



**Vilniaus
universitetas**

Ataskaitinė informatikos ir informatikos inžinerijos kryptių doktorantų ataskaitinė konferencija 2026-10-08

Kasparas Karlauskas, Informatikos inžinerija 4-as kursas

Preliminari tema:

Trimačių taškų debesų analizė ir objektų atpažinimas
(Three dimensional point cloud analysis and object detection)

Vadovas: prof. dr. Povilas Treigys

Studijų pradžios ir pabaigos metai: 2023–2027

Studijų metai: IV (2026 / 2027)

Ataskaitinis laikotarpis: III studijų metų pavasario semestras

Studijų plano vykdymo suvestinė

Vilniaus
universitetas

Studijų metai	Egzaminai	
	Planas	Įvykdyta
I (2023/2024)	2	2
II (2024/2025)	2	2
III (2025/2026)		
IV (2026/2027)		
Iš viso:	4	4

Studijų plano vykdymo suvestinė

Vilniaus
universitetas

Studijų metai	Dalyvavimas konferencijose				Publikacijos					
	Tarptautinėse		Nacionalinėse		Su citavimo rodikliu			Be citavimo rodiklio		
	Planas	Ivykdyta	Planas	Ivykdyta	Planas	Ivykdyta	Būklė	Planas	Ivykdyta	Būklė
I (2023/2024)	1	1						1	0	
II (2024/2025)	1	1			0	1		1 + 1 (skola iš I metų)	1	Publikuota
III (2025/2026)	1	1			1 - 1 (atlikta anksčiau II metais)	0		1 (skola iš II metų)	0	
IV (2026/2027)	1				1					
Iš viso:	4	3	0	0	2	1		2	1	

Tyrimo objektas

Vilniaus
universitetas

- Trimačiai taškų debesys, užfiksuoti naudojantis lazerinio skenavimo aparatais, pritvirtintais prie dronų, leidžia atlikti sudėtingesnę analizę negu dvimačiai vaizdai. Miškų taškų debesys naudojami miškininkijoje (medžių inventorizacija) bei infrastruktūroje (elektros perdavimo laidų saugumas).
- Individualių medžių segmentacija žemos skiriamosios gebos lazerinio skenavimo 3D taškų debesyse
 - Kol kas vyrauja geometriniai algoritmai, kurių veikimas tankiose medžių zonose ne idealus, sudėtinga atitaikyti algoritmo parametrus (per smulkus/stambus segmentavimas);
 - Literatūroje pateikiami mašininio mokymosi metodai orientuoti į antžeminius lazerinio skenavimo duomenis, arba dronu užfiksuotus aukštos skiriamosios gebos duomenis (žemas skrydis);
 - Sužymėti duomenys individualių medžių segmentacijai reti, gavimo procesas sudėtingas.

Moksliniai rezultatai

1. Tarptautinėje konferencijoje International Symposium on Remote Sensing 2026 pristatytas tyrimas apie „vokselių entropijos“ taikymą medžių aptikimo algoritmuose
 - a. Taškų pašalinimas aukštos entropijos regionuose (tikriausiai lapai) prieš naudojant kitą medžių aptikimo algoritmą, kad pagreitinti skaičiavimus [tas pats metodas kaip 2025 konferencijos straipsnyje, tik su greitesniu algoritmu]
 - b. Kamienų aptikimui tinkamo horizontalaus skerspjuvio aptikimas minimizuojant „vokselių entropiją“
 - c. Originalus greitas kamienų aptikimo algoritmas, taškų filtravimui naudojantis tik „vokselių entropiją“
2. Metodas patobulintas, pasiekta galutinė versija tinkama publikavimui (1.a ir 1.c taškų filtravimo pokyčiai)
3. Patobulintas ROBPCA algoritmas cilindro ašies nustatymui – daug greitesnis veikimas su ekvivalenčiomis išvestimis (naudojimas nukeltas dar sekančiam straipsniui, turbūt konferencijai)
4. Ruošiantis mašininio mokymosi tyrimui, atlikti šie darbai:
 - a. Tyrinėti GPU atminties reikalavimų mokymuisi sumažinimo metodai, „Checkpoint Activation“ funkcija „PyTorch“ pakete pritaikyta „Pointcept“ paketo duomenų tipui
 - b. Pradėtas duomenų žymėjimas – medžių koordinatės neapdorotuose SYSIFOSS taškų debesyse

