

# Mapovanie polohy a výšky stromov v PointCloud dátach získaných pomocou LiDAR technológie

Marián Gall<sup>1</sup>, Michal Malček<sup>1</sup>, Dávid Murín<sup>2</sup>, Robert Straka<sup>2</sup>, Lucia Demovičová<sup>3</sup>

<sup>1</sup>Národné superpočítačové centrum, Dúbravská cesta 9, 845 35 Bratislava

<sup>2</sup>SKYMOVE s.r.o., Kráľovské údolie 20, 811 02 Bratislava

<sup>3</sup> Centrum spoločných činností SAV. v. v. i., organizačná zložka Výpočtové stredisko, Dúbravská cesta 9, 845 35 Bratislava

Cieľom spolupráce medzi Národným superpočítačovým centrom (NSCC) a firmou SKYMOVE, v rámci projektu Národného kompetenčného centra pre HPC, bol návrh a implementácia pilotného softvérového riešenia pre spracovanie dát získaných technológiou LiDAR (Light Detection and Ranging) umiestnených na dronoch.

## 1 Zber dát

LiDAR je inovatívna metóda diaľkového merania vzdialenosti, ktorá funguje na princípe výpočtu doby šírenia impulzu laserového lúča odrazeného od objektov. LiDAR vysiela svetelné impulzy, ktoré zasiahnu zem, alebo daný objekt, a vrátia sa späť, kde sú zachytené senzormi. Meraním času návratu svetla LiDAR určí vzdialenosť bodu, v ktorom sa laserový lúč odrazil.

LiDAR dokáže vysielať 100- až 300 000 impulzov za sekundu, pričom z každého metra štvorcového povrchu zachytí niekoľko desiatok až stoviek impulzov, v závislosti od konkrétneho nastavenia a vzdialenosti snímaného objektu. Týmto spôsobom sa vytvára tzv. mračno bodov (PointCloud) pozostávajúce, potenciálne, z miliónov bodov. Moderným využitím LiDAR-u je zber dát zo vzduchu, kde sa zariadenie umiestňuje na drony, čím sa zvyšuje efektívnosť a presnosť zberu dát. Na zber dát v tomto projekte boli použité drony od spoločnosti DJI, hlavne dron DJI M300 a Mavic 3 Enterprise (obr. 1). Dron DJI M300 je profesionálny dron navrhnutý pre rôzne priemyselné aplikácie a jeho parametre umožňujú, aby bol vhodným nosičom pre LiDAR.

Dron DJI M300 bol využitý ako nosič pre LiDAR značky Geosun (obr. 1). Ide o strednorozsahový, kompaktný systém s integrovaným laserovým skenerom a systémom na určovanie polohy a natočenia. Vzhľadom na pomer medzi rýchlosťou zberu a kvalitou dát boli dáta snímané z výšky 100 m nad povrchom, čím je možné zosnímať za pomerne krátky čas aj väčšie územia v postačujúcej kvalite.

Zozbierané dáta boli geolokalizované v súradnicovom systéme S-JTSK (EPSG:5514) a Baltskom výškovom systéme po vyrovnaní (Bpv), pričom súradnice sú udávané v metroch alebo metroch nad morom. Okrem lidarových dát bola súčasne vykonaná aj letecká fotogrametria, ktorá umožňuje tvorbu tzv. ortofotomozaiky. Ortofotomozaiky poskytujú fotografický záznam skúmanej oblasti vo vysokom rozlíšení (3 cm/pixel) a s polohovou presnosťou do 5 cm. Ortofotomozaika bola použitá ako podklad pre vizuálne overenie polôh jednotlivých stromov.



Obrázok 1. Dron DJI M300 (vľavo) a LiDAR značky Geosun (vpravo).

