4] 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 identification of individual stalactites. Therefore, we have proposed an algorithm that can only identify 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	7
1 Vysvetlenie základných pojmov	8
1.1 Stalaktit	8
1.2 3D model a jeho uloženie v súbore formátu PLY	9
2 Súčasný stav problematiky	10
2.1 Lokálne minimá v 3D modeli	10
2.2 Identifikácia stalagmitov v 3D modeli	10
3 Návrh algoritmov	11
3.1 Identifikovanie stalaktitu po prvé rozvetvenie	11
3.2 Oddelenie stropu od stalaktitov	12
3.3 Lokálne minimá, ktoré nepatria stropu	14
3.4 Spojenie predchádzajúcich prístupov	15
3.5 Objem stalaktitov	16
4 Implementácia a vyhodnotenie	17
4.1 Načítanie a výroba súborov formátu PLY	17
4.2 Používateľské rozhranie	18
4.3 Lokálne minimá v 3D modeli podľa článku	19
4.4 Identifikovanie stalaktitu po prvé rozvetvenie	20
4.5 Oddelenie stropu od stalaktitov	20
4.6 Lokálne minimá, ktoré nepatria stropu	22
4.7 Spojenie predchádzajúcich prístupov	23
4.8 Objem stalaktitov	24
Záver	25
Zoznam použitej literatúry	26

Úvod

Rozsiahlosť jaskýň a podmienky v nich sťažujú ich skúmanie. Jedným zo spôsobov ako skúmať jaskynné útvary mimo prostredia jaskyne je napríklad skenovanie 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 [3] a [4], 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 [4] 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 identifikovanie stalaktitu v 3D modeli. Tento algoritmus ale dokáže identifikovať 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 filtrovanie tých lokálnych miním celého 3D modelu, ktoré zároveň patria stropu. Teda výstupom sú vrcholy lokálnych miním, ktoré neboli označené, že patria stropu. Podľa nás je pochopiteľné predpokladať o vrcholoch patriacich stropu, že by nemali predstavovať minimá niektorého stalaktitu. Predposledná podkapitola tretej kapitoly sa venuje návrhu algoritmu na identifikovanie výsledných stalaktitov v 3D modeli. Tento algoritmus je spojením prístupov popisovaných v predošlej časti tretej kapitoly. Posledným popisovaným je návrh algoritmu na vypočítanie objemu spomínaných výsledných stalaktitov.

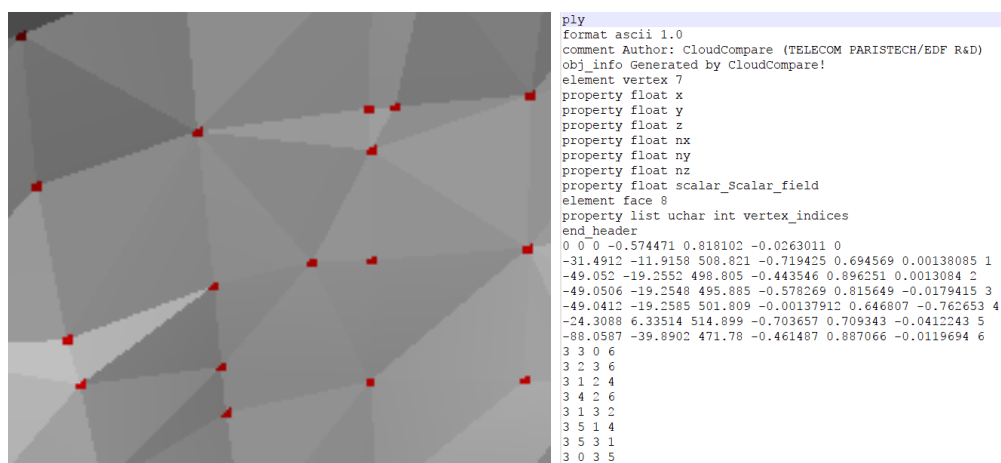
Pri každom spomínanom algoritme sme následne vyhodnocovali výstupy, ktoré sme dostali po implementácii.