Konferencijos ISRS2026 pranešimas

Stendinis pranešimas apie „vokselių entropijos“ taikymą medžių kamienų aptikimo algoritmuose:

Du atviro kodo metodai (kitų tyrėjų):

3DFin – medžių kamienams aptikti nauojamas **vienas horizontalus skerspjuvis**

TreeTool – medžių kamienams **naudojamas pilnas taškų debesis**, filtravimas pagal vietinį kreivumą

Naujas metodas:

VEstems-det – medžių kamienams naudojamas **pilnas taškų debesis**, filtravimas pagal „vokselių entropiją“



Voxel Entropy Applications for Individual Tree Detection in High-definition Forest Point Clouds

Kasparas KARLAUSKAS¹, Povilas TREIGYS¹

¹Institute of Data Science and Digital Technologies, Vilnius University, Vilnius, Lithuania.

Abstract

High-definition 3D point clouds have enabled detailed analysis of forest areas via classical and learning-based algorithms. Individual Tree Detection, the task of finding the coordinates of all individual trees in a point cloud, is a pre-requisite for tree-level analysis such as inventoryization or tree species classification. In Terrestrial and Mobile Laser Scanning, forest point clouds are captured from a low altitude below the canopy, making the stems of individual trees well-defined, providing a strong signal for detection. However, high-definition point clouds have high memory requirements, and due to the surface reflectivity and local geometry differences between stems and leaves, most laser pulse reflections and therefore most recorded points come from the canopy layer – data that is usually not necessary for Individual Tree Detection. Voxel Entropy, a measure of point distribution homogeneity within discrete cube volumes, can be used as a computationally efficient (linear time) coarse-scale discriminator between tree canopy and stem regions. This study showcases ways to seamlessly integrate Voxel Entropy into existing Individual Tree Detection workflows: pre-filtering trivial canopy points to reduce computational load of neighbourhood-based methods; automatically determining optimal regions of interest for methods based on narrow z-axis crops of the original forest point cloud. Finally, a competitive stand-alone method using Voxel Entropy as the only geometric feature for Individual Tree Detection is presented.

Voxel Entropy

Voxel Entropy [1] is a measure of point distribution homogeneity within discrete cube volumes, voxels, and can be used as a computationally efficient (linear time) coarse-scale discriminator between tree canopy and stem regions. Given Shannon entropy:

$$H = - \sum_{x \in X} p(x) \log(p(x)) \quad (1)$$

Voxel Entropy is computed by splitting each voxel into sub-voxels, and defining $p(x)$ as the probability for a randomly sampled point to be inside the subvoxel.

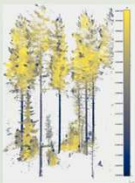
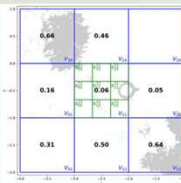


Figure 1: Voxel Entropy computation with 1-meter voxels and a horizontal 3 x 3 split.

