



**Slovenská technická univerzita v Bratislave**

Fakulta elektrotechniky a informatiky

---

Počítačové videnie a spracovanie obrazu

**Mračno bodov**

Zadanie 4

**Martin Berki, Viktoriia Fedyniak**

ZS 2022/23

## Zadanie 4 – Mračno bodov

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 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 4x4 metre) pre testovanie algoritmov a načítajte mračno dostupného datasetu (2B)
2. Pomocou knižnice (open3d - python) načítate vytvorené mračno bodov a zobrazíte. (2B)
3. Mračná bodov očistite od okrajových bodov. Pre tuto úlohu je vhodné použiť algoritmus RANSAC. (5B)
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+5B)
5. Detailne vysvetlite fungovanie zvolených algoritmov. (4B) (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 algoritmov) (2B).

Vysvetlenie pre RANSAC a segmentáciu obrazu

<https://towardsdatascience.com/how-to-automate-3d-point-cloud-segmentation-and-clustering-with-python-343c9039e4f5>

Pri riešení zadania je možné použiť aj iné knižnice, ktoré nie sú spomenuté v zadaní samotnom. Študent musí však pri odovzdaní vedieť zodpovedať ako dané knižnice fungujú, čo robia použité funkcie atď.

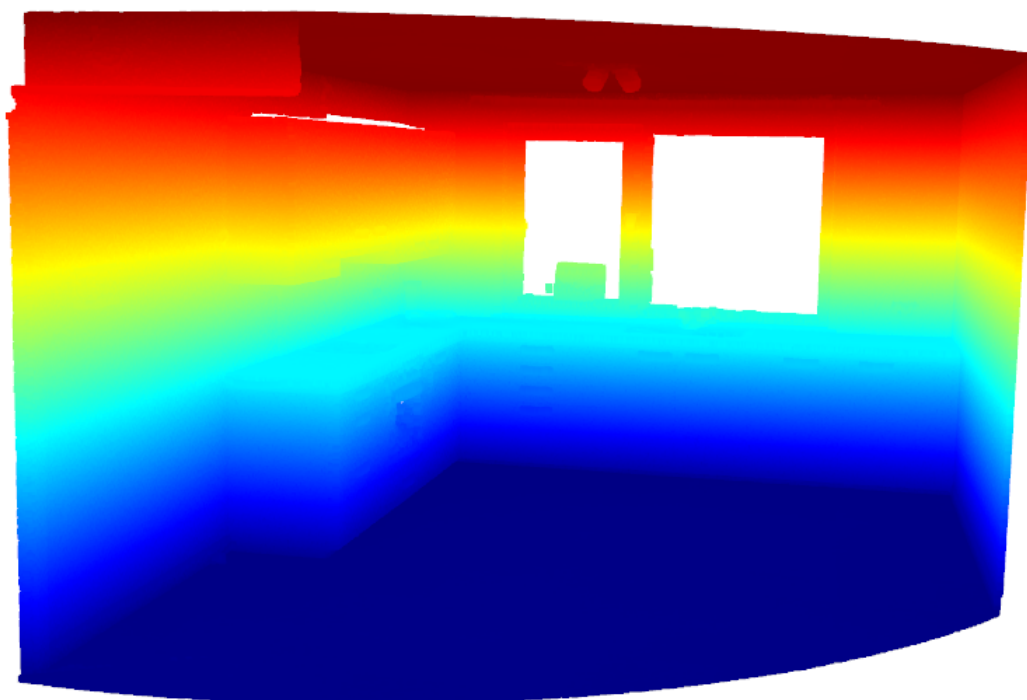
Zadanie sa odovzdáva na 12 cvičení.

