

Občutljivostna analiza algoritma DBSCAN za segmentacijo podatkov LiDAR

Niko Lukač, Niko Uremović, Monika Ferik Ovčjak, Domen Kavran, Marko Bizjak

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko,
Koroška cesta 46, 2000, Maribor, Slovenia

E-pošta: { niko.lukac@um.si, niko.uremovic@um.si, monika.ferk@um.si, domen.kavran1@um.si, m.bizjak@um.si }

Sensitivity analysis of DBSCAN algorithm for segmentation of LiDAR data

In this paper a sensitivity analysis is presented for applying DBSCAN (Density-Based Spatial Clustering of Applications with Noise) clustering algorithm over LiDAR (Light Detection And Ranging) data. At first, the classified LiDAR point cloud with corresponding features (e.g. rgb, intensity, number of returns etc.) is converted into a topological 2.5D grid, where the segmentation takes place. The segmentation is done by DBSCAN over 7-dimensional feature space, where in the experiments we provide a sensitivity analysis on the choice of input parameters and features' weights. The experiments were conducted on airborne LiDAR point cloud of Maribor with additional color features from true orthophoto, where the choice of optimal feature weights and clustering parameters resulted in a high agreement with the ground truth segmentation data.

1 Uvod

Podatki LiDAR (angl. Light Detection and Ranging) se pridobivajo z laserskim zajemom, ki omogoča merjenje razdalje do opazovanih objektov. Tovrsten zajem je čedalje bolj priljubljen in razširjen zaradi njegove natančnosti, učinkovitosti in široke možnosti uporabe v različnih panogah. Podatki zajema so predstavljeni v obliki nestrukturiranega oblaka točk, ki je pogosto zelo obsežen. Posledično je za izvajanje analize nad podatki ključnega pomena predhodna obdelava točk. Pogost pristop k predobdelavi je segmentacija podatkov LiDAR, s katero razdelimo točke oblaka v segmente na podlagi značilnic, najpogosteje geometrijskih [1, 2, 3, 4]. Na ta način pridobimo skupine sorodnih točk in zmanjšamo število elementov nad katerimi izvajamo nadaljne analize in jih tako učinkoviteje izvedemo. Zaradi količine podatkov najpogosteje uporabimo nenadzorovano segmentacijo, ki samodejno določi in razvrsti točke brez potrebe po ročnem označevanju, kar pospeši in olajša proces ustvarjanja referenčnih podatkov za nadaljnje analize.

Segmentacija nad oblaki točk je ključna pri okoljskih simulacijah [5, 6], kjer s segmentacijo poenostavimo ploskve nad katerimi izvajamo nadaljnje analize. Medtem ko segmentacija na podlagi geometrijskih značilnic služi za osnovno organizacijo in poenostavitev podatkov, t.i. semantična segmentacija doda plast interpretacije, kjer

so točke povezane z določenimi razredi objektov, kar bistveno razširi možnosti uporabe segmentov. Metode segmentacije na podlagi geometrijskih značilnic vključujejo širjenje regij [18], prileganje modelov [13, 14], ali obdelavo grafa, vzpostavljenega nad oblakom točk [17]. Rabani in sodelavci [18] so predstavili metodo za segmentacijo na osnovi širjenja regij, ki temelji na nastavitvi gladkosti. Uporablja lokalne normale med bližnjimi točkami za iskanje gladkih površin v oblakih točk. Algoritem RANSAC je bil široko uporabljen za prileganje geometrijskih primitivov v velikih oblakih točk. Monszpart in sodelavci [13] so uporabili RANSAC za prileganje ravnin za učinkovito segmentacijo ravnih površin. Sun je s sodelavci [14] razvil metodo segmentacije z uporabo grafov za združevanje primitivov, najdenih v notranjih prostorih, s prileganjem primitivov z uporabo RANSAC-a.

