

**SLOVENSKÁ TECHNICKÁ UNIVERZITA  
V BRATISLAVE  
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**MRAČNO BODOV (POINT CLOUD)**

**ZADANIE Č.4 Z PREDMETU POČÍTAČOVÉ VIDENIE A SPRACOVANIE OBRAZU**

Študijný program:	Robotika a kybernetika
Študijný odbor:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vypracoval:	Damián Bombara, Tomáš Stupeň

# 1 Zadanie

## Zadanie 4. Mračno bodov (Point cloud)

### Znenie zadania

Cieľom zadania, je oboznámiť sa s pracou s mračnom bodov (point cloud) a segmentácia objektov v priestore. Študent si vyskúša vytvorenie vlastného mračna bodov a aplikáciu metód na získanie segmentovaného priestoru. Použitie externých knižníc ako open3d, sklearn, opencv, a iných je dovolené a odporúčané.

Zadanie je dokopy za **25 b.** a pozostáva z viacerých úloh:

1. Vytvorenie mračna bodov pomocou Kinect v2 pre testovanie. Nájdite online na webe mračno bodov popisujúce väčší priestor (väčší objem dát aspoň 4 x 4 metre) pre testovanie algoritmov a načítajte mračno dostupného datasetu [2 b.].
2. Pomocou knižnice (open3d - python) načítate vytvorené mračno bodov a zobrazíte. [2 b.].
3. Mračná bodov očistite od okrajových bodov. Pre tuto úlohu je vhodné použiť algoritmus RANSAC [5 b.] .
4. Segmentujete priestor do klastrov pomocou vhodne zvolených algoritmov (K-means, DBSCAN, BIRCH, Gaussian mixture, mean shift ...). Treba si zvoliť aspoň 2 algoritmy a porovnať ich výsledky [5+5 b.].
5. Detailne vysvetlite fungovanie zvolených algoritmov. [4 b.] (Keďže neimplementujete konkrétny algoritmus ale používate funkcie tretích strán je potrebné rozumieť aj ako sú funkcie implementované)
6. Vytvorte dokumentáciu zadania:
  - popis implementovaných algoritmov,
  - grafické porovnanie výstupov,
  - vysvetlite rozdiel v kvalite výstupov pre rozdielne typy algortimov.[2 b.].

**Termin odovzdania:** 12. cvičenie semestra

## 2 Vypracovanie úloh

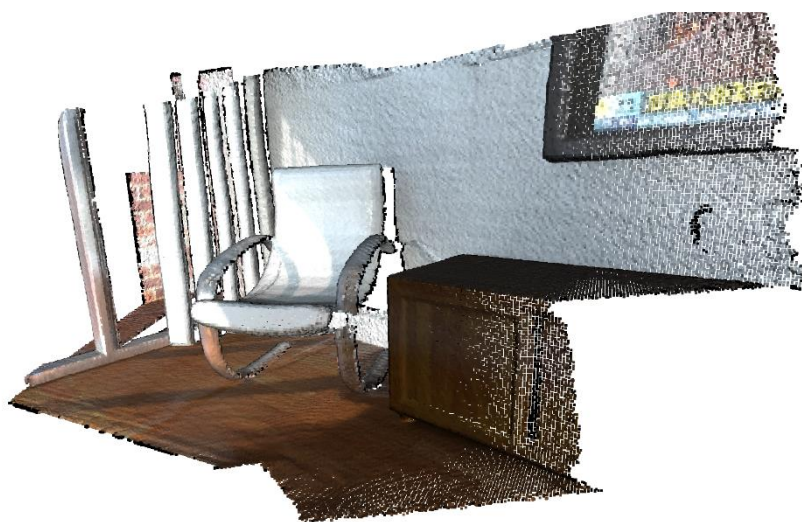
V tejto kapitole uvidíme postup pri vypracovávaní jednotlivých úloh. Keďže išlo o využitie už implementovaných algoritmov, dôležité bolo najmä porozumieť tomu, čo a prečo využívame.

### 2.1 Point cloud z tretej strany a náš vlastný point cloud

Prvou úlohou bolo vytvoriť mračno bodov pomocou Kinect v2 pre testovanie. Na to sme využili dopredu implementovaný kód, ktorý sa nachádzal na laboratórnom počítači.

Následne sme hľadali nejaký point cloud na internete, ktorý by nám vyhovoval, čiže taký aby sa na ňom nachádzalo viacero objektov.

Keďže sme zabudli vytvoriť snímku nášho vlastného mračna bodov ešte predtým, než sme naň aplikovali rôzne algoritmy, uvidíme len snímku point cloudu z internetu.



Obr. 1. Point cloud z internetu - originál

Naše mračno bodov si ukážeme už pri konkrétnych algoritmoch, kde bude pekne vidieť, čo obsahuje aj keď už bude pozmenený.

