

Samorazvijajoča se segmentacija globinske slike stereo kamere za notranja strukturirana okolja

Miloš Antić, Andrej Zdešar, Igor Škrjanc

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: {milos.antic, andrej.zdesar, igor.skrjanc}@fe.uni-lj.si

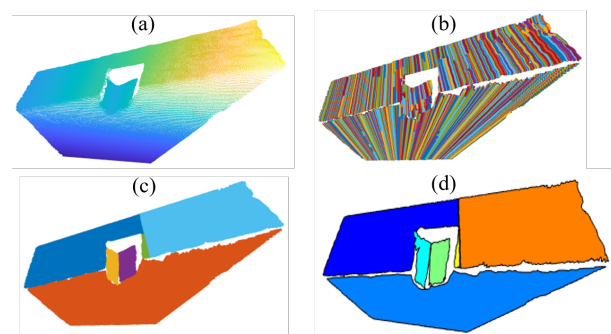
Evolving segmentation of a stereo-camera depth map for structured indoor environments

In this paper, we present an adaptive approach to point cloud segmentation based on the EPCC (Evolving Principal Component Clustering) method. The method exploits the properties of the ordered data and their neighbor relations, which are the result of the way the data of certain sensors are captured. Adaptability refers to the use of a recursive estimate of the parameters of linear prototypes, which are used to describe clusters of similar data. The main contribution of this work is the extension and application of the EPCC method in 3D space for recursive and real-time prototype detection, such as lines and planes describing similar clusters. The developed algorithm was compared with known methods for point cloud segmentation, such as RANSAC and region growing.

1 Uvod

Na področju 3D-segmentacije oblakov točk je bil dosežen velik napredek. Eden glavnih dejavnikov hitrega razvoja tega področja je dostopnost nizkocenovnih senzorjev, kot so laserski merilniki razdalje, stereo kamere itd. Pojavila se je visoka potreba po 3D-razumevanju okolja, oz. rekonstrukciji in klasifikaciji tridimenzionalnih nizov točk v robotiki, za namene lokalizacije robotskih ali mobilnih sistemov in avtonomne gradnje zemljevidov oz. SLAM [1], [2], ki je lahko podlaga za planiranje poti ali sledenju trajektorijam [3], [4]. Velik pomen ima tudi 3D-modeliranje objektov in klasifikacija le-teh, kar je lahko uporabno v aplikacijah kot so manipulacija objektov z robotskimi rokami [5] in hoja četveronožnega robota po stopnicah [6].

Veliko truda je bilo vložena v razvoj algoritmov za detekcijo ravnin v oblaku točk. Z detekcijo in predstavitvijo oblaka točk z modeli ravnin ali planarnimi območji dosežemo bolj kompaktno predstavitev tridimenzionalnega prostora (slika 1). Predlagana je bila 3D-segmentacija v zunanjih [7] in notranjih okoljih [8]. Notranja okolja so posebej prikladna za planarno segmentacijo oblaka točk, saj se v takih okoljih nahaja ogromno predmetov in površin, ki jih je naredil človek in so primerna za planarno rekonstrukcijo. Notranji prostori, katere lahko opišemo z ravninami ali pa z območji točk, ki jih pridobimo s pomočjo modelov prototipa ravnin, so torej bolj



Slika 1: Prikaz oblaka točk (a). Z metodo EPCC naredimo segmentacijo oblaka na daljice (b), kar uporabimo v naslednjem koraku pri določevanju ravnih površin (c). Rekonstrukcija prizora z ravnimi površinami (d).

strukturirani.

Grilli in drugi so v [9] naredili analizo metod in pristopov, ki se uporabljajo pri segmentaciji oblaka točk. Izpostavili so, da je 3D-segmentacija oblaka točk zahtevna naloga, saj imajo oblaki točk običajno neenakomerno gostoto vzorčenja točk, so pogosto neurejeni in vsebujejo veliko šuma. Ena izmed prednosti našega sistema za obdelavo oblaka točk je v naravi pridobivanja podatkov z globinskega senzorja, ki na izhodu vrača matriko urejenih točk, kar omogoča bolj intuitivno, lažjo in hitrejšo implementacijo algoritma za ekstrakcijo ravnin oz. planarnih območij oblaka točk. Še vedno pa ostaja problem neenakomerne gostote točk in prisotnosti šuma, ki se pri meritvi globine večja z razdaljo.

Evolucijsko računanje pridobiva, na področju računalniške inteligence in strojnega učenja, na pomenu, saj izkazuje velik potencial za uporabo v stvarnem času in sistemih ter okoljih, ki se skozi čas spreminjajo [10]. Veliko metod na omenjenem področju temelji na predpostavki, da imamo na voljo velik nabor zgodovinskih podatkov, ki jih uporabijo za generacijo modelov pri regresiji ali identifikaciji. Avtorji v [10] so izpostavili, da je ta predpostavka v realnih aplikacijah omajana, saj je v teh sistemih nabor prejšnjih podatkov lahko nezadosten, pri čemer imamo običajno opravka z dinamičnem okoljem ali sistemom. Poudarili so tudi neučinkovitost iterativnih algoritmov pri obdelavi naraščajočih podatkov, saj običajno zahtevajo

več prehodov čez enake dele podatkov. Opisano problematiko naslavlja področje samorazvijajočega se modeliranja [11–14], ki se je pokazalo za nov okvir, kjer obdelavo podatkov iz pretoka podatkov rešuje s pristopom samorazvoja (*angl. self-adaption*), učenjem z enim prehodom čez podatke in evolucijo, kot tudi kontrakcijo, parametrov modelov na zahtevo in spotoma.