Nenadzorovano učenje z algoritmi gručenja predstavlja temelj za pospešitev razvoja učne množice pri globokih nevronskih mrežah [7, 17] za doseganje semantične segmentacije ali t.i. instančne segmentacije, kjer vsaka točka poleg dodeljenega razreda pripada tudi instanci segmenta. Z razvojem globokega učenja je semantična segmentacija oblakov točk vidno napredovala, ki lahko iz oblakov točk pridobijo kompleksne značilnosti [17]. PointNet [7] je bila pionirska arhitektura nevronskih mrež, ki je neposredno obdelovala oblake točk in dodeljevala semantične oznake posameznim točkam. Z uporabo perceptronov in maksimalnega združevanja je PointNet zajemal globalne značilnosti, hkrati pa ohranjal invarianco permutacij. Vendar pa ni izkoriščal lokalnih struktur, kar je privedlo do razvoja PointNet++ [16]. PointNet++ je uvedel hierarhično ekstrakcijo značilnosti z abstraktnimi strukturami, kar je izboljšalo natančnost segmentacije segmentacije v zapletenih scenah.

V članku predstavimo delno avtomatično segmentacijo podatkov LiDAR z uporabo nenadzorovanega učenja preko algoritma gručenja DBSCAN (angl. Density-Based Spatial Clustering of Applications with Noise) [19, 20], ki predstavlja osnovo za gradnjo učne množice za nadaljno obdelavo kot s semantično segmentacijo ali z instančno segmentacijo z globokim učenjem. Nad podatki LiDAR najprej vzpostavimo topološko mrežo značilnic, nad katero apliciramo algoritem DBSCAN z ustreznimi parametri in pripadajočimi utežmi značilnic. Pri tem je izziv določiti najustreznejše parametre, zato predstavimo

občutljivostno analizo izbire parametrov. Dobljene segmente lahko nato obdelujemo dalje. Članek tako predstavlja pomemben doprinos za poenostavitev procesa semantičnega označevanja oblaka točk LiDAR.

Članek je razdeljen v štiri poglavja, kjer v naslednjem poglavju predstavimo uporabljeni metodologijo. V tretjem poglavju podamo rezultate preliminarnih eksperimentov, v zadnjem poglavju pa zaključimo članek.

2 Metodologija

Na vhod prejmemo oblak točk LiDAR P , ki je lahko tudi semantično segmentiran v osnovne razrede (npr. avtomobili, hiše, površje itd.). Vsaka točka LiDAR $p \in P$ je sestavljena iz več značilnic:

$$\{x, y, z, i, r, g, b, R, C\} \in p, \quad (1)$$

kjer $\{x, y, z\}$ definirajo pozicijo, i intenziteto odboja laserskega impulza, $\{r, g, b\}$ morebitno barvo (v primeru da je ortorektificiran aero posnetek poravnani s podatki LiDAR), število odbojev R na laserski impulz ter morebitni razred C . Vse te značilnosti so uporabne za nadaljnjo bolj natančno segmentacijo s pomočjo algoritma DBSCAN. Točke nato hranimo v 2,5 dimenzionalno mrežo G z ločljivostjo res_G [m²]. Z dano mrežo vzpostavimo topologijo. Dana točka p se nahaja v celici $c \in G$, če velja:

$$c_x = p_x - min_x / res_G, \quad (2)$$

$$c_y = p_y - min_y / res_G, \quad (3)$$

kjer min_x in min_y določata minimalno pozicijo po x in y v oblaku točk P . Višina celice c_z je odvisna od najvišje točke v celici:

$$c_z = \max\{p_z^i \in c, p_z^{i+1} \in c, \dots, p_z^m \in c\}, \quad (4)$$

kjer je m število točk, ki se nahajajo v celici c . Najvišje točke upoštevamo zato, da lahko dobimo površje celotnega posnetka LiDAR, pri čemer ne sme biti prisotnega šuma v podatkih. Prav tako 2,5D dimenzionalna lastnost mreže G nam omogoča določanje ene višine za vsako celico. Ko določimo najvišjo točko p v celici c bo le ta pridobila vse značilnice najvišje točke p :

$$\{p_x, p_y, p_z, p_i, p_r, p_g, p_b, p_R, p_C\} \in c, \quad (5)$$

Zaradi vrzeli in šuma v podatkih LiDAR lahko imamo določene celice prazne, brez vsebnosti točk. V tem primeru lahko uporabimo interpolacijo IDW (angl. inverse distance weighting), kjer je višina celice c odvisna od q sosednjih nepraznih celic:

$$c_z = \frac{\sum_{j=1}^q \left(\frac{c_z^j}{d(c^i, c^j)^\omega} \right)}{\sum_{j=1}^q \left(\frac{1}{d(c^i, c^j)^\omega} \right)}, \quad (6)$$

kjer c^j so sosednje neprazne celice, d pa funkcija evklidske razdalje med dvema celicama. IDW se inicializira nad robnimi celicami vrzeli in se iterativno razširi. Prav tako IDW izvedemo za vsak razred C posebej po

uporabniško nastavljenem prioritetnem vrstnem redu (npr. najprej celice površja nato celice stavb), sicer bi lahko prišlo do nezaželenega mešanja razredov. Parameter ω določa vpliv razdalje na težo. Poleg višine se hkrati interpolirajo postale značilnice celic.

2.1 Segmentacija z DBSCAN

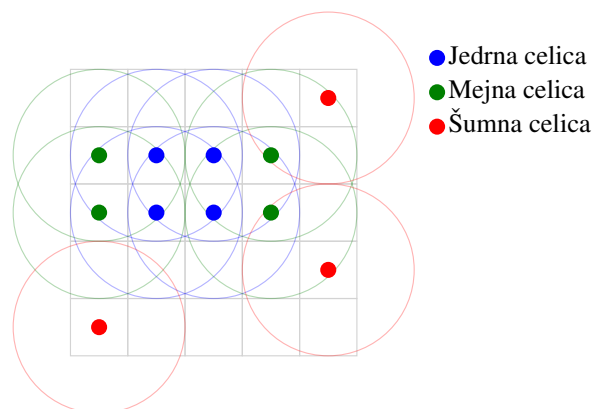
Algoritem DBSCAN izvedemo nad mrežo celic G . V algoritmu DBSCAN združevanje celic v gruče temelji na konceptu relativne gostote. Algoritem definira dve ključni merili za združevanje celic, in sicer jedrnost celice in dosegljivost celice znotraj določene razdalje. Dana celica c^i je jedrna, če velja:

$$|N_\epsilon(c^i)| \geq minPts; N_\epsilon(c^i) = \{c^j \in G | d(c^i, c^j) \leq \epsilon\} \quad (7)$$

kjer $N_\epsilon(c^i)$ predstavlja relativno ϵ -okolico celice c^i , $minPts$ pa minimalno število točk v tej okolici. Če celica c^i izpolnjuje pogoj relativne gostote glede na $minPts$, se postavi kot jedro nove gruče ali pa se združi z obstoječo gručo. Združevanje celic v gručo oz. segment se nadaljuje z razširitvijo na sosednje celice $N_\epsilon(c^i)$, ki so direktno dosegljive. Propagacija se nato izvede na vse dosegljive celice, kjer velja da je c^j dosegljiva iz c^i , če obstaja zaporedje celic $c^k, c^{k+1}, \dots, c^{k+n}$ za katerega velja:

$$c^i \rightarrow c^k \rightarrow c^{k+1} \rightarrow \dots \rightarrow c^{k+n} \rightarrow c^j, \quad (8)$$

kjer je vsaka celica c^{k+1} direktno dosegljiva iz prejšnje celice c^k , torej $d(c^k, c^{k+1}) \leq \epsilon$. Tako je združevanje celic v algoritmu DBSCAN osredotočeno na identifikacijo in širjenje jedrnih točk, ki imajo zadostno število sosednjih točk znotraj predpisane razdalje ϵ . V nasprotnem primeru se smatra, da celica ne pripada nobenemu segmentu ter je klasificirana kot šum. Na sliki 1 je prikazan primer jedrnih, mejnih in šumnih celic.



Slika 1: Primer DBSCAN jedrnih, mejnic in šumnih celic nad mrežo G z ločljivostjo 1 m² posamezne celice. Uporabljeni parametra sta $minPts = 6$ in $\epsilon = 1,5$ m.