# 1 Dáta

## 1.1 Point cloud získaný z kinectu



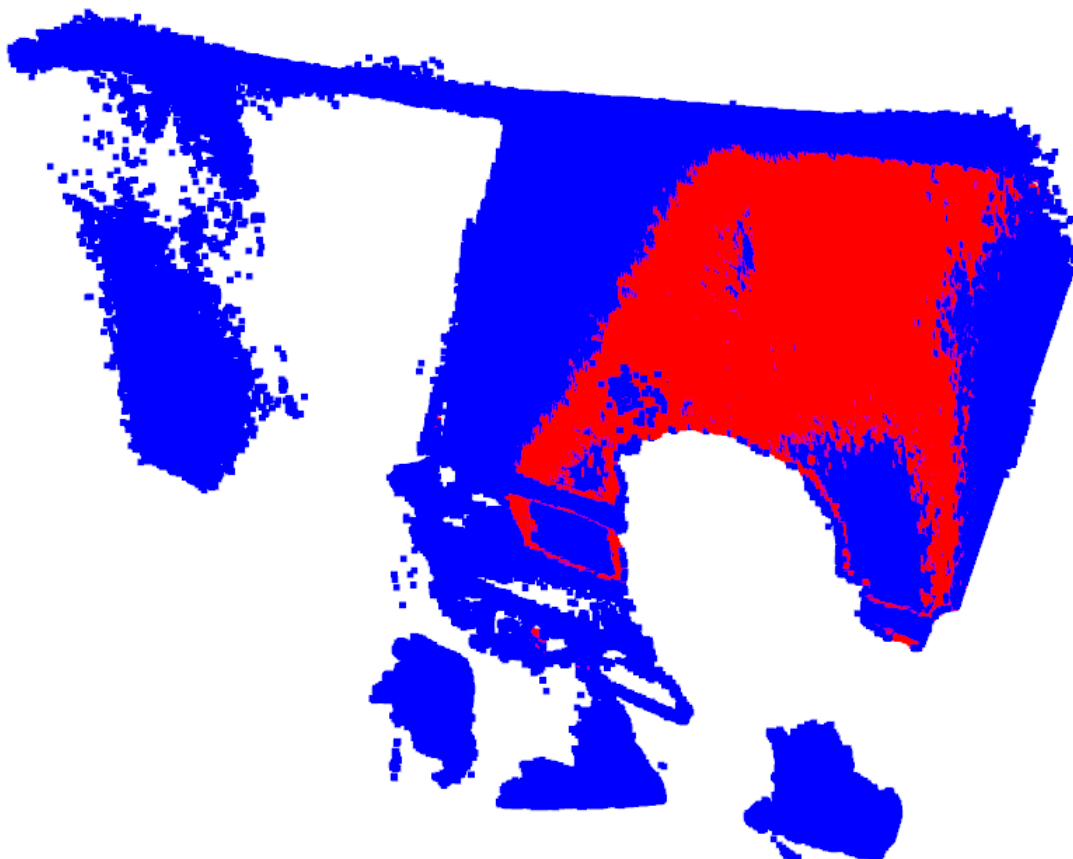
## 1.2 Point cloud získaný z internetu



## 2 Ransac

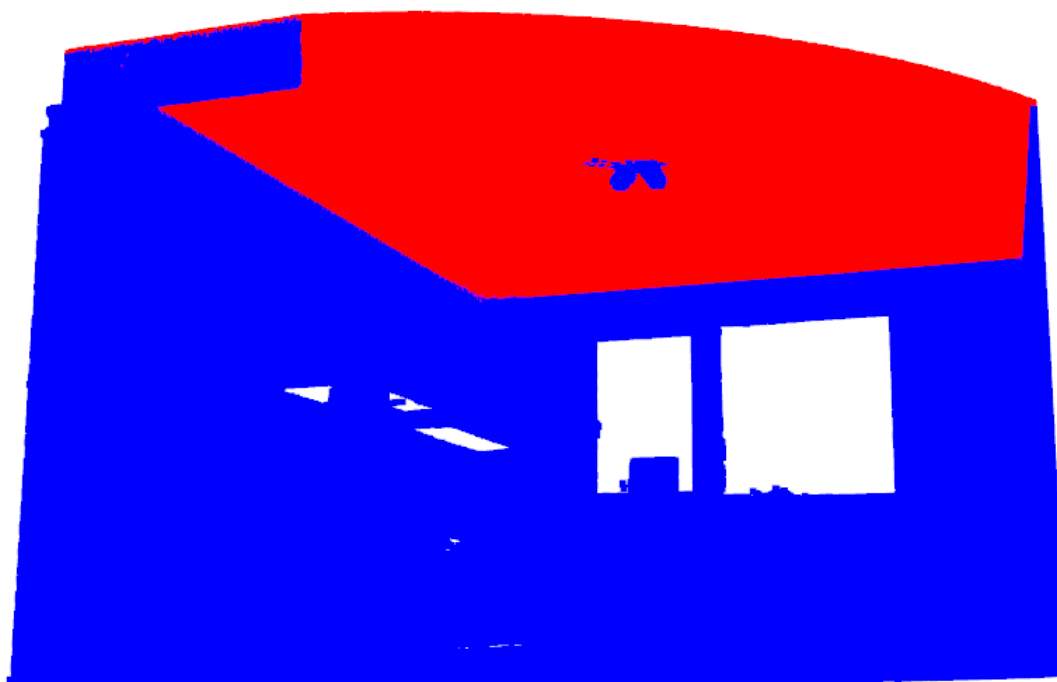
RANSAC (Random Sample Consensus) je iteratívny algoritmus ktorý sa používa na odhad parametrov modelu z dát, ktoré obsahujú odľahlé hodnoty. Cieľom tohto algoritmu je identifikovať príslušné hodnoty (dátá ktoré vyhovujú modelu) a odmietnuť hodnoty ktoré tomuto modelu nevyhovujú. Algoritmus náhodne vyberie podmnožinu z dát a snaží sa nafitovať model na tieto dáta. Ostatné dáta sú validované proti tomuto modelu a podľa nejakého threshold parametra sa určí, či tieto dáta vyhovujú danému modelu. Tento proces sa opakuje niekoľkokrát a model s najväčšou množinou vyhovujúcich hodnôt je považovaný za najlepší odhad skutočného modelu.

## 2.1 RANSAC aplikovaný na dáta z Kinectu



RANSAC aplikovaný na dáta z Kinectu vyhodnotil ako najlepšie vyhovujúcu množinu stenu v učebni D618. Parameter algoritmu „n\_ransac“ bol nastavený na 3 a počet iterácií bol nastavený na 1000.

## 2.2 RANSAC aplikovaný na dáta z internetu



V tomto prípade RANSAC taktiež ako naviac vyhovujúcu množinu vyhodnotil stenu, konkrétne strop kuchyne. Lepšie výsledky sa nepodarilo dosiahnuť ani rozličným modifikovaním parametrov algoritmu.