Figure 2: Voxel Entropy in a sparse cluster of trees. Darker points have lower entropy.

$$H(V_{ijk}) = - \frac{1}{\log(s_x s_y s_z)} \sum_i \sum_j \sum_k \frac{|S_{ijk}^R|}{|V_{ijk}|} \log \frac{|S_{ijk}^R|}{|V_{ijk}|} \quad (2)$$

where $| \cdot |$ denotes the number of elements in a set. When used as a point-wise feature, every point inside V_{ijk} has the same value of Voxel Entropy $H(V_{ijk})$. An application of point-wise Voxel Entropy for Individual Tree Detection will be as follows:

1. Compute Voxel Entropy (VE) with 1 m voxels and a horizontal 3 x 3 split;
2. Remove points with $VE < 0.5$;
3. Identify initial clusters with Connected Component Segmentation as in [5]
4. Post-process clusters – reject non-elongated clusters via standard deviation [5], merge clusters close on the XY-plane, remove shorter clusters closer than 0.5 m

Data

The ForestSemantic[3] dataset consists of three forest point clouds of increasing scene complexity (undergrowth density, occlusions, etc.) including locations and heights of every tree.

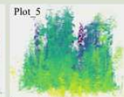
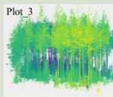


Figure 3: The ForestSemantic dataset. Points are colored by distance to the camera.

Key Results

Three relevant applications of Voxel Entropy (VE) in Individual Tree Detection were implemented and validated:

1. VE can be used to **determine the optimal height band for stem detection** – the algorithm 3DFin [2] fails for the difficult Plot_5 with the default value, while detecting some trees with the automatically determined value.
2. VE can be used to **preemptively remove a large part of foliage points**, leading to **sped up computation with comparable performance** for algorithms processing the entire point cloud – TreeTool [4] total runtimes were decreased; from 20 % in Plot_5 up to 34 % in Plot_3.
3. VE can be used as a **stand-alone feature for Individual Tree Detection** – preliminary experiments show performance comparable to other open-source algorithms.

Table 1: Individual Tree Detection evaluation results.

Plot	Method	TP	FP	FN	prec	rec	f1	t [s]
Plot_1	3DFin	48	21	3	0.696	0.941	0.800	148
	3DFin+VE	44	8	7	0.846	0.863	0.854	138
	TreeTool	44	5	7	0.898	0.863	0.880	70
	TreeTool+VE	44	4	7	0.917	0.863	0.889	50
Plot_3	VE	44	19	7	0.698	0.863	0.772	16
	3DFin	54	11	83	0.831	0.394	0.555	188
	3DFin+VE	86	1	51	0.989	0.628	0.768	120
	TreeTool	80	3	57	0.964	0.584	0.727	105
	TreeTool+VE	74	3	63	0.861	0.540	0.692	69
	VE	85	17	52	0.833	0.620	0.711	22
Plot_5	3DFin	0	0	116	0.000	0.000	0.000	0
	3DFin+VE	12	6	104	0.667	0.103	0.179	133
	TreeTool	34	1	82	0.971	0.293	0.450	75
	TreeTool+VE	30	1	86	0.968	0.259	0.408	60
	VE	28	22	88	0.550	0.241	0.337	18

Bold – best result for the plot; **Bold italic** – second best result for the plot

The stand-alone algorithm has recall comparable to the baseline algorithms, however its precision falls short. The current post-processing is a basic version of the aspect-ratio based filtering from Zhang et al. [5] with additional proximity merging and non-maximum suppression. Compared to the baseline approaches, there is still a lot of time budget for more stringent cluster post-processing – the stand-alone VE algorithm is still over three times faster than the fastest baseline run. Future work will focus on improving both precision and recall for the stand-alone algorithm.