Pri računanju ϵ -okolice posamezne celice upoštevamo uteženo evklidsko razdaljo, kjer je razdalja med dvema celicama odvisna od n-dimenzionalnih značilnic posamezne celice ter utežeh vezanih na posamezno značilnico.

Pred izračunom se normalizirajo vrednosti značilnic celic z uporabo normalizacije minmax, tako so vse vrednosti v razponu [0, 1]. Izračun kvadrata razdalje je tako definiran kot:

$$\begin{aligned}
 dw(c^i, c^j) = & ((c_x^i - c_x^j)^2 + (c_y^i - c_y^j)^2 + (c_z^i - c_z^j)^2) \\
 & + w_i(c_i^i - c_i^j)^2 \\
 & + w_{rgb}((c_r^i - c_r^j)^2 + (c_g^i - c_g^j)^2 + (c_b^i - c_b^j)^2) \\
 & + w_R(c_R^i - c_R^j)^2 \\
 & + w_C(c_C^i - c_C^j)^2
 \end{aligned} \quad (9)$$

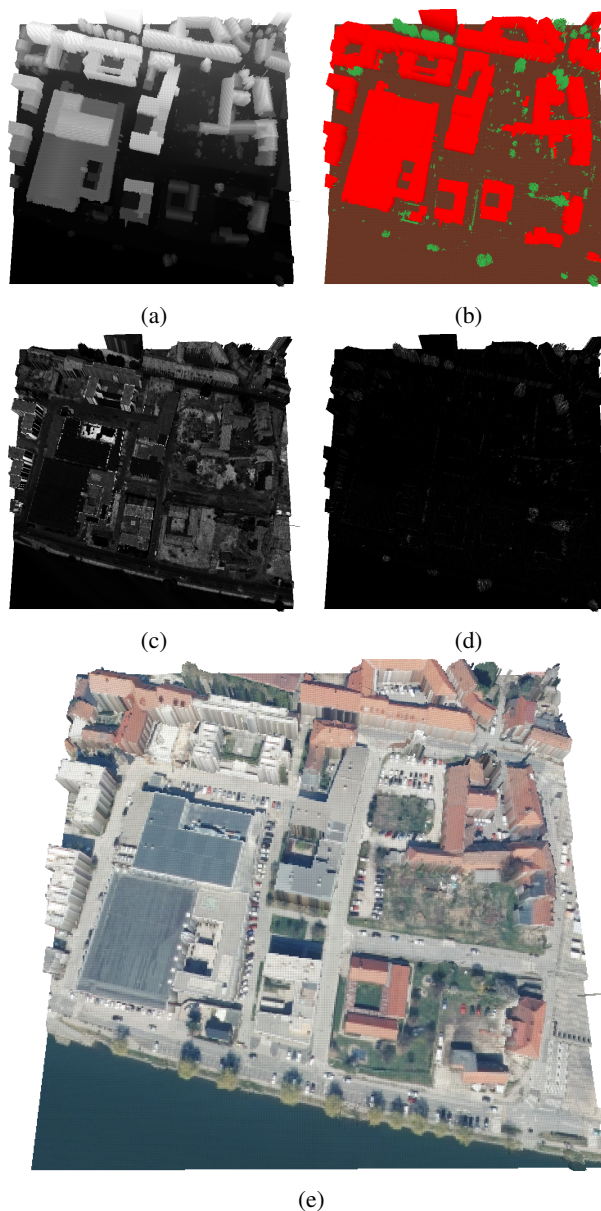
kjer so uteži za posamezne značilnice definirane z w_i , w_{rgb} , w_R in w_C . V naslednjem poglavju bomo predstavili rezultate občutljivostne analize segmentacije z algoritmom DBSCAN s spreminjanjem vrednosti danih uteži.