### 3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) je zhlukovací algoritmus, ktorý sa používa na zoskupovanie dátových bodov, ktoré sú tesne pri sebe, a zároveň na identifikáciu bodov, ktoré nie sú súčasťou žiadneho zhľuku, t.j. bodov so šumom.

Algoritmus funguje tak, že sa definuje okolie okolo každého dátového bodu na základe zadaného polomeru a potom sa spočíta počet dátových bodov v tomto okolí. Body s minimálnym počtom susedov (špecifikované ako parameter „min\_samples“) sa považujú za základné body a používajú sa na vytvorenie zhľukov. Body, ktoré nemajú dostatočný počet susedov na to, aby sa považovali za jadrové body, ale nachádzajú sa v susedstve jadrového bodu, sa považujú za hraničné body a sú tiež súčasťou zhľuku. Body, ktoré nemajú dostatok susedov na to, aby sa považovali za jadrové alebo hraničné body, sa považujú za šumové body.

Algoritmus DBSCAN má dva kľúčové parametre: „eps“, ktorý určuje polomer okolia každého bodu, a „min\_samples“, ktorý určuje minimálny počet susedov potrebných na to, aby sa bod považoval za jadrový bod. Tieto parametre možno zvoliť na základe hustoty údajov a ich vyladenie môže pomôcť zlepšiť výsledky zhľukovania.