References

- [1] Karlauskas, K., Treigys, P., 2025. Entropy-based Pre-filtering of High-density Forest Point Clouds for Individual Tree Detection Algorithms, in: Proceedings of the 19th International Symposium on Spatial and Temporal Data. Presented at the SSTD '25: 19th International Symposium on Spatial and Temporal Data, ACM, Osaka Japan, pp. 137–140. <https://doi.org/10.1145/3748777.3748806>
- [2] Laino, D., Cabo, C., Prendes, C., Janvier, R., Ordonez, C., Nikonov, T., Doerr, S., Santin, C., 2024. 3DFin: a software for automated 3D forest inventories from terrestrial point clouds. Forestry: An International Journal of Forest Research 97, 479–496. <https://doi.org/10.1093/forestry/cpae020>
- [3] Liang, X., Qi, H., Dong, X., Chen, J., Cai, S., Zhang, Q., Wang, Y., Kukko, A., Hyypyi, J., 2025. ForestSemantic: a dataset for semantic learning of forest from close-range sensing. Geospatial Information Science 28, 185–211. <https://doi.org/10.1080/10095020.2024.213325>
- [4] Montoya, O., Icaño-Hernández, O., Salas, J., 2021. TreeTool: A tool for detecting trees and estimating their DBH using forest point clouds. SoftwareX 16, 100889. <https://doi.org/10.1016/j.softx.2021.100889>
- [5] Zhang, W., Wan, P., Wang, T., Cai, S., Chen, Y., Jin, X., Yan, G., 2019. A Novel Approach for the Detection of Standing Tree Stems from Plot-Level Terrestrial Laser Scanning Data. Remote Sensing 11, 211. <https://doi.org/10.3390/rs11020211>

Acknowledgements

The authors are thankful for the high-performance computing resources provided by the Information Technology Research Center of Vilnius University.

Contact

kasparas.karlauskas@mif.stud.vu.lt

Konferencijos ISRS2026 pranešimas

Vilniaus
universitetas

3DFin + VE – skerspjūvis pasiūlomas automatiškai, pagal vidutinės entropijos minimumą (ieškomas praretėjimas tarp žemos augmenijos ir lajos)

Rezultatas: automatiškai nustatytas pjūvis efektyvus aptinkant aukštus medžius

TreeTool + VE – išankstinis filtravimas pašalinant didelės entropijos regionus, kad skaičiuoti vietinį kreivumą tik tiems taškams, kuriems to reikia

Rezultatas: skaičiavimai > 20 % trumpesni, bet sudėtingesniems scenoms rezultatas visada nežymiai prastesnis (ir *precision*, ir *recall*)

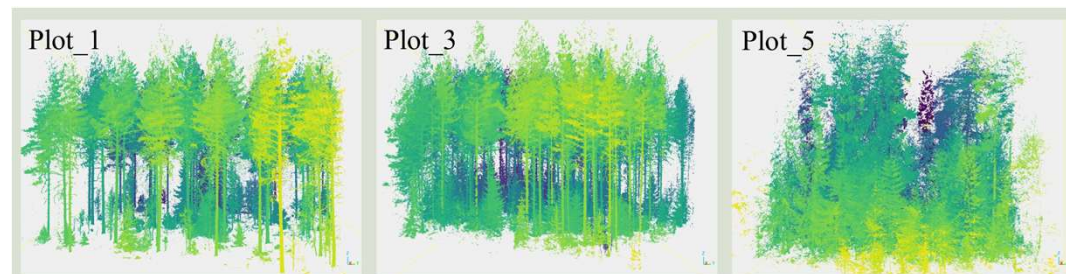


Figure 3: The *ForestSemantic* dataset. Points are colored by distance to the camera.

Table 1: Individual Tree Detection evaluation results.

Plot	Method	TP	FP	FN	prec	rec	f1	t [s]
Plot_1	3DFin	48	21	3	0.696	0.941	0.800	148
	3DFin+VE	44	8	7	0.846	0.863	0.854	138
	TreeTool	44	5	7	0.898	0.863	0.880	70
	TreeTool+VE	44	4	7	0.917	0.863	0.889	50
	VE	44	19	7	0.698	0.863	0.772	16
Plot_3	3DFin	54	11	83	0.831	0.394	0.535	188
	3DFin+VE	86	1	51	0.989	0.628	0.768	120
	TreeTool	80	3	57	0.964	0.584	0.727	105
	TreeTool+VE	74	3	63	0.961	0.540	0.692	69
	VE	85	17	52	0.833	0.620	0.711	22
Plot_5	3DFin	0	0	116	0.000	0.000	0.000	0
	3DFin+VE	12	6	104	0.667	0.103	0.179	133
	TreeTool	34	1	82	0.971	0.293	0.450	75
	TreeTool+VE	30	1	86	0.968	0.259	0.408	60
	VE	28	22	88	0.560	0.241	0.337	18