3 Rezultati

Segmentacijo smo izvedli na oblaku točk zračnega snemanja LiDAR za del mesta Maribor. Nad tem območjem smo konstruirali mrežo G z ločljivostjo 1 m^2 na celico (Slika 2a). Oblak točk je že semantično klasificiran v tri osnovne razrede (Slika 2b): površje, stavbe ter srednja in visoka vegetacija. IDW smo najprej izvedli nad celicami površja, nato nad stavbami. Visoke vegetacije nismo upoštevali v interpolaciji, saj semantično ni smiselno imeti vegetacijo čez celotno mrežo. Vsebuje tudi informacije o intenziteti laserskega odboja (Slika 2c) in številu odbojev na posamezni impulz, ki je zadel točko na površju (Slika 2d). Kot dodatno značilnico smo mrežo G obogatili z ortofotom, ki smo ga projicirali v pravilno ortogonalno projekcijo, kjer so objekti videni pod kotom 90° (Slika 2e).

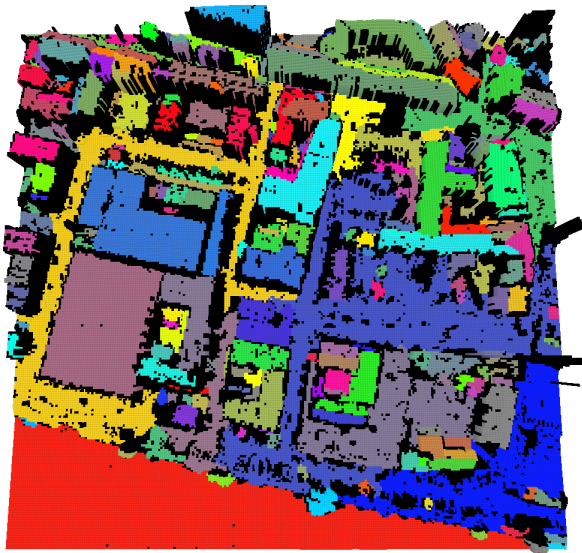
Na sliki 3 je prikazana pravilna segmentacija mreže na 243 enoličnih segmentov, ki jim lahko določimo semantični razred (npr. asfalt, trava, opeka itd.). Dano segmentacijo smo pridobili z algoritmom DBSCAN in ročnimi popravki. V nadaljevanju članka bomo predstavili rezultate občutljivostne analize algoritma DBSCAN, za doseganje segmentacije na sliki 3. Za ta namen smo spreminjali uteži w ter parametra ϵ in minPts , pri čemer smo uporabili mrežno iskanje (angl. grid search). Algoritem DBSCAN je relativno hiter in dosega v najslabšem logaritemsko časovno zahtevnost glede na število točk. Prav tako zahtevamo da so celice v segmentu tudi prostorsko strogo povezane komponente (to pomeni, da ni možno imeti segmenta iz celic na dveh koncih podatkov).

Rezultati občutljivostne analize so prikazani v tabeli 1, kjer je prikazanih 20 najuspešnejših segmentacij na podlagi indeksov Rand, prilagojen Rand (angl. adjusted Rand) ter normalizirane skupne informacije (angl. normalized mutual information) [21]. Indeks Rand meri podobnost med dvema segmentacijama glede na ujemanje se pare elementov. Ne upošteva naključnih ujemanj, kar lahko vodi do precenjene podobnosti. Prilagojen Rand indeks je izboljšana različica indeksa Rand, saj upošteva možnost naključnega ujemanja. Normalizirana skupna informacija meri količino skupne informacije med segmentacijami, prilagojeno glede na njihovo velikost. Zagota-



Slika 2: Podatki zračnega snemanja LiDAR vstavljeni v 2,5D mrežo G , kjer je (a) osnovni višinski prikaz (angl. heightmap), (b) osnovna semantična segmentacija, (c) intenziteta odbojev, (d) število odbojev na žarek, (e) poravnan pravi-ortofoto nad mrežo

vlja merilo za primerjavo segmentacij, ki upošteva kompleksnost in velikost skupin. Parameter w_c smo nastavili na vrednost 0.05, saj dajemo močni poudarek na osnovne razrede LiDAR. Pričakovano dobimo dobre rezultate z nizko vrednostjo w_{rgb} in w_i , saj s tem zagotovimo večjo razlikovanje po barvah in intenziteti, ki izhajajo posledično iz različnih materialov površja. V tabeli 2 so predstavljeni dodatni rezultati, če upoštevamo različne kombinacije vhodnih značilnic nad najboljšo izbiro ϵ in minPts iz tabele 1. Značilnica, ki ni upoštevana ima, utež 0. V primeru neupoštevanja vseh značilnic se pri segmentaciji upošteva le pozicija posamezne celice. Iz rezultatov je razvidno, da povečanja števila značilnic omogoča višjo diskriminacijo med segmenti in posledično bolj natančno



Slika 3: Pravilna segmentacija 2,5D mreže G na 243 segmentov. Barve so unikatne in določene naključno. Črna barva pripada šumnim celicam.