V tem prispevku predstavljamo razširitev metode EPCC [15] v 3D-prostor za detekcijo ravnih površin v oblaku točk. Segmentacija poteka na podatkih z globinske kamere, kjer smo 3D-predstavitev okolja pridobili s triangulacijo razdalj. Modificirana rekurzivna metoda EPCC ekstrahira daljice s prototipi premic, kjer je glavni poudarek na rekurzivni oceni parametrov le-teh, s čimer pohitrimo algoritem. Algoritem je dodatno pohitren z iskanjem daljic v 2D-projekcijah oblaka točk z eliminacijo ene izmed koordinat. Daljice uporabimo za ekstrakcijo planarnih območij oz. prototipov ravnin. Ker je oblak točk urejen, smo omenjeni metodi implementirali v algoritmu tako, da sproti odkriva daljice in ekstrahira prototipe ravnin. To pomeni, da lahko niz podatkov obdelamo v enem koraku. Prednost uporabljenega algoritma je tudi možnost upoštevanja urejenosti podatkov oblaka točk, kar pohitri in olajša razvoj metode za segmentacijo.

Preostanek prispevka ima naslednjo strukturo. Poglavlje 2 podaja kratek pregled sorodnih del. Poglavlje 3 opisuje ozadje planarne segmentacije z metodo EPCC. V poglavju 5 so prikazani rezultati eksperimentov na bazi podatkov, ki so prikazani v poglavju 4. V poglavju 6 so predstavljeni zaključki ob diskusiji rezultatov.

2 Sorodna dela

Prispevek [9] opisuje metodologije, ki se uporabljajo za segmentacijo oblaka točk. V našem prispevku smo se osredotočili na literaturo, ki je povezana z našim delom in izpostavili tri pogosto uporabljene metode za detekcijo ravnin v oblaku točk, in sicer pristope, ki temeljijo na metodi RANSAC (*angl. RANdom SAmple Consensus*), metodi rastočega območja (*angl. region growing*) in Houghovi transformaciji.

2.1 Pristopi z metodo RANSAC

Metodo RANSAC sta predstavila Fischler in Bolles v [16] za gradnjo modela oz. oceno parametrov le-tega iz eksperimentalnih podatkov.

Xu in drugi so v [8] predstavili planarno segmentacijo, kjer so z naučenimi SVM-modeli ter SVM-predikcijami klasificirali točke oblaka v različne ravninske površine in rezultate primerjali z navadno in NDT-RANSAC metodo.

Poz in Ywata sta v [17] predstavila adaptivni pristop k segmentaciji streh zgradb, ki vključuje tako predprocesiranje oblaka točk za ločevanje točk pripadajočim zgradbam, kot tudi segmentacijo ravnin z metodo RANSAC in postopek agregacije nadsegmentiranih območij.

Nguyen in drugi so v [18] predstavili segmentacijo delno urejenih MLS-oblakov točk, kjer najprej segmentirajo profile skeniranja z laserskim merilnikom na podlagi smernih vektorjev, nato te profile grupirajo in detektirajo

ravnine glede na planarne vrednosti različnih sosedskih profilov skeniranja.

2.2 Pristopi s Houghovo transformacijo

Houghova transformacija, ki jo je prvič predstavil Hough v [19], je poleg metode RANSAC standardna metoda za detekcijo parametrov modelov ravnin ali krivulj v 2D- ali 3D-prostoru.

Vera in drugi so v [20] predstavili detekcijo ravnin v globinskih slikah za delovanje v realnem času, ki uporablja implicitno 'quadtree' drevesno strukturo za identifikacijo rojev približno koplanarnih točk v 2,5-D prostoru.

Khanh in drugi so v [21] predstavili izboljšavo segmentacije tal v oblaku točk za mobilne robote, ki temelji na metodi RHT (*angl. Randomized Hough Transform*) v kombinaciji z omejitvami razdalj ter kotov in jo primerjali z metodo RANSAC.

Tian in drugi so v [7] predstavila novo metodo za segmentacijo planarnih lastnosti v neurejenih oblakih točk na podlagi ekstrahiranih daljic z uporabo 2D Houghove transformacije in drevesne strukture (*angl. octree*).

2.3 Pristopi z metodo rastočega območja

Prednost metode rastočega območja je, da izkorišča lastnosti urejenih oblakov točk in sosedske strukture točk.