Bold – best result for the plot; **Bold italic** – second best result for the plot

Konferencijos ISRS2026 pranešimas

Vilniaus
universitetas

3DFin + VE – skerspjūvis pasiūlomas automatiškai, pagal vidutinės entropijos minimumą (ieškomas praretėjimas tarp žemos augmenijos ir lajos)

Rezultatas: automatiškai nustatytas pjūvis efektyvus aptinkant aukštus medžius

TreeTool + VE – išankstinis filtravimas pašalinant didelės entropijos regionus, kad skaičiuoti vietinį kreivumą tik tiems taškams, kuriems to reikia

Rezultatas: skaičiavimai > 20 % trumpesni, bet sudėtingesnėms scenoms rezultatas visada nežymiai prastesnis (ir *precision*, ir *recall*)



Figure 3: The *ForestSemantic* dataset. Points are colored by distance to the camera.

Table 1: Individual Tree Detection evaluation results.

Plot	Method	TP	FP	FN	prec	rec	f1	t [s]
Plot_1	3DFin	48	21	3	0.696	0.941	0.800	148
	3DFin+VE	44	8	7	0.846	0.863	0.854	138
	TreeTool	44	5	7	0.898	0.863	0.880	70
	TreeTool+VE	44	4	7	0.917	0.863	0.889	50
	VE	44	19	7	0.698	0.863	0.772	16
Plot_3	3DFin	54	11	83	0.831	0.394	0.535	188
	3DFin+VE	86	1	51	0.989	0.628	0.768	120
	TreeTool	80	3	57	0.964	0.584	0.727	105
	TreeTool+VE	74	3	63	0.961	0.540	0.692	69
	VE	85	17	52	0.833	0.620	0.711	22
Plot_5	3DFin	0	0	116	0.000	0.000	0.000	0
	3DFin+VE	12	6	104	0.667	0.103	0.179	133
	TreeTool	34	1	82	0.971	0.293	0.450	75
	TreeTool+VE	30	1	86	0.968	0.259	0.408	60
	VE	28	22	88	0.560	0.241	0.337	18

Bold – best result for the plot; **Bold italic** – second best result for the plot

Konferencijos ISRS2026 pranešimas

Vilniaus
universitetas

VEstems-det (plakate VE) turi palyginimą arba net geresnį *recall*, tačiau yra daug daugiau klaidingai aptiktų kamienų (žemesnis *precision*)

Galbūt įdomu, bet nežinojau kiek verta rimtesnio žurnalo dėmesio...

Kaip pagerinti medžių aptikimo tikslumą nepaaukojant šio metodo pagrindinės patrauklios savybės – skaičiavimų greičio?



Figure 3: The *ForestSemantic* dataset. Points are colored by distance to the camera.

Table 1: Individual Tree Detection evaluation results.

Plot	Method	TP	FP	FN	prec	rec	f1	t [s]
Plot_1	3DFin	48	21	3	0.696	0.941	0.800	148
	3DFin+VE	44	8	7	0.846	0.863	0.854	138
	TreeTool	44	5	7	0.898	0.863	0.880	70
	TreeTool+VE	44	4	7	0.917	0.863	0.889	50
	VE	44	19	7	0.698	0.863	0.772	16
Plot_3	3DFin	54	11	83	0.831	0.394	0.535	188
	3DFin+VE	86	1	51	0.989	0.628	0.768	120
	TreeTool	80	3	57	0.964	0.584	0.727	105
	TreeTool+VE	74	3	63	0.961	0.540	0.692	69
	VE	85	17	52	0.833	0.620	0.711	22
Plot_5	3DFin	0	0	116	0.000	0.000	0.000	0
	3DFin+VE	12	6	104	0.667	0.103	0.179	133
	TreeTool	34	1	82	0.971	0.293	0.450	75
	TreeTool+VE	30	1	86	0.968	0.259	0.408	60
	VE	28	22	88	0.560	0.241	0.337	18