## 2.2 Načítanie mračien a zobrazenie

Druhou úlohou bolo pomocou knižnice *open3d* načítať mračná bodov a zobrazit' ich. To sa nám podarilo, keďže bez tohto kroku by sme nemohli aplikovať algoritmy, ani si pozrieť aký následok algoritmy na point cloudy mali.

## 2.3 Algoritmus RANSAC

RANSAC (RANdom SAmple Consensus) je algoritmus, pomocou ktorého vieme mračno bodov očistiť od okrajových bodov (outliers). Teda vieme point cloud rozdeliť na „inlier set“ a „outlier set“.

Fungovanie tohto algoritmu vieme rozdeliť do štyroch krokov:

1. Výber bodov, ktoré považujeme za inlier body.
2. S týmito vybranými bodmi vypočítame parametre modelu.
3. Vyhodnotenie modelu (koľko zvyšných bodov „podporuje“ vypočítaný model).
4. Opakujeme až kým nenájdeme najlepší model

```
pcd.segment_plane(distance_threshold=0.10, ransac_n=3,  
num_iterations=1000)
```

V tomto kuse kódu máme tri parametre, ktoré ovplyvňujú výsledok:

- *distance\_threshold* – určuje maximálnu vzdialenosť bodu od modelu roviny, aby bol považovaný za súčasť roviny.
- *ransac\_n* – počet bodov, ktoré sú náhodne vybrané pre každú iteráciu na vytvorenie hypotetického modelu (roviny). Zvolili sme hodnotu 3, čo je minimálny počet bodov potrebných na definíciu roviny v 3D priestore.
- *num\_iterations* – určuje maximálny počet iterácií algoritmu, každá iterácia zahŕňa náhodný výber bodov a vytvorenie hypotetického modelu.

Takto sa zmenili naše mračná bodov po aplikovaní tohto algoritmu:



Obr. 2. Point cloud z internetu - RANSAC



Obr. 3. Náš point cloud - RANSAC

Na obrázkoch je zadná stena vymaľovaná červenou farbou, toto značí spomínaný „outlier set“, čiže okrajové body. Všetko ostatné sú body, ktoré sme sa rozhodli ponechať.

Mimochodom obrázok č.3 je náš vytvorený point cloud, teda ak by stena nebola zamaľovaná červenou farbou bolo by to originálne, nepozmenené mračno bodov.

## 2.4 Segmentácia priestoru do klastrov

Segmentovať priestor do klastrov sme mohli viacerými algoritmami, my sme sa rozhodli vybrať si algoritmy DBSCAN a BIRCH.

Teraz si ukážeme, čo jednotlivé algoritmy s našimi mračnami bodov urobili a porovnáme ich výsledky.

### 2.4.1 Segmentácia pomocou DBSCAN algoritmu

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) je algoritmus, ktorý dokáže dať do klastra hocaký tvar. Potrebuje na to len dva parametre.

```
pcd.cluster_dbscan(eps=0.034, min_points=60)
```

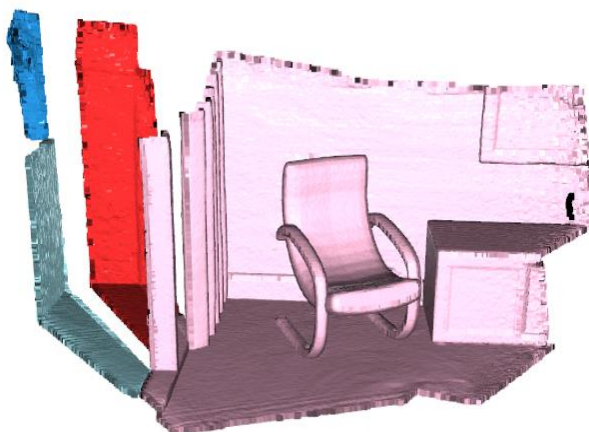
V tomto kuse kódu, sa nachádzajú vyššie spomínané parametre:

- *epsilon* – je to v podstate polomer, z ktorého náhodne vybraný bod „vysiela“ dĺžku polomeru do všetkých strán a tvorí okruh okolo toho bodu.
- *minimum\_points* – minimálny počet bodov (vrátane toho náhodne vybraného), ktoré musia existovať v okruhu vybraného bodu, aby sa mohol vytvoriť klaster.

Akonáhle sa klaster sformuje môže rásť ďalej, a to tak, že všetky body, ktoré už sú v klasteri takisto budú vytvárať okruh. Ak sa v tom okruhu budú nachádzať ďalšie body, stanú sa súčasťou klastru, aj keď nie sú v dosahu *epsilonu* štartovacieho bodu (náhodne vybraného).

Tento proces sa opakuje až kým body, nenájdu v svojom okruhu žiadne iné, vtedy je rast klastra ukončený.