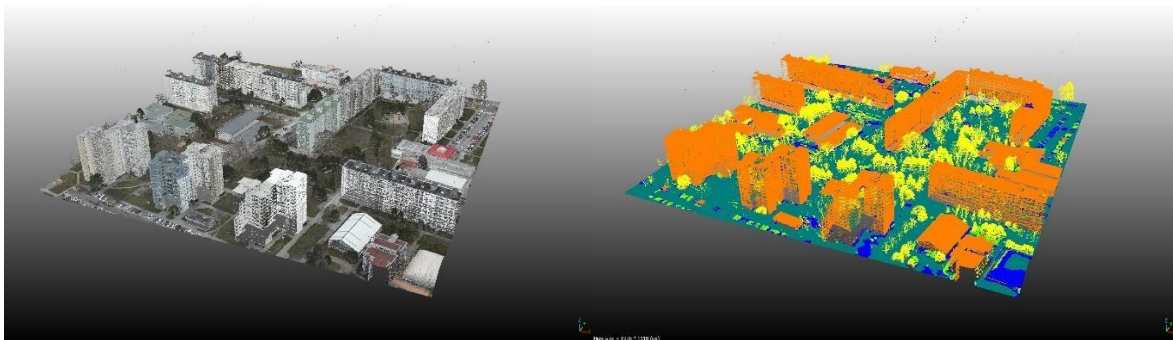
## 2 Klasifikácia dát

Nosným datasetom, ktorý vstupoval do automatickej identifikácie stromov, bolo lidarové mračno bodov vo formáte LAS/LAZ (nekomprimovaná a komprimovaná forma). LAS súbory sú štandardizovaným formátom pre ukladanie lidarových dát navrhnutý tak, aby zabezpečil efektívne ukladanie veľkého množstva bodových dát s presnými 3D súradnicami. LAS súbory obsahujú informácie o polohe (x, y, z), intenzite odrazu, klasifikácii bodov a ďalšie atribúty, ktoré sú nevyhnutné pre analýzu a spracovanie lidarových dát. Vďaka svojej štandardizácii a kompaktnosti sa LAS súbory často používajú v geodézii, kartografii, lesníctve, urbanistickom plánovaní a mnohých ďalších oblastiach, kde je potrebná detailná a presná 3D reprezentácia terénu a objektov.

Mračno bodov bolo potrebné najskôr spracovať do takej podoby, aby na ňom bolo možné čo najjednoduchšie identifikovať body jednotlivých stromov alebo vegetácie. Ide o proces, pri ktorom sa každému bodu v mračne bodov priradí určitá trieda, čiže hovoríme o klasifikácii.

Na klasifikáciu mračna bodov je možné použiť viacero nástrojov. V našom prípade sme sa, vzhľadom na dobré skúsenosti, rozhodli použiť softvér Lidar360 od spoločnosti GreenValley International [1]. V rámci klasifikácie mračna bodov boli jednotlivé body mračna klasifikované do nasledovných tried: neklasifikované (1), povrch (2), stredná vegetácia (4), vysoká vegetácia (5), budovy (6). Na klasifikáciu bola využitá metóda strojového učenia, ktorá po natrénovaní na reprezentatívnej trénovacej vzorke dokáže automaticky klasifikovať body ľubovoľného vstupného datasetu (obr. 2).

Trénovacia vzorka je vytvorená manuálnym klasifikovaním bodov mračna do jednotlivých tried. Na účely automatizovanej identifikácie stromov sú pre tento projekt podstatné hlavne triedy povrch a vysoká vegetácia. Avšak, pre čo najlepší výsledok klasifikácie vysokej vegetácie je vhodné zaradiť aj ostatné klasifikačné triedy. Trénovacia vzorka bola tvorená súborom viacerých menších oblastí z celého územia a zahŕňala všetky typy vegetácie, či už listnaté alebo ihličnaté, a taktiež rôzne typy budov. Na základe vytvorenej trénovacej vzorky boli následne automaticky klasifikované zvyšné body mračna. Kvalita trénovacej množiny má preto podstatný vplyv na výslednú klasifikáciu celého územia.



Obrázok 2. Ukážka mračna bodov oblasti zafarbeného pomocou ortofotomozaiky (vľavo) a pomocou príslušnej klasifikácie (vpravo) v programe CloudCompare.