1 Vysvetlenie základných pojmov

1.1 Stalaktit

Slovensko je morfológicky pestrá krajina s výrazným zastúpením rozpustných karbonatických hornín (najmä druhohorné vápence a dolomity), v ktorých sa vytvárajú jaskyne (Hochmuth, 2008) [2]. Na Slovensku je vyše 5000 krasových jaskýň, ktorých charakter sumarizuje Bella et al. (2007) [1]. Typickými akumuláčnymi formami jaskynnej výzdoby sú kvaple, ktoré vznikajú vyzrážaním rozpusteného uhličitanu vápenatého. Kvaple visiace z jaskynného stropu sa nazývajú stalaktity, kvaple vyrastajúce z dna jaskyne smerom nahor sú stalagmity. Spojením stalagmitov a stalaktitov vznikajú stĺpy - stalagnáty. Početnosť, rozmery a objem hmoty v kvapľových útvaroch indikuje rýchlosť rozpúšťania a vek kvapľov. Na základe takýchto informácií možno lepšie pochopiť vývoj jaskyne a v konečnom dôsledku krajiny nad ňou. Prakticky je veľmi zdĺhavé spočítavať kvaple v jaskyni, pričom navyše ide o veľmi vzácne útvary chránené zákonom. Z toho dôvodu vzniká potreba navrhnuť automatizované postupy identifikácie kvapľov z dát získaných bezkontaktnými metódami mapovania jaskýň akým je laserové skenovanie. V tejto práci sme sa zamerali iba na stalaktity. Vzorové dáta použité v práci reprezentujú časť stropu v jaskyni Domica, ktorá je verejne sprístupnenou jaskyňou nachádzajúcou sa v juhozápadnej časti Slovenského krasu pri hranici s Maďarskom. Ide o pamiatku evidované v rámci svetového prírodného dedičstva UNESCO.

1.2 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 vpravo 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 [4] 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 (zodpovedajúcu 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.

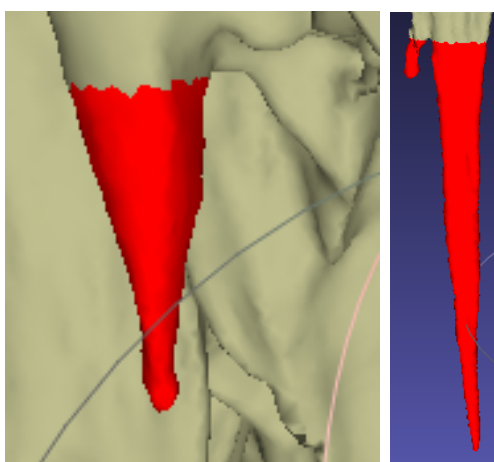
```
1: procedure LOKALNEMINIMA( $G = (V, E)$ )
2:    $L_m \leftarrow$  prázdny zoznam
3:   for  $v \in V$  do
4:      $N(v) \leftarrow$  zoznam susedov vrcholu  $v$ 
5:     if  $(z_v \leq z_{v_i}) \wedge (n_z < 0)$  pre všetky  $v_i \in N(v)$  then
6:        $L_m \leftarrow v$ 
7:   return  $L_m$  ▷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

2.2 Identifikácia stalagmitov v 3D modeli

Iný prístup je popisovaný v článku [3] 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 algoritmov

3.1 Identifikovanie stalaktitu po prvé rozvetvenie



Obr. 2: Vľavo ohyb, na ktorom sa algoritmus zastavil. Vpravo rozvetvenie stalaktitu na dve časti medzi ktorými je ohyb.

Keďže vieme nájsť minimá stalaktitov podľa algoritmu na hľadanie lokálnych miním, tak sme skúsili navrhnúť algoritmus na ich identifikovanie v 3D modeli. 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 najmenej súradnice z najprv pridá vrchol lokálneho minima. A potom tento algoritmus vždy vyberie vrchol zo začiatku zoznamu (má najmenšiu súradnicu z) a do toho usporiadaného zoznamu následne pridá všetkých jeho susedov, ktorý v tom zozname ešte predtým neboli. Celé to skončí, keď sa nájde prvé ohyb (obrázok 2) – vybraný 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 najmenej súradnice z a zaznamenávame si, 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
```

3.2 Oddelenie stropu od stalaktitov

Ako iný prístup sme sa rozhodli 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 zadáný v parametri a obsahuje vrchol, ktorý predpokladáme, že patrí stropu. Takže vráti zoznam vrcholov 3D modelu, ktoré podľa tohto algoritmu patria stropu.