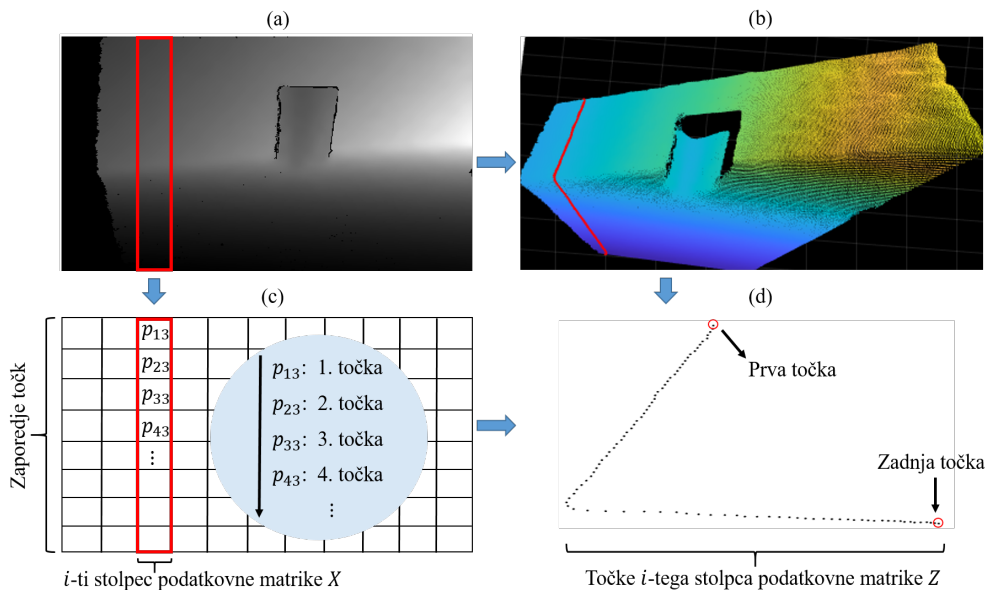
Dong in drugi so v [22] predstavili novo hibridno metodo rastočega območja, kjer so problem segmentacije oblaka točk predstavili kot robustno globalno energijsko optimizacijo. Konvergenca algoritma oz. energijske funkcije so zagotovili s pristopom simuliranega ohlajanja.

Wu in drugi so v [23] predstavili planarno segmentacijo TLS-oblakov točk z metodo MSTVM za boljše določitev točke, ki predstavlja seme algoritma. Predstavili so novo lastnost — t. i. indikator planarne jakosti (*angl. plane strength indicator*) — za bolj intuitivno določitev točke semena.

Huang in drugi so v [24] predstavili algoritem EVBS (*angl. encoding voxel-based segmentation*), ki temelji na metodi rastočega območja s preiskovanjem strukture kvadrov (*angl. voxels*) in upoštevanju omejitev zveznosti ter gladkosti.

3 Detekcija daljic

Slika 2 prikazuje način pojavljanja podatkov v oblaku točk oz. globinski sliki. Algoritem za detekcijo daljic izkorišča vzorec zaporednosti točk in naravo rekurzivne obdelave podatkov metode EPCC. Struktura podatkov in narava zajemanja točk prostora znotraj (stožčastega) vidnega kota kamere omogoča ekstrakcijo daljic v 2D-projekcijah oblaka točk oz. podatkovnih matrikah X . Prostor iskanja daljic je razdeljen na tri dele (osrednje območje in dva zunanja) glede na horizontalni vidni kot α globinske kamere, ki znaša približno 90° . Pri delitvi prostora upoštevamo koordinatni sistem kamere, kjer os z kaže iz kamere, x -os je v horizontalni smeri in y -os je v vertikalni smeri. 3D-točke matrike Z , ki se nahajajo v osrednjem območju oz. za α na intervalu $-30^\circ - 30^\circ$, projiciramo na yz -ravnino. Točke, ki se nahajajo v zunanjema



Slika 2: Globinska slika (a) in pripadajoč oblak točk (b). Pod (c) je prikazana matrika globinske slike, kjer so točke po posameznih stolpcih urejene po zaporedju od prve do zadnje vrstice. Prikaz točk označenega stolpca 3D-matrike oz. oblaka točk (d).

območjema oz. za kote, ki so po absolutni vrednosti večji od $\pm 30^\circ$, pa projiciramo na xy -ravnino.

V vsaki projekciji poteka rojenje točk z rekurzivno oceno statističnih lastnosti 2D-podatkov in parametrov modelov premic, ki so določeni direktno iz kovariančne matrike podatkov Σ_j . j -ti model premice opisuje normalni oz. lastni vektor \mathbf{p}_j , ki pripada najmanjši lastni vrednosti matrike Σ_j

$$\lambda_j = \mathbf{p}_j^T \Sigma_j \mathbf{p}_j \quad (1)$$

Lastni vektor j -tega roja je določen iz Σ_j kot

$$\mathbf{p}_j^T = \begin{bmatrix} -\frac{\theta_1}{\sqrt{\theta_1^2+1}} & \frac{1}{\sqrt{\theta_1^2+1}} \end{bmatrix} \quad (2)$$