Tabela 1: Rezultati občutljivostne analize, parameter $M = \minPts$. RI je indeks Rand, ARI je prilagojen indeks Rand, ter NMI je normalizirana skupna informacija.

ϵ	w_i	w_{rgb}	w_R	M	RI	ARI	NMI
2,00	0,20	0,10	0,05	3	0,95	0,85	0,83
2,00	0,20	0,10	0,10	3	0,95	0,85	0,83
2,00	0,20	0,10	0,20	3	0,95	0,85	0,83
2,00	0,20	0,10	0,30	3	0,95	0,85	0,83
2,00	0,20	0,10	0,50	3	0,95	0,85	0,83
1,75	0,10	0,20	0,05	3	0,94	0,84	0,83
1,75	0,10	0,20	0,10	3	0,94	0,84	0,83
1,75	0,10	0,20	0,20	3	0,94	0,84	0,83
1,75	0,10	0,20	0,30	3	0,94	0,84	0,83
1,75	0,10	0,20	0,50	3	0,94	0,84	0,83
2,00	0,10	0,20	0,05	4	0,94	0,85	0,82
2,00	0,10	0,20	0,10	4	0,94	0,85	0,82
2,00	0,10	0,20	0,20	4	0,94	0,85	0,82
2,00	0,10	0,20	0,30	4	0,94	0,85	0,82
2,00	0,10	0,20	0,50	4	0,94	0,85	0,82
1,75	0,20	0,10	0,05	3	0,95	0,85	0,82
1,75	0,20	0,10	0,10	3	0,95	0,85	0,82
1,75	0,20	0,10	0,20	3	0,95	0,85	0,82

Tabela 2: Rezultati občutljivostne analize pri različni kombinaciji vhodnih značilnic pri $\epsilon = 2$ in $M = 3$.

w_i	w_{rgb}	w_R	RI	ARI	NMI
0,20	0,10	0,05	0,95	0,85	0,83
0,20	0,10	0	0,86	0,81	0,79
0	0,10	0,05	0,79	0,78	0,70
0,20	0	0,05	0,76	0,79	0,67
0,20	0	0	0,75	0,78	0,68
0	0,10	0	0,71	0,78	0,65
0	0	0,05	0,63	0,78	0,58
0	0	0	0,60	0,79	0,57

segmentacijo. Prav tako je razvidno, da je pomembnost značilnic v naslednjem vrstnem redu: intenziteta, barva, ter število odbojev na pulz.

4 Zaključek

V članku smo predstavili učinkovito uporabo algoritma gručenja DBSCAN, kjer so bili eksperimenti izbire parametrov algoritma izvedeni nad oblakom točk LiDAR dela mesta Maribor. S pravilno izbiro parametrov in uteži značilnic je segmentacija dosegla približno 85 % ujemanje (po prilagojenem Rand indeksu) s pravilno segmentacijo. Čeprav to ujemanje ni popolno, hitrost algoritma DBSCAN omogoča hitro segmentacijo za nadaljnjo obdelavo z npr. algoritmi semantične segmentacije temelječih na globokih mrežah. Tako algoritem DBSCAN, kljub svoji starosti, ostaja uporabno orodje pri predobdelavi podatkov LiDAR.

Nadaljno delo predstavlja uporabe danega pristopa za polavtomatski pristop generiranja učnih množic semantične segmentacije zračnega snemanja za učenje globokih nevronske mreže. S tem se bo pripravilo bolj napredne značilnice površja pri algoritmih okoljskih simulacij (npr. izračun temperature površja v odvisnosti od značilnosti površja).