Algoritmus pracuje nasledovne. Uchováva si rad vrcholov, ktoré patria doteraz nájdenému stropu a zároveň trojuholníky, ktoré ich obsahujú ešte všetky neboli vyhodnotené či patria alebo nepatria stropu. Do tohto zoznamu sa na začiatku vloží vrchol stropu zadáný v parametri. 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

```

3.3 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
```

3.4 Spojenie predchádzajúcich prístupov

Nakoniec ako najlepší prístup na identifikáciu stalaktitov v 3D modeli sa nám zdá nasledujúci postup, ktorý využíva viacero algoritmov, ktoré sme spomínali 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 Identifikovanie stalaktitov.

```

1: ▷  $L_{\text{strop}}$  vrcholy stropu
2: ▷  $L_{\text{minima}}$  lokálne minimá nepatriace stropu
3: procedure IDENTIFIKOVANIESTALAKTITOV( $G = (V, E), L_{\text{strop}}, L_{\text{minima}}$ )
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 identifikovaných stalaktitov

```

3.5 Objem stalaktitov

Postup pre vyrátanie objemu sme si zvolili nasledovný:

Identifikované 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 najmensej 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 maximálnu a minimálnu 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 súčtom objemov týchto telies teoreticky dostaneme objem stalaktitu bez rozvetvení.

4 Implementácia a vyhodnotenie

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.

4.1 Načítanie a výroba súborov formátu PLY

Na načítanie a výrobu súborov formátu PLY sme naprogramovali triedu VyrobcuANacitavacSuboruPLY. Informácie o 3D modeli zo vstupného súboru PLY ukladáme do triedy NacitanySubor, s ktorou pracujú implementácie jednotlivých algoritmov popisovaných v predchádzajúcej kapitole.

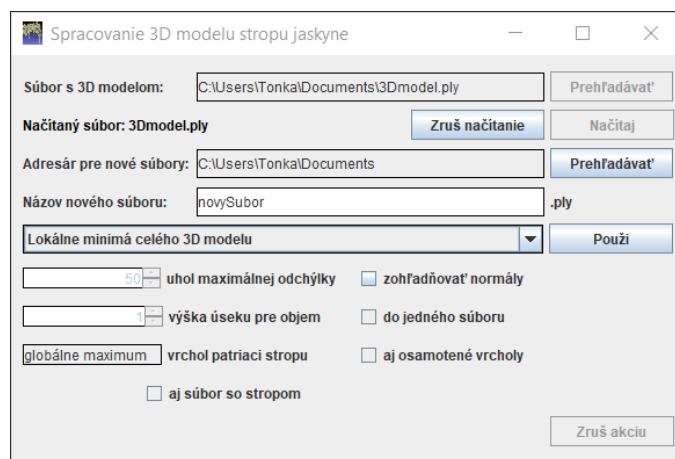
NacitanySubor si uchováva celú hlavičku z pôvodného súboru PLY, aby sme vedeli vyrobiť hlavičky výstupných súborov s rovnakou štruktúrou a 3D model ako pole objektov triedy Vertex. Trieda Vertex si uchováva informácie o jednom načítanom vrchole zo súboru PLY. Okrem jeho súradníc a súradníc jeho normálového vektora si vrchol uchováva aj množinu indexov svojich susedov z pôvodného súboru PLY. Informáciu o stenách 3D modelu sme sa rozhodli uchovávať tiež v triede Vertex ako množinu dvojíc indexov susedov daného vrcholu s ktorými ich vytvára.

Metóda pre vyrobenie výstupného súboru PLY, ktorú sme navrhli potrebuje ako základné parametre hlavičku pôvodného súboru PLY, z ktorého sú zapisované vrcholy, zoznam samotných vrcholov, ktoré majú byť zapísané a názov výstupného súboru. Aby sme pri vyrábaní nových súborov zabránili premazávaniu už existujúcich, tak sa robí kontrola, že či sa v zadanom adresári už nachádza súbor so zadaným názvom. Ak sa zistí, že taký súbor už existuje, tak sa za zadaným názvom postupne inkrementuje číslo, až pokiaľ sa nenájde názov, ktorý ešte neexistuje a s týmto názvom sa vytvorí nový súbor. Pre korektný zápis stien ukladaného 3D modelu sa musí v tejto metóde urobiť mapovanie indexov z pôvodného súboru PLY na nové poradie vrcholov, ktoré je zapisované do výstupného súboru, keďže reprezentácia steny 3D modelu v súbore PLY

sa odkazuje na poradové čísla vrcholov, ktoré ju tvoria v aktuálne zapísanom zozname vrcholov.