Bold – best result for the plot; **Bold italic** – second best result for the plot

ROBPCA pagreitinimas naudojant vokselius su svoriais

Dėl gretimos augmenijos ir šakų įtakos, kartais kamieno viduryje atsiranda didelės entropijos regionų, taigi po filtravimo kamienai turi tarpus viduryje.

Praeitą semestrą mėginau atgauti klaidingai atmestus kamienų taškus tinkindamas cilindro ašį su ROBPCA (*ROBust Principal Component Analysis*) metodu.

Problema: skaičiuojant *outlier score* ir imant taškus cilindro ašies tinkinimui, reikia daug kartų projektuoti ir rūšiuoti kiekvieną tašką

1. Outlier score (*project-pursuit*)
 - ① **Project every point** to random line $l \in L$
 - ② **Sort** by projected coordinate $x_{l,i}$
 - ③ Find a subset of consecutive points with the lowest variance σ_l (corresponding mean is μ_l)
 - ④ **After repeating 1-3 for each $l \in L$**
Every point's outlier score is
$$o_i = \max_l |x_{l,i} - \mu_l| / \sigma_l$$
2. Fit axis (*robust least trimmed squares*)
 - ① Sort outlier scores
 - ② Take a portion of points in order of increasing outlier score
 - ③ Use PCA to find the cylinder's axis direction

ROBPCA algoritmas nustatyti cilindro ašies kryptį

ROBPCA pagreitinimas naudojant vokselius su svoriais

Šis algoritmas gali būti ženkliai pagreitinamas apribojant jo erdvinę skyrą, dirbant su vokseliais ir taškų kiekiais:

1. Atsitiktinės kryptys L parenkamos ne su atsitiktinėmis taškų poromis, bet su atsitiktiniais vokseliais, kurių ištraukimo tikimybė proporcinga jų taškų kiekiui
2. Visi taškai tame pačiame vokselyje projektuojami vienu veiksmu naudojant bendrą koordinatę
3. Kovariacijos matrica skaičiuojama pagal vokselius, su svoriais lygiais jų vidaus taškų kiekiui

1. Outlier score (*project-pursuit*)
 - ① **Project every point** to random line $l \in L$
 - ② **Sort** by projected coordinate $x_{l,i}$
 - ③ Find a subset of consecutive points with the lowest variance σ_l (corresponding mean is μ_l)
 - ④ **After repeating 1-3 for each $l \in L$**
Every point's outlier score is
$$o_i = \max_l |x_{l,i} - \mu_l| / \sigma_l$$
2. Fit axis (*robust least trimmed squares*)
 - ① Sort outlier scores
 - ② Take a portion of points in order of increasing outlier score
 - ③ Use PCA to find the cylinder's axis direction

ROBPCA algoritmas nustatyti cilindro ašies kryptį

ROBPCA pagreitinimas naudojant vokselius su svoriais

Vilniaus
universitetas

Prieš

```
[1] Start voxel entropy calculation for 76928087 points
[1] Finish voxel entropy calculation, took 4.47181 s
[2] Start filtering point cloud to values below voxel entropy threshold 0.5
[2] Finish filtering, took 0.295381 s and reduced point count 76928087 -> 19006992
[3] Start union find clustering
[3] Finish clustering, took 10.2533 s and found 95950 connected components
[4] Start cluster postprocessing
[4] Finish cluster postprocessing, took 0.867675 s and finalized 96 connected components
[5] Start fitting cylinders
[5] Finish fitting cylinders, took 10.4466
[6] Start reclaiming points
[6] Finish reclaiming points, took 3.04009
```