kjer je

$$\theta_1 = \frac{\sigma_{12}}{\sigma_{11}} \quad (3)$$

in kjer so $\sigma_{io}, i, o \in \{1, 2\}$ elementi matrike Σ_j . Pseudokoda detekcije daljic je predstavljena v Algoritmu 1. Na začetku algoritma je potrebno nastaviti parametre k_{min} , σ_{init} in n_{buf} . k_{min} pomeni minimalno število točk, ki jih potrebujemo za zanesljivo gradnjo novega prototipa roja, σ_{init} pa začetna varianca razdalje roja. S parametrom n_{buf} nastavimo največje število točk, ki jih še obravnavamo v postopku odkrivanja rojev. Algoritem roji podatke po stolpcih podatkovne matrike Z tako, da inicializira nov roj in ga postavi na aktivni seznam ter ga širi glede na določen kriterij. Širjenje roja je ustavljeno, ko podatek ne izpolnjuje kriterija in je odstranjen z aktivnega seznama. Tako širimo vedno tiste roje, ki so bili nazadnje odkriti. Kriterij za dodajanje v aktivni oz. j -ti roj je odvisen od ortogonalne razdalje do j -tega modela premice

$$d_j(k) = |(\mathbf{z}(k) - \boldsymbol{\mu}_j)^T \cdot \mathbf{p}_j| \quad (4)$$

Algoritem 1 Odkrivanje daljic z metodo EPCC

```

1: Definicija parametrov  $k_{min}, k_{max}, n_{buf}=3k_{min},$ 
    $\sigma_{init}, n$  (število podatkov matrike  $Z$ ).
2: for  $k = 1 : n$ 
3:   switch state
4:     case 0
5:        $\mathbf{z}(k)$  shranjujemo v medpomnilnik  $buf$ .
6:       if  $\text{length}(buf) \geq k_{min}$ 
7:         Poiščemo kandidata za nov roj v  $buf$  in
           ocenimo  $\boldsymbol{\mu}_j, \Sigma_j, \mathbf{p}_j$  in  $d_j$  naslednje točke
           v  $buf$  ter postavimo  $\sigma_j = \sigma_{init}$ .
8:         if  $d_j < k_{max} \sqrt{\sigma_j}$ 
9:           Ustvarimo roj in posodobimo  $\boldsymbol{\mu}_j,$ 
            $\Sigma_j, \mathbf{p}_j$  in  $\sigma_j$ . Postavimo  $state = 1$ 
           in počistimo medpomnilnik  $buf$ .
10:        end
11:      end
12:      if  $\text{length}(buf) \geq n_{buf}$ 
13:        Odstranimo najstarejši podatek iz  $buf$ .
14:      end
15:    case 1
16:      Izračunamo  $d_j(k)$  vzorca  $\mathbf{z}(k)$ .
17:      if  $d_j(k) \leq k_{max} \sqrt{\sigma_j}$ 
18:        Dodamo  $\mathbf{z}(k)$  v  $j$ -ti roj in posodobimo
            $\boldsymbol{\mu}_j, \Sigma_j, \mathbf{p}_j$  in  $\sigma_j$ .
19:      else
20:         $\mathbf{z}(k)$  damo v  $buf$  in postavimo  $state = 0$ 
21:      end
22:    end
23: end

```

kjer je $\mathbf{z}(k)$ točka novega vzorca in $\boldsymbol{\mu}_j$ povprečje j -tega roja. V zajetih podatkih je prisoten šum, ki se veča s kvadratom razdalje. Da omogočimo robustno rojenje podat-

kov, je potrebno upoštevati prisotnost šuma. V ta namen vpeljemo v kriterij rojenja (5) definicijo variance ortogonalnih razdalj vseh točk v j -tem roju σ_j

$$d_j(k) < k_{max} \sqrt{\sigma_j} \quad (5)$$

kjer je k_{max} pozitivna konstanta, ki določa občutljivost kriterija rojenja v primeru normalne porazdelitve šuma.

Vsaki širitvi roja z novim vzorcem, sledi posodobitev normalnega vektorja in statističnih lastnosti po rekurzivnih enačbah (6)

$$\begin{aligned} \sigma_j(k_j) &= \sigma_j(k_j - 1) \frac{k_j - 2}{k_j - 1} + \frac{1}{k_j} d_j(k)^2 \\ \boldsymbol{\mu}_j(k_j) &= \frac{k_j - 1}{k_j} \boldsymbol{\mu}_j(k_j - 1) + \frac{1}{k_j} \mathbf{z}(k_j) \\ \boldsymbol{\Sigma}_j(k_j) &= \frac{k_j - 2}{k_j - 1} \boldsymbol{\Sigma}_j(k_j - 1) \\ &+ \frac{1}{k_j} (\mathbf{z}(k_j) - \boldsymbol{\mu}_j(k_j - 1)) (\mathbf{z}(k_j) - \boldsymbol{\mu}_j(k_j - 1))^T \end{aligned} \quad (6)$$

kjer je k_j trenutni indeks podatkov in $\mathbf{z}(k_j)$ je trenutni podatek j -tega roja.