4.2 Používateľské rozhranie

Na to, aby sa dali implementované algoritmy používať sme naprogramovali jednoduché používateľské rozhranie využitím Swing (Java).



Obr. 3: Používateľské rozhranie využitím Swing (Java).

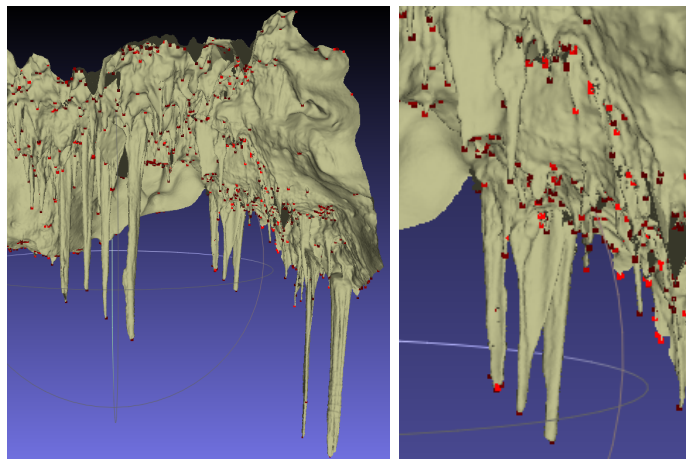
Umožňuje výber a načítanie súboru so vstupným 3D modelom, určenie adresára pre výstupné súbory a výber samotného algoritmu, ktorý sa má na načítanom 3D modeli spustiť. Načítanie vstupného súboru aj výpočet každého algoritmu sa dá zrušiť počas toho ako sa vykonáva príslušným tlačidlom bez nutnosti vypínania celého programu.

Po výbere požadovaného algoritmu sa dajú nastaviť parametre, s ktorými sa má tento algoritmus spustiť a pri niektorých algoritmoch je tiež možné nastaviť, že koľko a akých súborov sa má po dobehnutí vybraného algoritmu vytvoriť. Po stlačení tlačidla pre použitie vybraného algoritmu sa vykoná algoritmus podľa zadaných parametrov a jeho výstup sa zapíše do výstupného súboru / súborov PLY. Ak sa pri algoritme vyberie možnosť, že výstup nechceme do jedného výstupného súboru, tak sa použije algoritmus na hľadanie komponentov súvislosti grafu a každá nájdená súvislá časť výstupného 3D modelu sa zapíše osobitne do nového súboru.

Popis implementácie a vyhodnotenia jednotlivých algoritmov, ktoré je možné použiť je na nasledujúcich stranách.

4.3 Lokálne minimá v 3D modeli podľa článku

Ako východiskový sme si zvolili algoritmus na nájdenie lokálnych miním popísaný v článku [4].

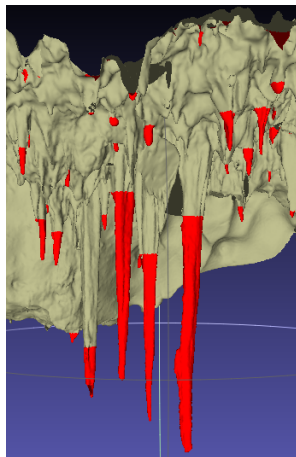


Obr. 4: Lokálne minimá celého 3D modelu.

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 vrchole. 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 rávanie normál pre vrcholy pomocou programu, v ktorom bol 3D model vytvorený neprinieslo dobré výsledky. V takomto 3D modeli nebolo nájdených veľa lokálnych miním, ktoré by určite 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.