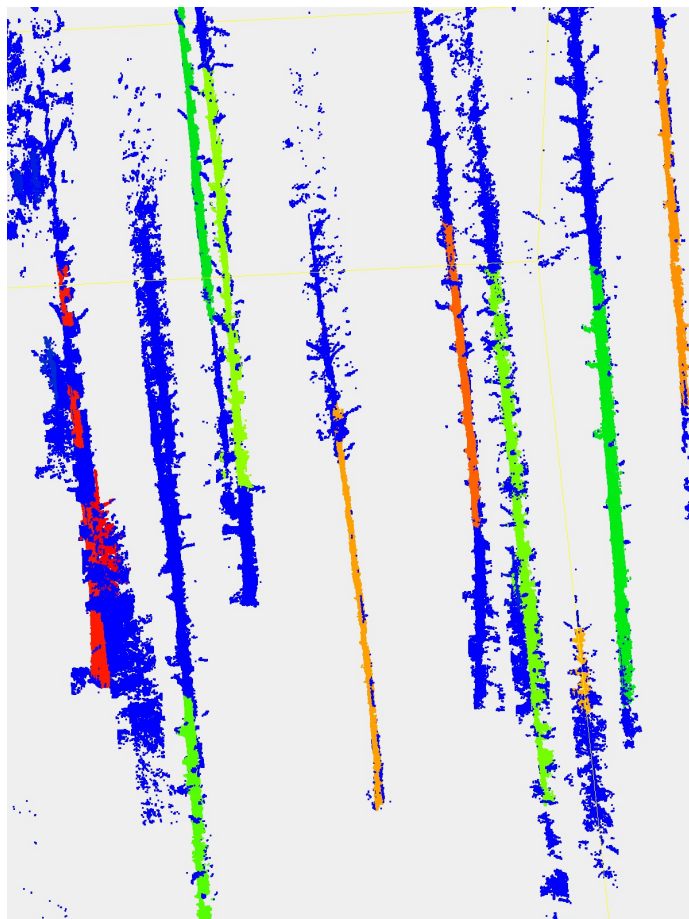
Po – beveik 10
kartų greičiau!

```
[1] Start voxel entropy calculation for 76928087 points
[1] Finish voxel entropy calculation, took 4.49995 s
[2] Start filtering point cloud to values below voxel entropy threshold 0.5
[2] Finish filtering, took 0.297176 s and reduced point count 76928087 -> 19006992
[3] Start union find clustering
[3] Finish clustering, took 10.8743 s and found 95950 connected components
[4] Start cluster postprocessing
[4] Finish cluster postprocessing, took 0.887739 s and finalized 96 connected components
[5] Start fitting cylinders
[5] Finish fitting cylinders, took 1.31005
[6] Start reclaiming points
[6] Finish reclaiming points, took 3.03855
```