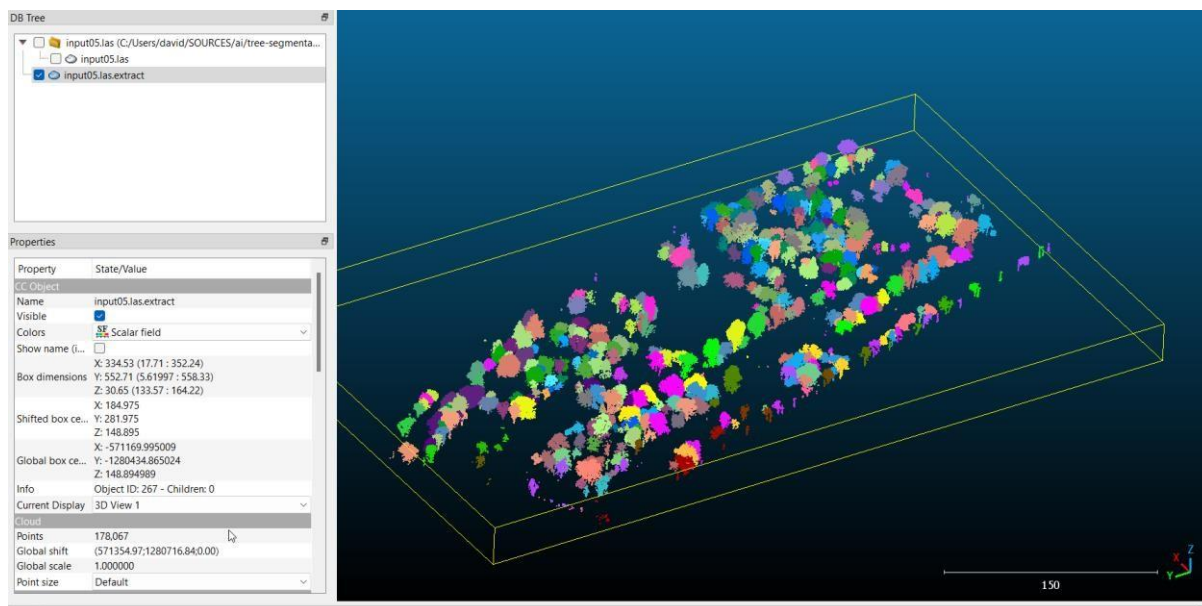
### 3 Segmentácia dát

Klasifikované mračno bodov bolo následne segmentované pomocou softvéru CloudCompare [2]. Segmentácia vo všeobecnosti znamená rozdelenie klasifikovaných dát na menšie celky – segmenty, ktoré spĺňajú spoločné charakteristické vlastnosti. Pri segmentácii vysokej vegetácie bolo cieľom priradiť jednotlivé body ku konkrétnemu stromu.

Na účely segmentácie stromov bol použitý plugin TreeIso v softvérovom balíku CloudCompare, ktorý automaticky rozpoznáva stromy na základe rôznych výškových a polohových kritérií (obr. 3). Celková segmentácia sa skladá z troch krokov:

1. Spájanie bodov, ktoré sú blízko seba, do segmentov a odstraňovanie šumu.
2. Spájanie susedných segmentov bodov do väčších celkov.
3. Zloženie jednotlivých segmentov do celku, ktorý tvorí jeden strom.

Výsledkom je kompletná segmentácia vysokej vegetácie. Tieto segmenty sa následne uložia do jednotlivých LAS súborov a použijú sa na následné spracovanie pre určenie polohy jednotlivých stromov. Veľkým nedostatkom tohto nástroja je, že pracuje len v sériovom režime, čiže dokáže využiť len jedno CPU jadro, čo značne limituje jeho použitie v HPC prostredí.



Obrázok 3. Segmentované mračno bodov v programe CloudCompare použitím plugin modulu TreeIso.

Ako alternatívnu metódu na segmentovanie sme skúmali aj využitie ortofotomozaiky daných oblastí. Pomocou metód strojového učenia sme sa pokúsili identifikovať jednotlivé koruny stromov na snímkach a na základe takto určených geolokalizačných súradníc identifikovať príslušné segmenty v LAS súbore. Na detekciu korún stromov z ortofotomozaiky bol použitý model YOLOv5 [3] s predtrénovanými váhami z databázy COCO128 [4]. Tréningové dáta pozostávali z 230 snímkov, ktoré boli manuálne anotované pomocou nástroja LabelImg [5]. Trénovacia jednotka pozostávala z 300 epoch, snímky boli rozdelené do sád po 16 vzoriek a ich veľkosť bola nastavená na 1000x1000 pixelov, čo sa ukázalo ako vhodný kompromis medzi výpočtovou náročnosťou a počtom stromov na daný výsek. Nedostatočná kvalita tohto prístupu bola obzvlášť markantná pre oblasti s hustou vegetáciou (zalesnených oblastí), ako je znázornené na obrázku 4. Domnievame sa, že to bolo spôsobené nedostatočnou robustnosťou zvolenej trénovacej sady, ktorá nedokázala dostatočne pokryť rôznorodosť obrazových dát (obzvlášť pre rôzne vegetatívne obdobia). Z týchto dôvodov sme segmentáciu z fotografických dát ďalej nerozvíjali a sústredili sme sa už iba na segmentáciu v mračne bodov.



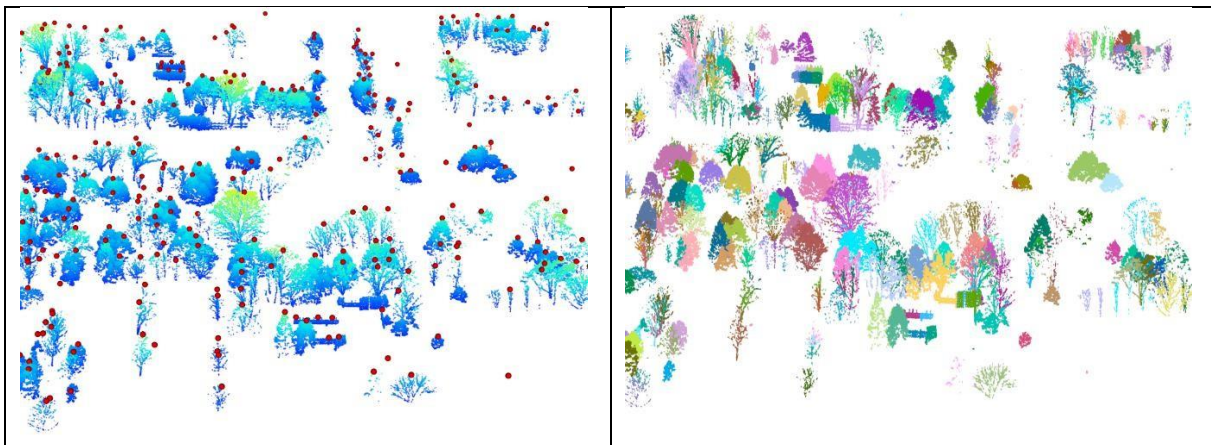
Obrázok 4. Segmentovanie stromov v ortofotomozaike pomocou nástroja YOLOv5. Obrázok ilustruje problém detekcie jednotlivých stromov v prípade hustej vegetácie (súvislého porastu).