3.1 Detekcija ravnih površin

Izhod algoritma 1 je urejena matrika rojev točk M , ki predstavljajo daljice. Urejenost matrike zopet izkoristimo z uporabo metode EPCC, tokrat v 3D-prostoru. V tem primeru poteka rojenje podatkov s prototipi ravnin (7), ki jih pridobimo direktno iz matrike $\boldsymbol{\Sigma}_j$ za primer 3D-podatkov

$$\mathbf{p}_j^T = \left[-\frac{\theta_1}{\sqrt{\theta_1^2 + \theta_2^2 + 1}} \quad -\frac{\theta_2}{\sqrt{\theta_1^2 + \theta_2^2 + 1}} \quad \frac{1}{\sqrt{\theta_1^2 + \theta_2^2 + 1}} \right] \quad (7)$$

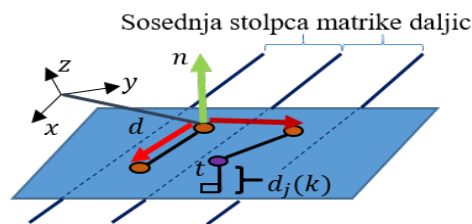
kjer sta

$$\theta_1 = \frac{\sigma_{13}\sigma_{22} - \sigma_{23}\sigma_{12}}{\sigma_{11}\sigma_{22} - \sigma_{12}^2} \quad \theta_2 = \frac{\sigma_{13}\sigma_{12} - \sigma_{23}\sigma_{11}}{\sigma_{12}^2 - \sigma_{11}\sigma_{22}} \quad (8)$$

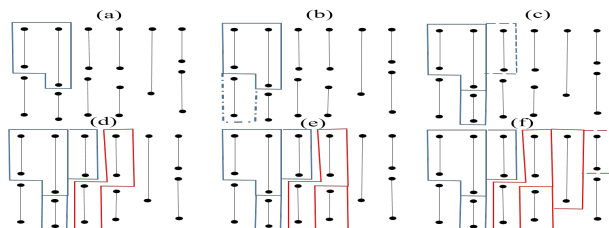
Algoritem za detekcijo ravnin roji podatke s primerjanjem daljic med sosednjimi stolpci matrike M . Postopek se začne z inicializacijo roja ravne površine s primerjanjem začetnih in končnih točk daljic med sosednjima stolpcema (slika 3). Pri vsaki inicializaciji pripišemo razvrščenim daljicam oznako pripadnosti j -temu roju (na sliki 4 so roji označeni z različno barvo). S pomočjo oznak lahko širimo in sledimo rojem skozi stolpce matrike M . Po vsaki inicializaciji sledi postopek širjenje roja, ki je pogojeno s kriterijem (5) oz. se preverja ustreznost ortogonalne razdalje začetne točke d_{j1} in končne točke d_{j2} opazovane daljice do j -tega prototipa ravnine. Da bi posamezno daljico razvrstili v pravi roj, je potrebno preveriti pripadnost le-te do vseh rojev. Daljice, ki jih ni možno razvrstiti, gredo v naslednjem koraku v postopek inicializacije novega roja.

Zaradi podvrženosti podatkov šumu, se napaka meritev prenese tudi na odkrite daljice. Ta problem lahko omilimo z modeliranjem šuma, ki so ga predstavili Sung in drugi v [25]. Tako v kriterij rojenja (5) vpeljemo model šuma

$$\sigma_z(z) = k_1 z^2 + k_2 z + k_3 \quad (9)$$



Slika 3: Inicializacija roja ravne površine, glede na ortogonalno razdaljo $d_j(k)$ do le-te.



Slika 4: Tvorimo roj ravne površine (a). Kandidat za širjenje (b), ki ne ustreza kriterijem razvrstitve. Črtkan kandidat, ki se preverja v naslednjem koraku (c). Postopek se ponovi za vsak novi roj.

kjer je $k_1 = 0.003949$, $k_2 = 0.0007278$ in $k_3 = 0.001063$ ter dobimo

$$d_j(k) < k_{max} \sqrt{\sigma_j} + \sigma_z \quad (10)$$

V oblaku točk je prisotnih veliko lukenj, ki lahko predstavljajo ločnice med površinami. Zveznost ravnih površin ohranimo z uvedbo kriterija (11), ki upošteva najmanjšo evklidsko razdaljo d_{evk} , začetne ali končne točke daljice, do opazovanega roja ravne površine,

$$d_{evk} \leq d_1 \quad (11)$$

kjer je d_1 konstanta in določa zgornjo mejo d_{evk} . V Algoritemu 2 je predstavljena psevdokoda detekcije ravnin.

4 Eksperimenti

Eksperimenti so izvedeni na podatkovni bazi, ki smo jo pridobili z globinskim senzorjem Intel RealSense D435i in predstavlja štiri notranje prizore (slika 5). Posamezni prizor vsebuje 129600 točk objektov z ravnimi površinami. Modificirano metodo EPCC smo primerjali z metodo rastočega območja in RANSAC, ki sta del knjižnice za obdelavo oblakov točk Point Cloud Library [26]. V tabeli 1 so opisani nastavljeni parametri algoritmov, ki so določeni s preizkušanjem za posamezen notranji prizor.