Zahvala

Raziskovalno delo je bilo sofinancirano s strani Javne agencije za znanstvenoraziskovalno in inovacijsko dejavnost Republike Slovenije, v okviru temeljnega projekta št. J7-50095 in raziskovalnega programa št. P2-0041. Za podatke LiDAR se zahvaljujemo Agenciji Republike Slovenije za okolje (ARSO). Za podatke pravih ortofotofov mesta Maribor se zahvaljujemo Geodetskemu inštitutu Slovenije.

Literatura

- [1] M. Bizjak, B. Žalik, and N. Lukač, "Parameter-Free Half-Spaces Based 3D Building Reconstruction Using Ground and Segmented Building Points from Airborne LiDAR Data with 2D Outlines," *Remote Sensing*, vol. 13, no. 21, p. 4430, 2021.
- [2] Mirceta, K., Bohak, C., Kim, B.H., Kim, M.Y. and Marolt, M., 2018, September. Drone segmentation and tracking in grounded sensor scanned LiDAR datasets. In Proceedings of the Zbornik sedemindvajsete mednarodne Elektrotehniške in računalniške konference ERK (pp. 384-387).
- [3] Kim, B.H., Khan, D., Bohak, C., Choi, W., Lee, H.J. and Kim, M.Y., 2018. V-RBNN based small drone detection in augmented datasets for 3D LADAR system. *Sensors*, 18(11), p.3825.
- [4] Bohak, C., Slemenik, M., Kordež, J. and Marolt, M., 2020. Aerial LiDAR data augmentation for direct point-cloud visualisation. *Sensors*, 20(7), p.2089.
- [5] N. Lukač, D. Mongus, B. Žalik, G. Štumberger, and M. Bizjak, "Novel GPU-accelerated high-resolution solar potential estimation in urban areas by using a modified diffuse irradiance model," *Applied Energy*, vol. 353, p. 122129, 2024.
- [6] M. Bizjak, B. Žalik, G. Štumberger, and N. Lukač, "Large-scale estimation of buildings' thermal load using LiDAR data," *Energy and Buildings*, vol. 231, p. 110626, 2021.
- [7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jul 2017.

- [8] O. Matuschek and Others, "Identifying Repeated Objects from Multi-room Indoor Scanning Data," *Journal of Vision and Computer Graphics*, 2014.
- [9] X. Hu and Others, "Dynamic Region Growing for Surface Patches Segmentation," *Journal of Computer Vision*, 2012.
- [10] L. Nan and Others, "Search-Classify Pipeline for Scene Modeling," *Journal of Graphics and CAD*, 2012.
- [11] B. Li and Others, "Object-Retrieved Approach for Scene Reconstruction," *Journal of Visual Computing*, 2015.
- [12] J. Shi and Others, "Classifiers for Objects and Object Groups," *Journal of Artificial Intelligence Research*, 2016.
- [13] Á. Monszpart and Others, "RANSAC-Based Plane Fitting for Large-Scale Indoor Point Clouds," *Journal of Computer Graphics and Applications*, 2015.
- [14] L. Sun and Others, "Graph-Cut Segmentation for Primitive Grouping," *Journal of Computer Vision and Image Understanding*, 2018.
- [15] S. Yu and Others, "Patch Relation Classifier for Instance Segmentation," *Journal of Pattern Recognition*, 2019.
- [16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," *Advances in Neural Information Processing Systems*, pp. 5099–5108, 2017.
- [17] Y. Sun, X. Zhang, and Y. Miao, "A Review of Point Cloud Segmentation for Understanding 3D Indoor Scenes," *Visual Intelligence*, vol. 2, no. 14, 2024.
- [18] T. Rabbani, F. den Heuvel, and G. Vosselman, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.
- [19] K. Khan, S. U. Rehman, K. Aziz, S. Fong, in S. Sarasvady, "DBSCAN: Past, present and future," in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, str. 232–238, 2014.
- [20] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, vol. 96, no. 34, str. 226–231, 1996.
- [21] A. Amelio, C. Pizzuti: Correction for closeness: Adjusting normalized mutual information measure for clustering comparison. *Computational Intelligence*, 33 (3), str. 579-601, 2017.