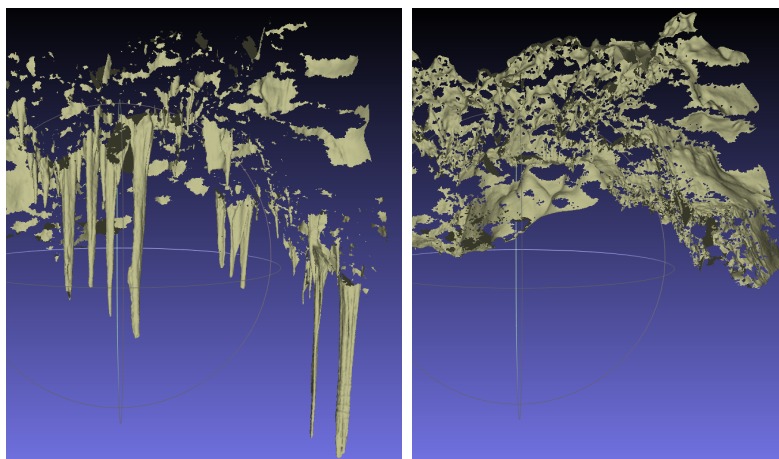
4.4 Identifikovanie stalaktitu po prvé rozvetvenie



Obr. 5: Stalaktity po prvé rozvetvenie.

Nevýhodu tohto prvotného 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).

4.5 Oddelenie stropu od stalaktitov



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

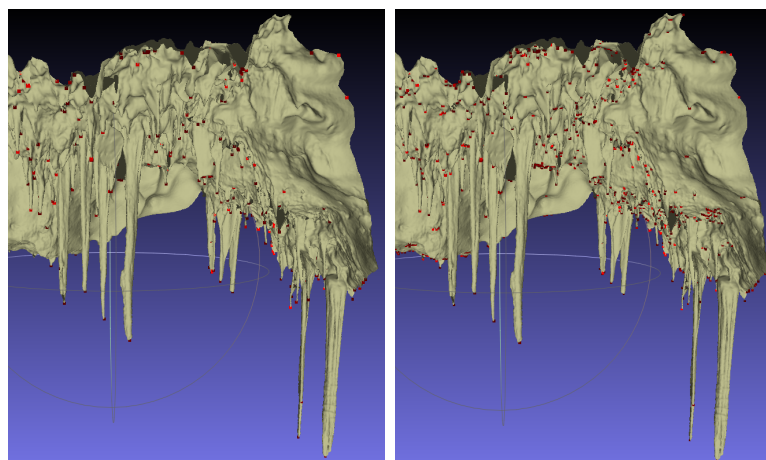
Metóda, ktorá implementuje tento algoritmus, vráti ako výstup pole boolean, ktoré má veľkosť ako počet vrcholov načítaného 3D modelu. V tomto poli je hodnota true na indexoch vrcholov, ktoré algoritmus označil za vrcholy stropu. Číže prejdením

hodnôt v tomto poli vieme nájsť, že ktoré vrcholy pôvodného 3D modelu tento algoritmus označil ako stalaktity po oddelení stropu.

Nedostatky tohto prístupu, ktoré sa ukázali po implementácii a následnom testovaní sú nasledovné:

1. Na to, aby tento algoritmus vrátil čo najpresnejšie výsledky, treba vhodne zadať parameter 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. V tomto algoritme treba zadať vrchol 3D modelu, o ktorom 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 určite 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 s 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 s malým sklonom, ktorú by tento algoritmus inak našiel, ak by mal nejaký vrchol, ktorý tomuto úseku stropu patrí.