Aby sme naplno využili možnosti superpočítača Devana, nasadili sme v jeho prostredí knižnicu lidR [6]. Táto knižnica, napísaná v jazyku R, je špecializovaný nástroj určený na spracovanie a analýzu lidarových dát, poskytuje rozsiahly súbor funkcií a nástrojov pre čítanie, manipuláciu, vizualizáciu a analýzu LAS súborov. S knižnicou lidR je možné efektívne vykonávať úlohy ako filtrovanie, klasifikácia, segmentácia a extrakcia objektov priamo z mračien bodov. Knižnica tiež umožňuje interpoláciu povrchov, vytváranie digitálnych modelov terénu (DTM) a digitálnych modelov povrchu (DSM) a výpočet rôznych metrických parametrov vegetácie a štruktúry krajiny. Vďaka svojej flexibilita a výkonnosti je lidR populárnym nástrojom v oblasti geoinformatiky a je zároveň vhodným nástrojom pre prácu v HPC prostredí, keďže väčšina funkcií a algoritmov je plne paralelizovaná v rámci jedného výpočtového uzla, čo umožňuje naplno využívať dostupný hardvér. V prípade spracovania veľkých datasetov, keď výkon alebo kapacita jedného výpočtového uzla už nie je postačujúca, môže byť rozdelenie datasetu na menšie časti, a ich nezávislé spracovanie, cesta k využitiu viacerých výpočtových HPC uzlov súčasne.

V knižnici lidR je dostupná funkcia `locate_trees()`, ktorá dokáže pomerne spoľahlivo identifikovať polohu stromov. Na základe zvolených parametrov a algoritmu funkcia analyzuje mračno bodov a identifikuje polohu stromov. V našom prípade bol použitý algoritmus `lmf` pre lokalizáciu založený na maximálnej výške [7]. Algoritmus je plne paralelizovaný, takže dokáže efektívne spracovať relatívne veľké zvolené oblasti v krátkom čase.

Takto určené polohy stromov sa dajú následne použiť v algoritme `silva2016` na segmentáciu vo funkcii `segment_trees()` [8]. Táto funkcia segmentuje príslušné nájdené stromy do osobitných LAS súborov (obr. 5), podobne ako plugin modul `TreeIso` v programe `CloudCompare`. Následne sa takto segmentované stromy v LAS súboroch použijú na ďalšie spracovanie, konkrétne na určenie polohy jednotlivých stromov, napríklad pomocou klastrovacieho algoritmu `DBSCAN` [9].