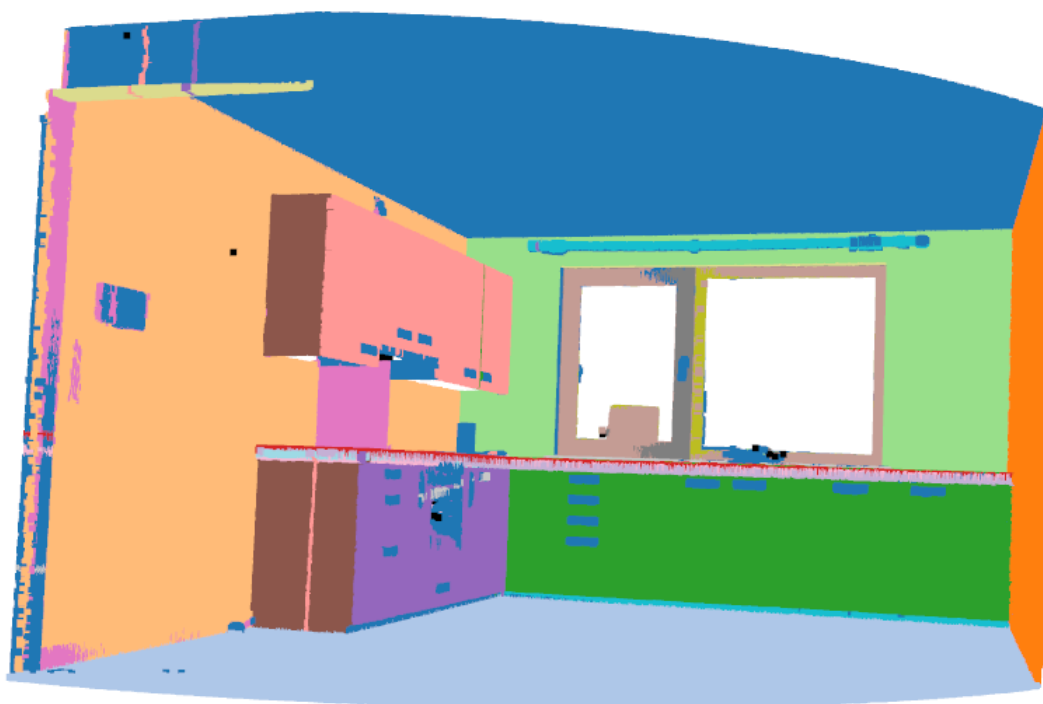
Jednou z výhod metódy DBSCAN je, že dokáže identifikovať zhľuky ľubovoľných tvarov a veľkostí a dokáže spracovať súbory údajov s rôznou hustotou. Je tiež relatívne efektívny, s časovou zložitou  $O(n \log n)$ . DBSCAN má uplatnenie v mnohých oblastiach vrátane spracovania obrazu, počítačového videnia a rozpoznávania vzorov.

### 3.1 DBSCAN aplikovaný na dáta z Kinectu



DBSCAN aplikovaný na naše dáta pomerne spoľahlivo zhlukuje mračno bodov. V tomto konkrétnom príklade algoritmus našiel 74 zhlukov.

### 3.2 DBSCAN aplikovaný na dáta z internetu



DBSCAN aplikovaný na mračno bodov stiahnuté z internetu zhlukuje objekty s vysokou presnosťou, kde v tomto prípade algoritmus našiel 97 zhlukov.