5 Rezultati

Na sliki 6 so prikazani rezultati planarne segmentacije po posameznih prizorih podatkovne baze. V primeru tretjega in četrtega prizora smo za vse metode upoštevali uspešno detekcijo tistih ravnih površin, ki so po zajemu z globinske kamere ostale dovolj nepopačene. Površini, ki sta pri tem izvzeti sta označeni na sliki 5. Omenimo naj,

Tabela 1: Opis vhodnih parametrov

Param.	Vrednost				Opis	Metoda
c_{th}	1.0				Prag ukrivljenosti	Rastoče območje
Θ_{th}	2°	3°	2°	2°	Prag kota med dvema normalama	
num_k	30				Število upoštevanih sosednjih točk pri izračunu normale izbrane točke	
min_c	50				Minimalna velikost roja	
max_c	10 ⁶				Maksimalna velikost roja	
k_n	50				Upoštevanje k -najbližjih sosedov v postopku segmentacije	
d_{th}	0.5 cm	1 cm	1.5 cm	1.5 cm	Prag oddaljenosti do modela ravnine	RANSAC
it_{max}	100				Največje število iteracij	
σ_{initP}	10 ⁻⁶ m ²	10 ⁻⁶ m ²	5 · 10 ⁻⁶ m ²	5 · 10 ⁻⁶ m ²	Začetna varianca razdalj pri detekciji ravnin	EPCC
σ_{init}	2.5 · 10 ⁻⁷ m ²					
k_{min}	9					
k_{max}	3					
d_1	2 cm	1 cm	3 cm	4 cm	Glej poglavje 3	
Prizor	(a)	(b)	(c)	(d)		

Algoritem 2 Odkrivanje ravnin z metodo EPCC

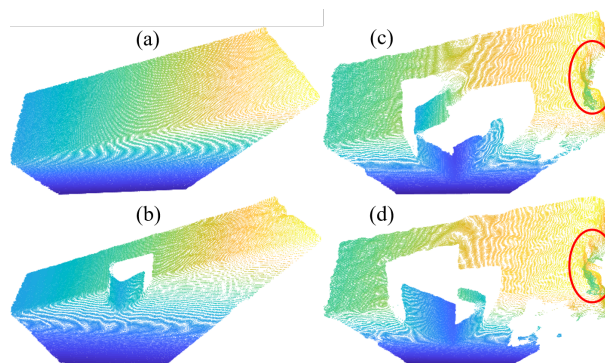
```

1: Definicija parametrov  $\sigma_{initP}$ ,  $d_1$ ,  $k_{max}$ ,
    $c$  (št. stolpcev matrike  $M$ ) in matrika  $M$ .
2: for  $k = 1 : c$ 
3:    $k$ -ti stolpec matrike  $M$  shranimo v  $st_k$ ,
   ( $k - 1$ )-ti pa v  $st_{k-1}$ .
4:   for  $i = 1 : length(st_k)$ 
5:      $i$ -to daljico shranimo v  $seg_i$ 
6:     for  $h = 1 : length(st_{k-1})$ 
7:        $h$ -to daljico shranimo v  $seg_h$ 
8:       if ( $seg_h$  pripada  $j$ -temu roju)
9:         if ( $(d_{j1}, d_{j2}) \leq k_{max} \sqrt{\sigma_j} + \sigma_z$  &
            &  $d_{evk} \leq d_1$ )
10:           $seg_i$  dodamo v  $j$ -ti roj in posodobimo
             $\mu_j$ ,  $\Sigma_j$ ,  $p_j$ ,  $\sigma_j$  ter prekinemo
            izvajanje trenutne zanke.
11:        end
12:      end
13:    if (Klasifikacija neuspešna)
14:      if ( $d_j(k) \leq k_{max} \sqrt{\sigma_{initP}} + \sigma_z$  &
          &  $d_{evk} \leq d_1$ )
15:        Tvorimo nov prototip in ocenimo:
           $\mu_j$ ,  $\Sigma_j$ ,  $p_j$ ,  $\sigma_j$ .
16:      end
17:    end
18:  end
19: end
20: end

```

da rdeče točke, v oblaku točk, predstavljajo nerazvrščene točke v primeru segmentacije z metodo rastočega območja.