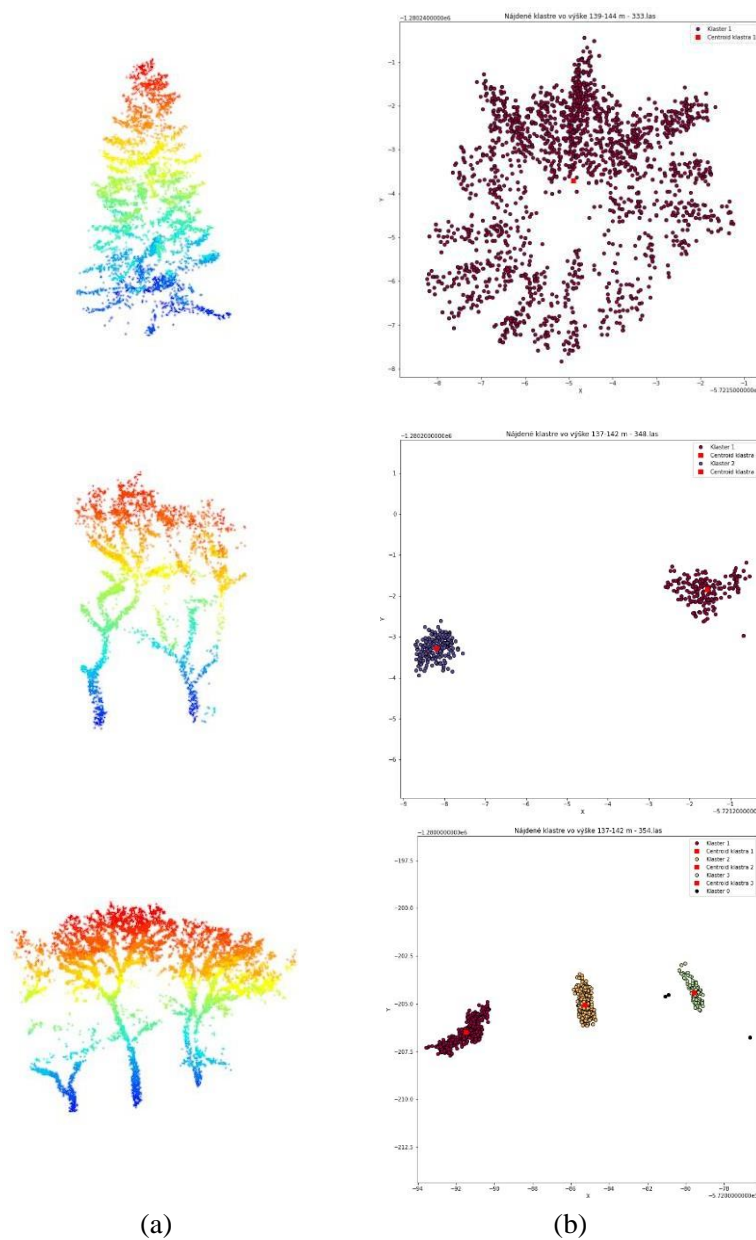


Obrázok 5. Polohy stromov zistené pomocou algoritmu "lmf" (vľavo, červené body) a príslušné segmenty stromov určené algoritmom `silva2016` (vpravo), pomocou knižnice lidR.

#### 4 Detekcia kmeňov stromov pomocou klastrovacieho algoritmu DBSCAN

Na určenie polohy a výšky stromov v jednotlivých LAS súboroch získaných segmentáciou sme použili rôzne prístupy. Výška jednotlivých stromov bola získaná na základe z-ových súradníc pre jednotlivé LAS súbory ako rozdiel minimálnej a maximálnej súradnice mračien bodov. Keďže jednotlivé výšky z mračna bodov obsahovali v niektorých prípadoch aj viac ako jeden strom, bolo potrebné identifikovať počet kmeňov stromov v rámci týchto výškov.

Kmene stromov boli identifikované na základe klastrovacieho algoritmu DBSCAN, pracujúceho s nasledovnými nastaveniami: maximálna vzdialenosť dvoch bodov v rámci jedného klastra (= 1 meter) a minimálny počet bodov v jednom klastru (= 10). Poloha každého identifikovaného kmeňa bola následne získaná na základe x-ových a y-ových súradníc geometrických stredov (centroidov) klastrov. Identifikácia klastrov pomocou DBSCAN algoritmu je ilustrovaná na obrázku 6.

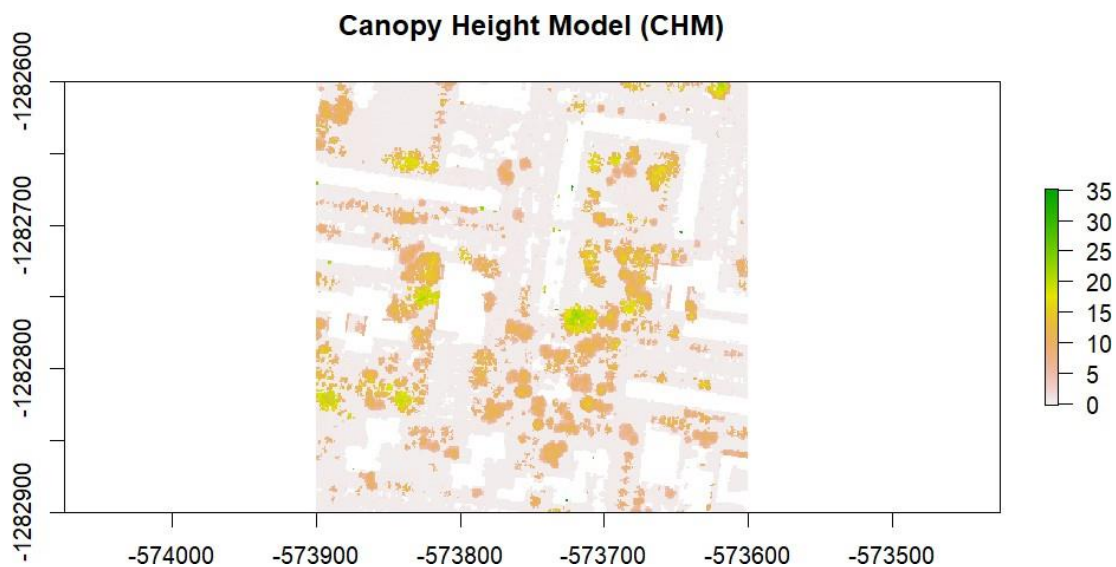


Obrázok 6. Výšky z mračna bodov, PointCloud-u (stĺpec vľavo) a príslušné detegované klastre vo výške 1-5 metrov (stĺpec vpravo).