Takto sa zmenili naše mračná bodov po aplikovaní algoritmu DBSCAN:



Obr. 4. Point cloud z internetu - DBSCAN



Obr. 5. Náš point cloud - DBSCAN

Tu vidíme ako algoritmus zmenil point cloudy. Mračno bodov z internetu v podstate rozdelil na štyri klastre, dalo by sa povedať na „časti miestnosti“.

Naše mračno bodov zase na viacero klastrov – modrou farbou sú monitory, zelenou stôl, fialovou myš, čiernou počítač...

Touto metódou sa nám lepšie podarilo segmentovať point cloud z internetu.

### 2.4.2 Segmentácia pomocou BIRCH algoritmu

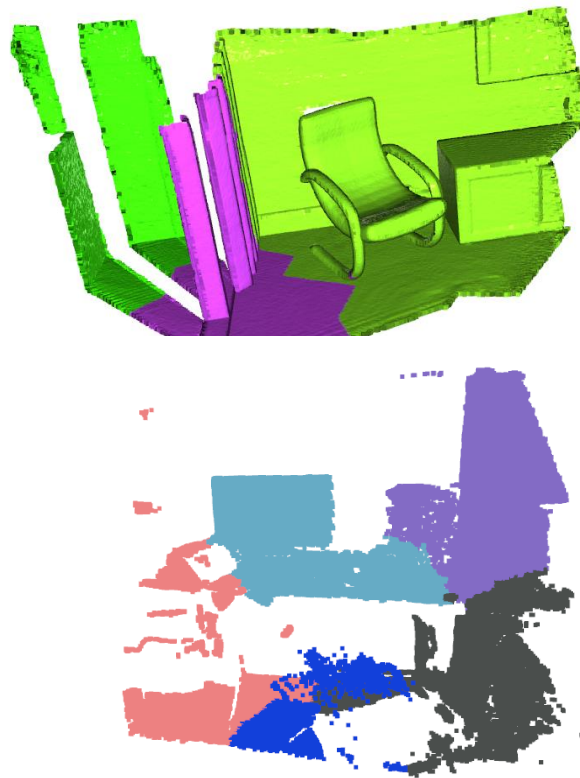
BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) je hierarchický klasterovací algoritmus.

```
Birch(threshold=0.10, n_clusters=5)
```

V tomto kuse kódu sa nachádzajú dva parametre, ktoré ovplyvňujú výsledky:

- *threshold* – určuje maximálnu veľkosť podstromu podľa počtu vzoriek pred rozdeľovaním na klastre. V podstate určuje, ako sú vzdialené klastre, aby sa spojili do jedného väčšieho klastra
- *n\_clusters* – počet klastrov, ktorý dostaneme po dokončení algoritmu.

Takto sa zmenili naše mračná bodov po aplikovaní algoritmu BIRCH:



Obr. 7. Náš point cloud - BIRCH



Tu vidíme ako algoritmus zmenil point cloudy. Mračno bodov z internetu rozdelil na tri klastre, ale horšie ako algoritmus DBSCAN. Opäť rozdelil miestnosť na časti, no fialovou farbou zasahuje do dvoch častí naraz.

Naše mračno bodov algoritmus BIRCH rozdelil lepšie ako pri predchádzajúcom prípade, rozdelil point cloud na niekoľko klastrov – monitor so stolom modrou, monitor na kraji obrazovky s počítačom fialovou, a spodok stola pod krajným monitorom čiernou...

Touto metódou sa nám lepšie podarilo segmentovať náš vytvorený pointcloud.

### 3 Záver

Cieľom zadania bolo oboznámiť sa s prácou s mračnom bodov (point cloud) a segmentáciou objektov v priestore. Mali sme vyskúšať vytvoriť vlastné mračno bodov a aplikovať metódy na získanie segmentovaného priestoru. Pri vypracovaní sme využili knižnicu open3d.

Vytvorili sme mračno bodov pomocou Kinect v2 pre testovanie a takisto sme našli externý point cloud z internetu.

Pomocou knižnice open3d sme vytvorené mračno bodov aj to z internetu zobrazili.

Obe mračná bodov sme očistili od okrajových bodov pomocou RANSAC algoritmu oba point cloudy sme rozdelili na „inliers“ a „outliers“ body. V našom prípade okrajové body predstavovali zadné steny.

Následne sme segmentovali priestor do klastrov pomocou algoritmov DBSCAN a BIRCH. Keď sme porovnávali ich výsledky zistili sme, že DBSCAN algoritmus lepšie segmentoval point cloud nájdený na internete a BIRCH bol úspešnejší pri našom mračne bodov.

Vytvorili sme dokumentáciu a detailne vysvetlili fungovanie zvolených algoritmov, keďže sme splnili všetky body zadania môžeme povedať, že sa nám zadanie podarilo úspešne vypracovať.