## 4 K-MEANS

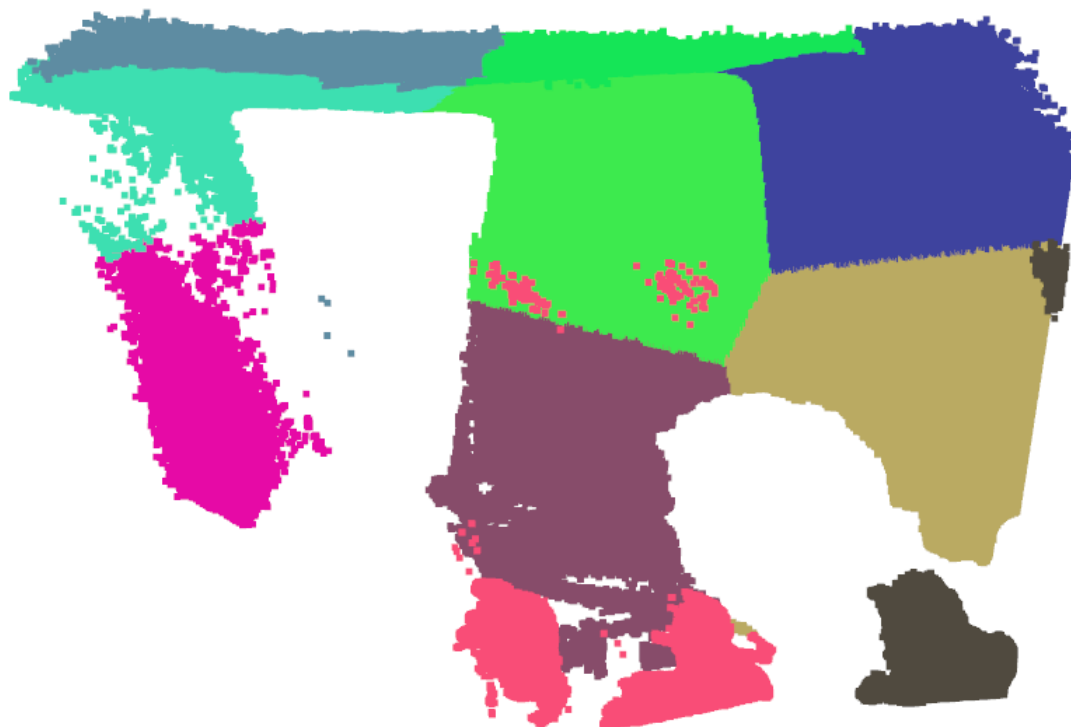
K-means je zhukovací algoritmus, ktorý sa používa na rozdelenie daného súboru údajov do pevného počtu zhukov ( $K$ ). Tento algoritmus sa radí do kategórie unsupervised learning, čo znamená, že na tréovanie nevyžaduje označené údaje.

Algoritmus funguje tak, že najprv náhodne vyberie  $K$  bodov zo súboru údajov, ktoré slúžia ako počiatočné centrá zhukov. Potom sa každý dátový bod priradí k najbližšiemu centru zhuku na základe jeho vzdialenosti od tohto centra. Po priradení všetkých dátových bodov k zhuku sa centrá zhukov aktualizujú výpočtom priemeru všetkých bodov priradených k danému zhuku. Tento proces sa opakuje, kým sa priradenia zhukov a centrá zhukov nezbližia, t. j. nedôjde k ďalšej zmene priradení alebo centier zhukov.

K-means je iteračný algoritmus, ktorého cieľom je minimalizovať účelovú funkciu súčtu štvorcov v rámci zhuku (WCSS), ktorá meria vzdialenosť medzi každým dátovým bodom a jeho priradeným stredom zhuku. Algoritmus zaručene konverguje k lokálnemu optimu, ale nie nevyhnutne ku globálnemu optimu, čo znamená, že konečné zhukovanie môže závisieť od počiatočného náhodného výberu zhukových centier.

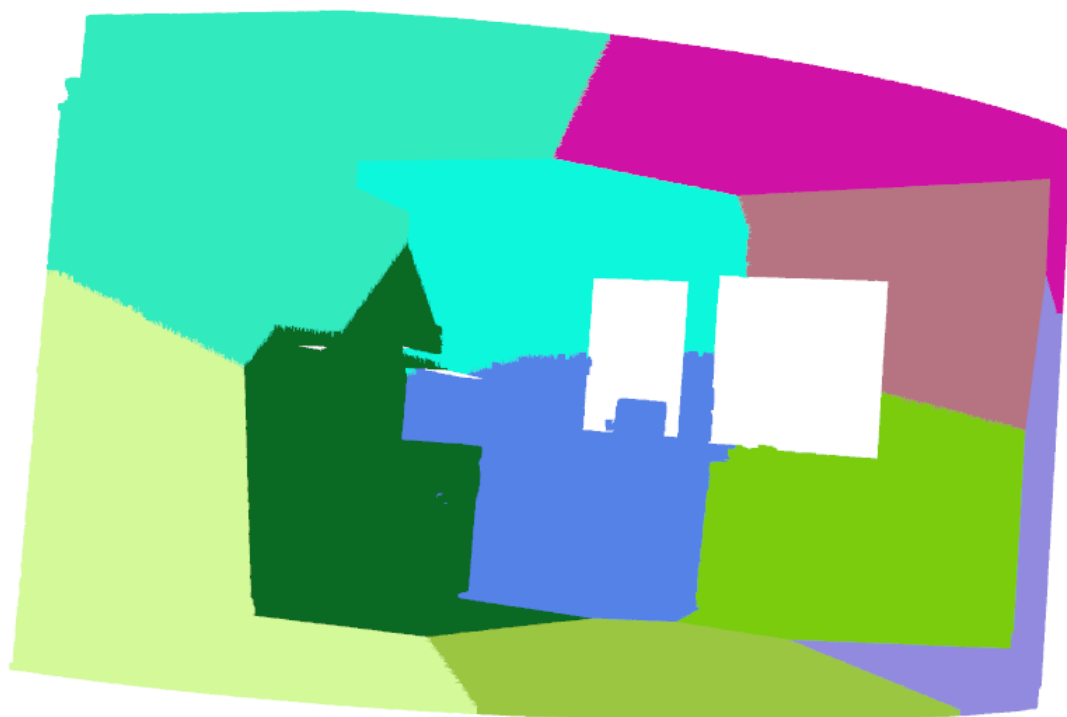
Jednou zo silných stránok K-means je jeho jednoduchosť a efektívnosť, s časovou zložitosťou  $O(n * K * I)$ , kde  $n$  je počet dátových bodov,  $K$  je počet zhukov a  $I$  je počet iterácií potrebných na konvergenciu. Tento algoritmus má vysoké uplatnenie v odvetví spracovania obrazu, avšak nemusí dobre fungovať, ak majú údaje zložitú štruktúru alebo ak počet zhukov nie je vopred známy.