4.6 Lokálne minimá, ktoré nepatria stropu



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

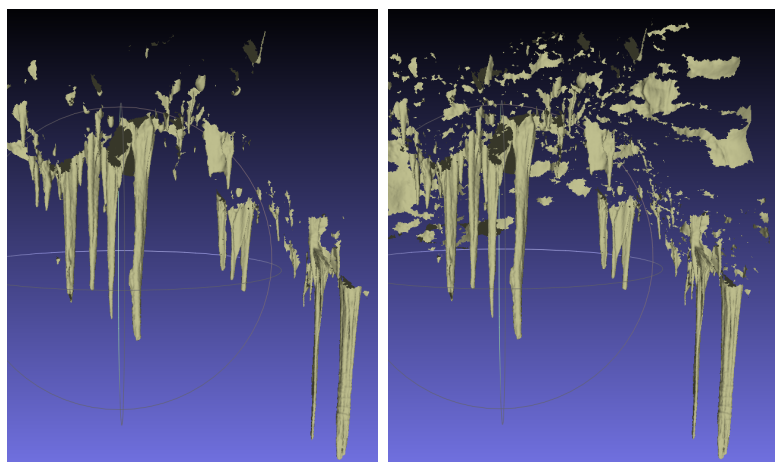
Implementácia tohto algoritmu pracuje s poľami typu boolean, ktoré majú dĺžku ako počet vrcholov načítaného 3D modelu. Ako vstup dostane lokálne minimá celého 3D modelu, ktoré sú v poli boolean zaznačené hodnotou true na ich indexoch a vrcholy patriace stropu dostane v poli boolean tiež zaznačené hodnotou true. Následne pomocou jedného prechodu cez tieto dve polia vyrobí táto metóda výstupné pole boolean, v ktorom je hodnota true na indexoch vrcholov, ktoré patria lokálnym minimám, ktoré nepatria stropu.

Po implementácii a 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.

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

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.

4.7 Spojenie predchádzajúcich prístupov



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

Tieto identifikované 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.

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 je už spomínané zadávanie parametrov algoritmu na oddelenie stropu.

4.8 Objem stalaktitov

Pri implementácii algoritmu na rátanie objemu stalaktitu bez rozvetvení zohľadňujeme aj reálnu výšku tohto stalaktitu, ak vrch najvrchnejšieho úseku prekročí jeho výšku (dôsledkom je, že od hodnoty výšky úseku, ktorá zodpovedá dĺžke najdlhšieho stalaktitu sa so zvyšujúcou výškou úseku už objem nebude meniť). Taktiež riešime situáciu, keď je niektorá množina vrcholov úseku prázdna. V takomto prípade ju vynecháme a vyrátame objem telesa medzi dvoma elipsami úsekov pred a za vynechaným úsekom a výšku tohto telesa zadáme ako dvojnásobok výšky úseku (Prirodzene za predpokladu, že vynechaný úsek nebol predposledný odhora. V takom prípade sa zohľadní výška len po reálnu výšku vrchu stalaktitu).

Výhodou a zároveň nevýhodou tohto prístupu 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 vrcholov, aby tam boli zahrnuté vrcholy, ktorých maximálna a minimálna súradnica 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 stropu jaskyne. Po ich implementácii 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 identifikovaní stalaktitov v zadanom 3D modeli. Algoritmus na identifikovanie stalaktitov, kde sme spojili viaceré navrhnuté prístupy vie pomerne dobre nájsť časť 3D modelu, ktorá by mala zodpovedať hľadaným stalaktitom. Algoritmus na vyrátanie objemu síce vráti len približný odhad, ale pre predstavu o objeme tento údaj nie je zlý. Je tu teda 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 navrhnúť presnejší algoritmus na nájdenie stropu alebo zefektívniť implementáciu navrhnutých algoritmov.

Zoznam použitej literatúry

- [1] BELLA, P., HLAVÁČOVÁ, I., AND HOLÚBEK, P. Zoznam jaskýň v Slovenskej republike. *SMOPaJ – SSJ – SSS, Liptovský Mikuláš* (2007), 364 s.
- [2] HOCHMUTH, Z. Krasové územia a jaskyne slovenska. *Geographia Cassoviensis* 210, 2 (2008), 1–209.
<https://geo.ics.upjs.sk/index.php/13-geographia-cassoviensis/239-gc-2009-3-3>.
- [3] JAILLET, S., SADIÉ, B., HAJRI, S., PLOYON, E., AND DELANNOY, J.-J. Une analyse 3d de l'endokarst : applications lasergrammétriques sur l'aven d'ornac (ardèche, france). *Géomorphology* 442, 4 (2011), 379–394. ISBN: 2-913282-54-4
<http://journals.openedition.org/geomorphologie/9594>.
- [4] 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>.

Prílohy

Príloha A: CD médium obsahujúce bakalársku prácu v elektronickej podobe, zdrojové kódy implementácie algoritmov uvedených v práci a spustiteľný JAR súbor naprogramovanej aplikácie so spomínaným používateľským rozhraním pre spúšťanie jednotlivých algoritmov.