V tabeli 2 so prikazani rezultati segmentacij v smislu deleža pravilno detektiranih ravnih površin $\frac{D}{N}$, kjer je D število pravilno detektiranih površin in N število vseh ravnih površin v določenem prizoru. Ocenjujemo tudi delež nadsegmentiranih $\frac{N_p}{N}$ in podsegmentiranih $\frac{N_p}{N}$



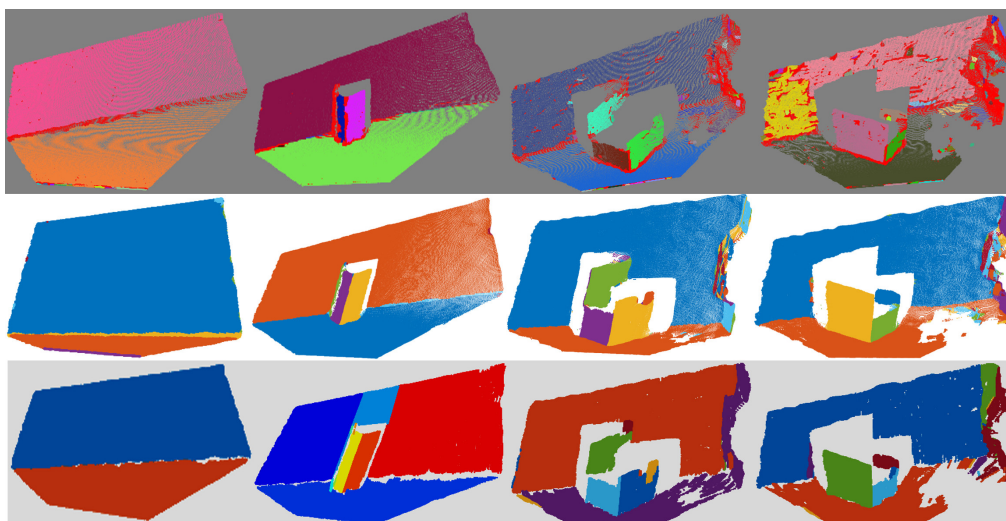
Slika 5: Podatkovna baza: prvi (a), drugi (b), tretji (c) in četrti prizor (d)

površin, kjer je N_n število nadsegmentiranih in N_p število podsegmentiranih površin. Glede na rezultate vidimo, da lahko metodo EPCC primerjamo z ustaljenimi metodami za planarno segmentacijo oblakov točk na podani podatkovni bazi. Za razliko od metode RANSAC, pa metoda EPCC, poleg rastočega območja, izloči neplanarne točke v postopku ekstrakcije ravnih površin. Slednje, poleg izbire vhodnih parametrov, pripomore tudi k manjši nadsegmentaciji površin, kot je razvidno iz tabele 2.

Pomankljivost uporabljene metode RANSAC je, da ne ohranja zveznosti površin, kar lahko vodi do prekrivanja določenih območij, v kolikor niso bila v prejšnjih korakih detektirana kompletna območja. Prednost osnovne metode RANSAC pa je, da ima manjše število vhodnih parametrov, ki jih moramo empirično določiti.

6 Zaključek

Predstavili smo rekurziven pristop k segmentaciji oblaka točk, ki temelji na metodi EPCC. Razvit algoritem lahko predstavlja sestavni del avtonomnega mobilnega sistema



Slika 6: Rezultati planarne segmentacije prizorov z metodo rastočega območja (zgoraj), RANSAC (sredina) in EPCC (spodaj).

Tabela 2: Rezultati segmentacije notranjih prizorov

Metoda	$D \left(\frac{D}{N} \right)$	$\frac{N_R}{N}$	$\frac{N_P}{N}$	N
Prvi Prizor				
Rastoče območje	2 (100%)	0.5	0.0	2
RANSAC	2 (100%)	1	0.0	
EPCC	2 (100%)	0.0	0.0	
Drugi Prizor				
Rastoče območje	4 (100%)	0.0	0.5	4
RANSAC	4 (100%)	0.5	0.0	
EPCC	4 (100%)	0.25	0.0	
Tretji Prizor				
Rastoče območje	5 (50%)	0.2	0.3	10
RANSAC	8 (80%)	0.5	0.0	
EPCC	8 (80%)	0.2	0.0	
Četrti Prizor				
Rastoče območje	5 (62.5%)	0.38	0.5	8
RANSAC	8 (100%)	0.38	0.0	
EPCC	6 (75%)	0.25	0.13	

za potrebe vizualne navigacije ali gradnje zemljevida. Predlagana metoda izkorišča urejenost podatkov, kar omogoča hitrejšo in lažjo implementacijo ter sprotno obdelavo podatkov. Metoda EPCC je torej dobra izbira za odkrivanje linearnih prototipov v urejenih podatkih, saj daje dobre rezultate, kljub navzočnosti šuma. Algoritem tudi omogoča delovanje z drugimi senzorskimi sistemi, ki imajo podoben princip zajemanja podatkov (npr. laserski merilniki razdalje). Rezultati uporabljenih metod so pokazali, da EPCC lahko primerjamo z že ustaljenimi pristopi za obdelavo oblakov točk v smislu natančnosti. V prihodnosti želimo algoritem uporabiti za avtomatsko gradnjo zemljevida, ki bo podlaga za lokalizacijo in navigacijo avtonomnega mobilnega sistema v notranjih prostorih.

Zahvala

Raziskovalni program št. P2-0219 je sofinancirala Javna agencija za raziskovalno dejavnost Republike Slovenije iz državnega proračuna.