## 5 Zistenie výšky stromov pomocou interpolácie povrchov

Ako alternatívnu metódu na určenie výšok stromov sme použili tzv. Canopy Height Model (CHM). CHM je digitálny model, ktorý predstavuje výšku stromovej obálky nad terénom. Tento model sa používa na výpočet výšky stromov v lese alebo inom vegetačnom poraste. CHM sa vytvára odčítaním digitálneho modelu terénu (DTM) od digitálneho modelu povrchu (DSM). Výsledkom je mračno bodov alebo raster, ktorý zobrazuje výšku stromov nad povrchom terénu (obr. 7).

Ak teda poznáme súradnice polohy stromu, pomocou tohto modelu môžeme jednoducho zistiť príslušnú výšku objektu (stromu) v danom bode. Výpočet tohto modelu je možné jednoducho uskutočniť použitím knižnice lidR pomocou funkcií `grid_terrain()`, ktorá vytvára DTM, a `grid_canopy()`, ktorá počíta DSM.

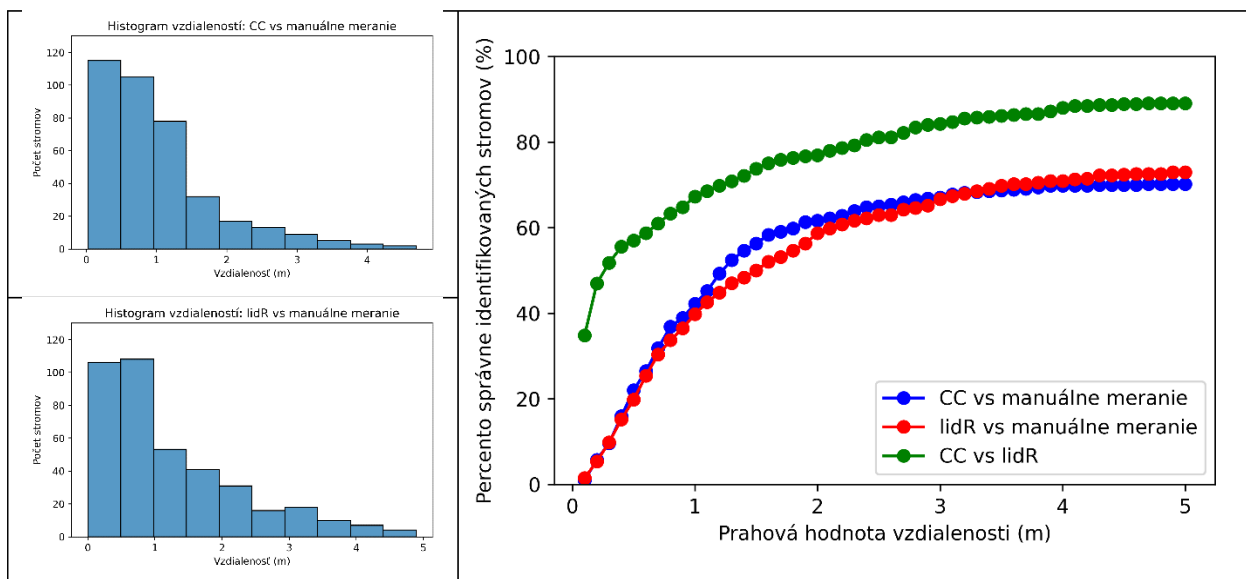


Obrázok 7. Canopy Height Model (CHM) pre skúmanú oblasť (na osiach X a Y sú uvedené súradnice v metroch), výška každého bodu v metroch je reprezentovaná pomocou farebnej škály.