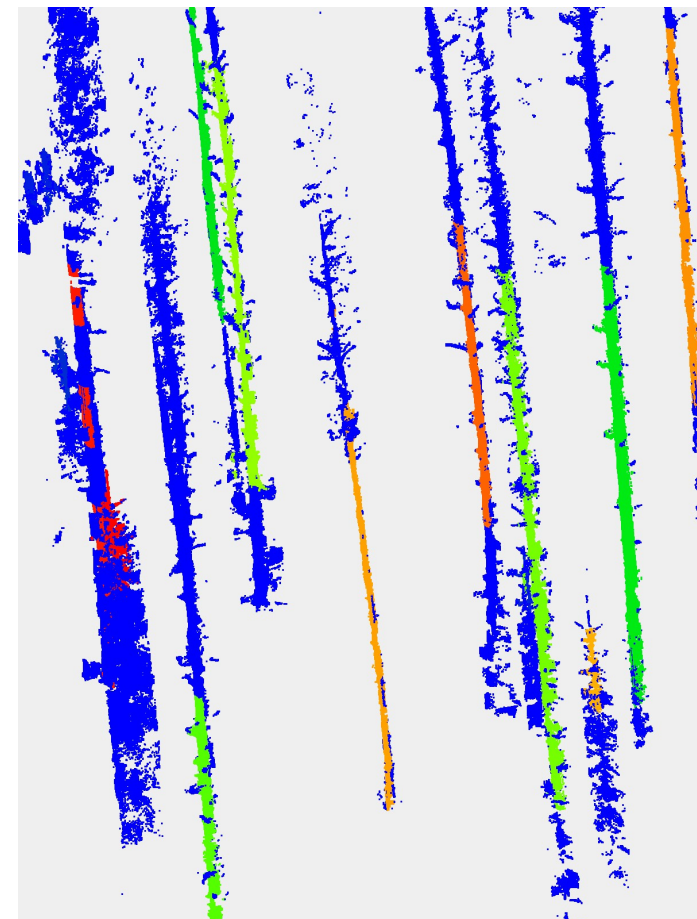
ROBPCA pagreitinimas naudojant vokselius su svoriais

Vilniaus
universitetas

Mėlyni taškai –
filtravimo metu
atmesti, bet cilindro
ašies tinkinimu
sugrąžinti taškai



Originalas



Modifikuotas

Paskutinis metodo patobulinimas

Prieš atmetant aukštos (virš duotos slenkstinės vertės) entropijos regionus, patikrinti ar jie nėra tarp dviejų vienodo ar didesnio ilgio žemos entropijos regionų.

Išsaugomas kamienų tęstinumas be papildomų skaičiavimų, tačiau neatgaunami taškai virš ir po aptiktais taškais (priešingai negu cilindro tinkinimo metodu)

Preliminarūs skaičiavimai rodo, jog TreeTool + VE su nauju filtravimo metodu kartais aplenkia parastą TreeTool *precision* arba *recall* (t.y. sunku iš anksto nuspėti kuris metodas duos geresnį rezultatą, tačiau galima nuspėti, kuris greitesnis).

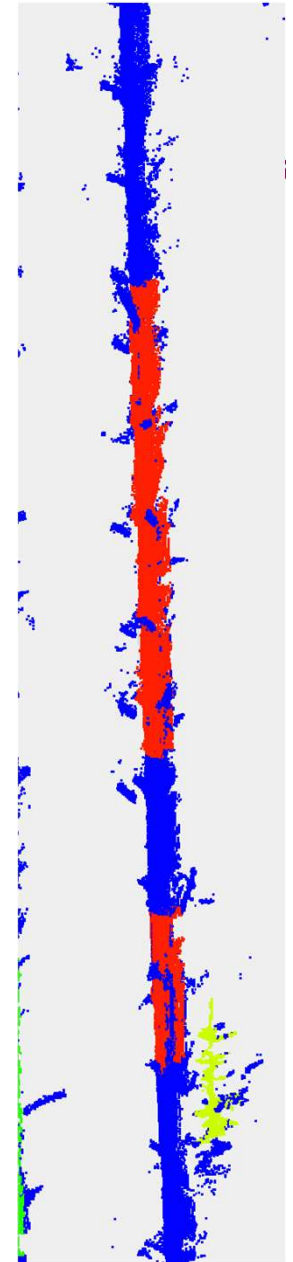
atmesti

Visada
palikti

palikti

Visada
palikti

atmesti



Kito pusmečio darbo planas

1. Atlikti papildomą tyrimą ir paruošti abstraktą bei pranešimą konferencijai *International Symposium of Remote Sensing 2027* (abstraktai pateikiami iki 2027 vasario 28 d.);
2. Pabaigti ruošti publikaciją galutiniam kamienų aptikimo metodui, pateikti WoS indeksuotam žurnalui;
3. Atlikti mašininio mokymosi eksperimentą medžių lokacijų nustatymui.