Literatura

- [1] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane slam using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, p. 3795, 2019.
- [2] G. Klančar, L. Teslić, and I. Škrjanc, "Mobile-robot pose estimation and environment mapping using an extended kalman filter," *International Journal of Systems Science*, vol. 45, no. 12, pp. 2603–2618, 2014.
- [3] A. Zdešar, I. Škrjanc, and G. Klančar, "Visual trajectory-tracking model-based control for mobile robots," *International journal of advanced robotic systems*, vol. 10, no. 323, pp. 1–12, 2013.
- [4] A. Zdešar, I. Škrjanc, and G. Klančar, "Homography estimation from circular motion for use in visual control," *Robotics and autonomous systems*, vol. 62, pp. 1486–1496, Oct. 2014.
- [5] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," *The IEEE International Conference on Computer Vision (ICCV)*, pp. 2901–2910, 2019.
- [6] S. Woo, J. Shin, Y.-H. Lee, Y.-H. Lee, H. Kang, H. Choi, and H. Moon, "Stair-mapping with point-cloud data and stair-modeling for quadruped robot," *2019 16th International Conference on Ubiquitous Robots (UR)*, pp. 81–86, 2019.
- [7] P. Tian, X. Hua, K. Yu, and W. Tao, "Robust segmentation of building planar features from unorganized point cloud," *IEEE Access*, vol. 8, pp. 30873–30884, 2020.
- [8] D. Xu, F. Li, and H. Wei, "3d point cloud plane segmentation method based on ransac and support vector machine," *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 943–948, 2019.
- [9] E. Grilli, F. Menna, and F. Remondino, "A review of point clouds segmentation and classification algorithms," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 339–344, 2017.
- [10] I. Škrjanc, J. A. I. Martinez, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification : a survey," *Information sciences*, vol. 490, pp. 344–368, July 2019.

- [11] I. Škrjanc, “Cluster-volume-based merging approach for incrementally evolving fuzzy gaussian clustering - egauss+,” *IEEE transactions on fuzzy systems*, pp. 1–11, Apr. 2019.
- [12] I. Škrjanc, B. Sašo, E. Lughofer, and D. Dovžan, “Inner matrix norms in evolving cauchy possibilistic clustering for classification and regression from data streams,” *Information sciences*, vol. 478, pp. 540–563, Apr. 2019.
- [13] I. Škrjanc and B. Sašo, “Incremental fuzzy c-regression clustering from streaming data for local-model-network identification,” *IEEE transactions on fuzzy systems*, vol. 28, pp. 758–767, Apr. 2020.
- [14] D. Dovžan and I. Škrjanc, “Fuzzy space partitioning based on hyperplanes defined by eigenvectors for takagi-sugeno fuzzy model identification,” *IEEE transactions on industrial electronics*, vol. 67, pp. 5144–5153, June 2020.
- [15] G. Klančar and I. Škrjanc, “Evolving principal component clustering with a low run-time complexity for lrf data mapping,” *Applied Soft Computing*, vol. 35, pp. 349–358, 2015.
- [16] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Readings in Computer Vision*, pp. 726–740, 1987.
- [17] H. P. D. Poz and M. S. Y. Ywata, “Adaptive random sample consensus approach for segmentation of building roof in airborne laser scanning point cloud,” *International Journal of Remote Sensing*, vol. 41, no. 6, pp. 2047–2061, 2020.
- [18] H. L. Nguyen, D. Belton, and P. Helmholz, “Planar surface detection for sparse and heterogeneous mobile laser scanning point clouds,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 151, pp. 141–161, 2019.
- [19] P. Hough, “Method and means for recognizing complex patterns,” in: *US Patent*, 1962.
- [20] E. Vera, D. Lucio, L. A. Fernandes, and L. Velho, “Hough transform for real-time plane detection in depth images,” *Pattern Recognition Letters*, vol. 103, pp. 8–15, 2018.
- [21] H. D. Khanh, V. S. Nguyen, A. Do, and N. T. Dzung, “An effective randomized hough transform method to extract ground plane from kinect point cloud,” *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1053–1058, 2019.
- [22] Z. Dong, B. Yang, P. Hu, and S. Scherer, “An efficient global energy optimization approach for robust 3d plane segmentation of point clouds,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 137, pp. 112–133, 2018.
- [23] H. Wu, X. Zhang, W. Shi, S. Song, A. Cardenas-Tristan, and K. Li, “An accurate and robust region-growing algorithm for plane segmentation of tfs point clouds using a multiscale tensor voting method,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 10, pp. 4160–4168, 2019.
- [24] M. Huang, P. Wei, , and X. Liu, “An efficient encoding voxel-based segmentation (evbs) algorithm based on fast adjacent voxel search for point cloud plane segmentation,” *Remote Sensing*, vol. 11, no. 23, 2019.
- [25] M. Sung, H. Chae, D. Noh, N. H, and D. Hong, “Analysis and noise modeling of the intel realsense d435 for mobile robots,” *International Conference on Ubiquitous Robots (UR)*, pp. 707–711, 2019.
- [26] “Point cloud library.” <http://www.pointclouds.org/>. Accessed: 2020-2-12.