## 6 Porovnanie výsledkov

Pre porovnanie dosiahnutých výsledkov vyššie popísanými prístupmi sme sa zameriavali na oblasť Petržalky v Bratislave, kde už boli vykonané manuálne merania polôh a výšok stromov. Z celej oblasti (približne 3500x3500 m) sme vybrali reprezentatívnu menšiu oblasť o rozmeroch 300x300 m (obr. 2). Získali sme tak výsledky pre plugin modul TreeIso v

programe CloudCompare (CC), pričom sme pracovali na PC v prostredí Windows, a výsledky pre algoritmy vo funkciách `locate_trees()` a `segment_trees()` pomocou knižnice `lidR` v HPC prostredí superpočítača Devana. Polohy stromov sme následne kvalitatívne a kvantitatívne vyhodnotili pomocou algoritmu Munkres (Hungarian Algorithm) [10] na optimálne párovanie. Algoritmus Munkres, tiež známy ako Maďarský algoritmus, je efektívny algoritmus na nájdenie optimálneho párovania v bipartitných grafoch. Jeho použitie pri párovaní stromov s manuálne určenými polohami stromov znamená nájdenie najlepšej zhody medzi identifikovanými stromami z lidarových dát a ich známymi polohami. Následne pri určení vhodnej hranice vzdialenosti v metroch (napríklad 5 m) potom vieme kvalitatívne zistiť počet presne určených polôh stromov. Výsledky sú spracované pomocou histogramov a percentuálne určujú správne polohy stromov v závislosti od zvolenej hranice presnosti (obr. 8). Zistili sme, že obe metódy dosahujú pri hranici vzdialenosti 5 metrov takmer rovnaký výsledok, približne 70% správne určených polôh stromov. Metóda použitá v programe CloudCompare vykazuje lepšie výsledky, resp. vyššie percento pri nižších prahových hodnotách, čo odzrkadľujú aj príslušné histogramy (obr. 8). Pri porovnaní oboch metód navzájom dosahujeme až približne 85% zhody pri prahovej hodnote do 5 metrov, čo poukazuje na kvalitatívnu vyrovnanosť oboch použitých prístupov. Kvalitu dosiahnutých výsledkov ovplyvňuje hlavne presnosť klasifikácie vegetácie v bodových mračnách, pretože prítomnosť rôznych artefaktov, ktoré sú nesprávne klasifikované ako vegetácia, skresľuje finálne výsledky. Algoritmy na segmentáciu stromov nedokážu vplyv týchto artefaktov eliminovať.



Obrázok 8. Histogramy vľavo zobrazujú počet správne identifikovaných stromov v závislosti od zvolenej prahovej hodnoty vzdialenosti v metroch (hore CC metóda a dole lidR metóda). Grafy vpravo ukazujú percentuálnu úspešnosť správne identifikovaných polôh stromov v závislosti od použitej metódy a od zvolenej prahovej hodnoty vzdialenosti v metroch.

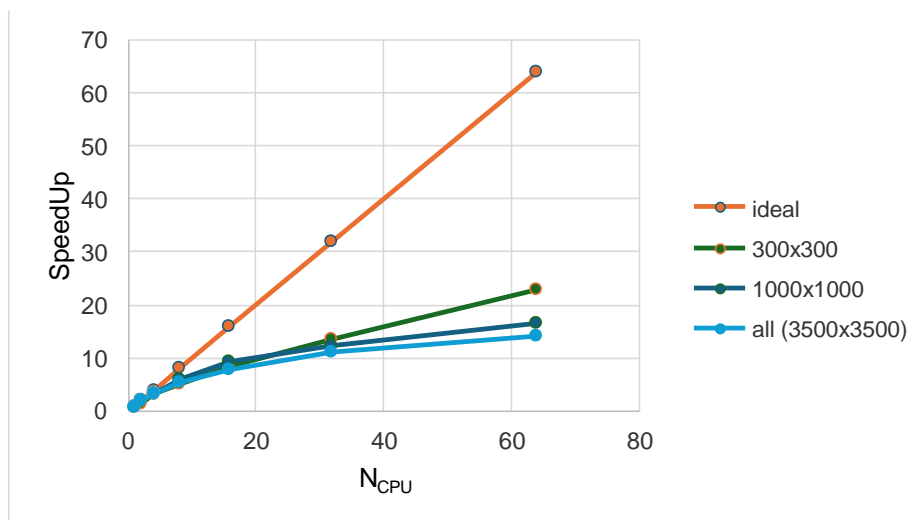


## 7 Analýza paralelnej efektivity algoritmu `locate_trees()` v knižnici `lidR`

Na zistenie efektivity paralelizácie hľadania vrcholov stromov v knižnici `lidR`, pomocou funkcie `locate_trees()`, sme daný algoritmus aplikovali na rovnaké študované územie s rôznym počtom CPU jadier – 1, 2, 4 až po 64 (maximum HPC uzla). Aby sme zistili, či je daný algoritmus citlivý aj na veľkosť problému, otestovali sme ho na troch územiach s rôznou veľkosťou – 300x300, 1000x1000 a 3500x3500 metrov. Dosiagnuté časy sú zobrazené v Tabuľke 1 a škálovateľnosť algoritmu je znázornená na obrázku 9. Výsledky ukazujú, že škálovateľnosť algoritmu nie je ideálna. Pri použití približne 20 jadier CPU klesá efektivita algoritmu na približne 50%, pri použití 64 jadier CPU je efektivita algoritmu už len na úrovni 15-20%. Efektivitu algoritmu ovplyvňuje aj veľkosť problému - čím väčšie územie, tým menšia efektivita, aj keď tento efekt nie je až tak výrazný. Na záver môžeme konštatovať, že na efektívne využitie daného algoritmu je vhodné použiť 16-32 CPU jadier a vhodným rozdelením daného skúmaného územia na menšie časti dosiahnuť maximálne efektívne využitie dostupného hardvéru. Použitie viac ako 32 CPU jadier síce už nie je efektívne, ale umožňuje ďalšie urýchlenie výpočtu.

A / [m x m]	CPU						
	1	2	4	8	16	32	64
300x300	34,4	24,0	10,7	6,8	4,1	2,6	1,5
1000x1000	363,0	187,8	110,4	60,6	38,8	29,7	22,0
3500x3500	11520,0	5832,0	3376,8	2156,4	1502,4	1029,6	822,6

Tabuľka 1. Dosiagnuté časy behu [s] algoritmu `lmf` vo funkcii `locate_trees()` knižnice `lidR` ( $t$ , v sekundách) pri rôznom počte jadier CPU a rôznych veľkostiach študovaného územia (A, [m x m]).



Obrázok 9. Zrýchlenie (SpeedUp) algoritmu `lmf` vo funkcii `locate_trees()` knižnice `lidR` v závislosti od počtu CPU jadier ( $N_{CPU}$ ) a veľkosti študovaného územia (v metroch).

## 8 Závěrečné zhodnotenie

Zistili sme, že pre dosiahnutie dobrých výsledkov je extrémne dôležité správne nastavenie parametrov použitých algoritmov, keďže počet a kvalita výsledných polôh stromov sú od nich veľmi závislé. Na získanie čo najpresnejších výsledkov je vhodné vybrať reprezentatívnu časť skúmanej oblasti, manuálne zistiť polohy stromov a následne nastaviť parametre príslušných algoritmov. Takto optimalizované nastavenia môžu následne byť použité na analýzu celej skúmanej oblasti.

Kvalitu výsledkov ovplyvňuje taktiež množstvo iných faktorov, ako napríklad ročné obdobie, ktoré má vplyv na hustotu vegetácie, alebo hustota stromov v danej oblasti a druhová variabilita vegetácie. Kvalitu výsledkov ovplyvňuje aj kvalita klasifikácie vegetácie v mračne bodov, pretože prítomnosť rôznych artefaktov, ako sú časti budov, cesty, dopravné prostriedky a iné objekty, môže následne negatívne skresliť výsledky, keďže použité algoritmy na segmentáciu stromov nedokážu tieto artefakty vždy spoľahlivo odfiltrovať.

Z hľadiska efektivity výpočtov môžeme konštatovať, že použitie HPC prostredia poskytuje zaujímavú možnosť násobného urýchlenia vyhodnocovacieho procesu. Na ilustráciu môžeme uviesť, že spracovanie, napríklad, celej skúmanej oblasti Petržalky (3500x3500 m) trvalo na jednom výpočtovom uzle HPC systému Devana približne 820 sekúnd, pri využití všetkých (t.j. 64) CPU jadier. Spracovanie danej oblasti v programe CloudCompare na výkonnom PC, pri použití jedného CPU jadra, trvalo približne 6200 sekúnd, čo je asi 8-krát pomalšie.

### Zdroje:

- [1] <https://www.greenvalleyintl.com/LiDAR360/>
- [2] <https://github.com/CloudCompare/CloudCompare/releases/tag/v2.13.1>
- [3] <https://github.com/ultralytics/yolov5>
- [4] <https://www.kaggle.com/ultralytics/coco128>
- [5] <https://github.com/heartexlabs/labelImg>
- [6] Roussel J., Auty D. (2024). Airborne LiDAR Data Manipulation and Visualization for Forestry Applications.
- [7] Popescu, Sorin & Wynne, Randolph. (2004). Seeing the Trees in the Forest: Using Lidar and Multispectral Data Fusion with Local Filtering and Variable Window Size for Estimating Tree Height. *Photogrammetric Engineering and Remote Sensing*. 70. 589-604. 10.14358/PERS.70.5.589.
- [8] Silva C. A., Hudak A. T., Vierling L. A., Loudermilk E. L., Brien J. J., Hiers J. K., Khosravipour A. (2016). Imputation of Individual Longleaf Pine (*Pinus palustris* Mill.) Tree Attributes from Field and LiDAR Data. *Canadian Journal of Remote Sensing*, 42(5).
- [9] Ester M., Kriegel H. P., Sander J., Xu X.. *KDD-96 Proceedings* (1996) pp. 226–231
- [10] Kuhn H. W., "The Hungarian Method for the assignment problem", *Naval Research Logistics Quarterly*, 2: 83–97, 1955