## 4.1 K-MEANS aplikovaný na dáta z Kinectu



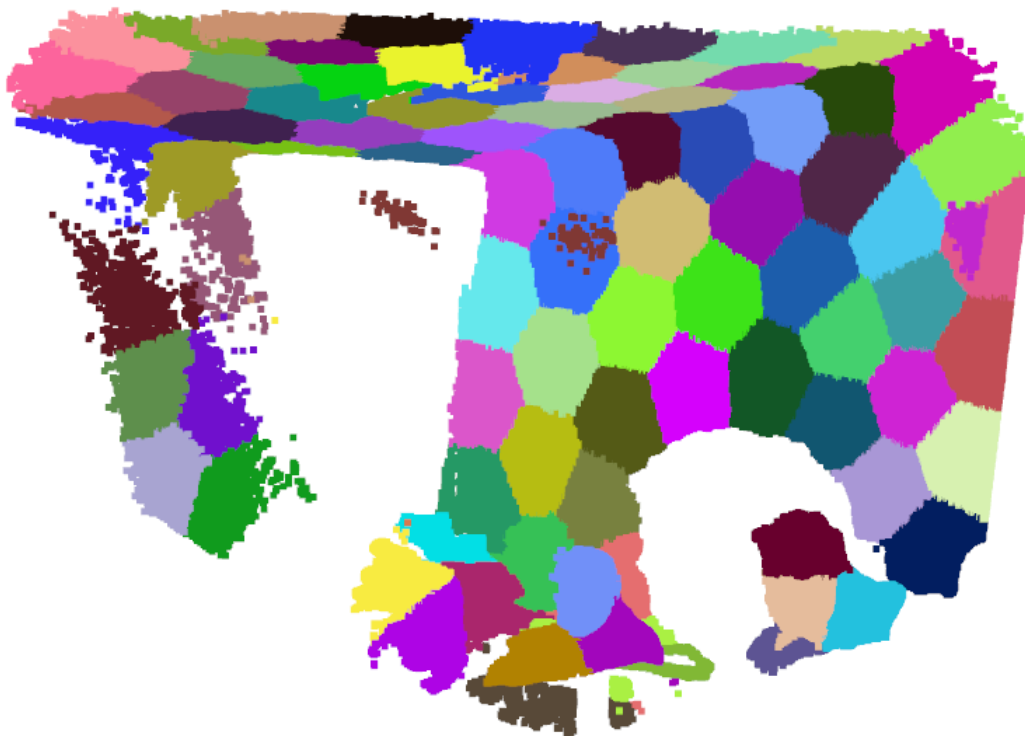
K-MEANS s preurčeným počtom 10 zhlukov síce do istej miery splnil účel, no algoritmus DBSCAN dosiahol neporovnateľne lepšie výsledky.

## 4.2 K-MEANS aplikovaný na dáta z internetu



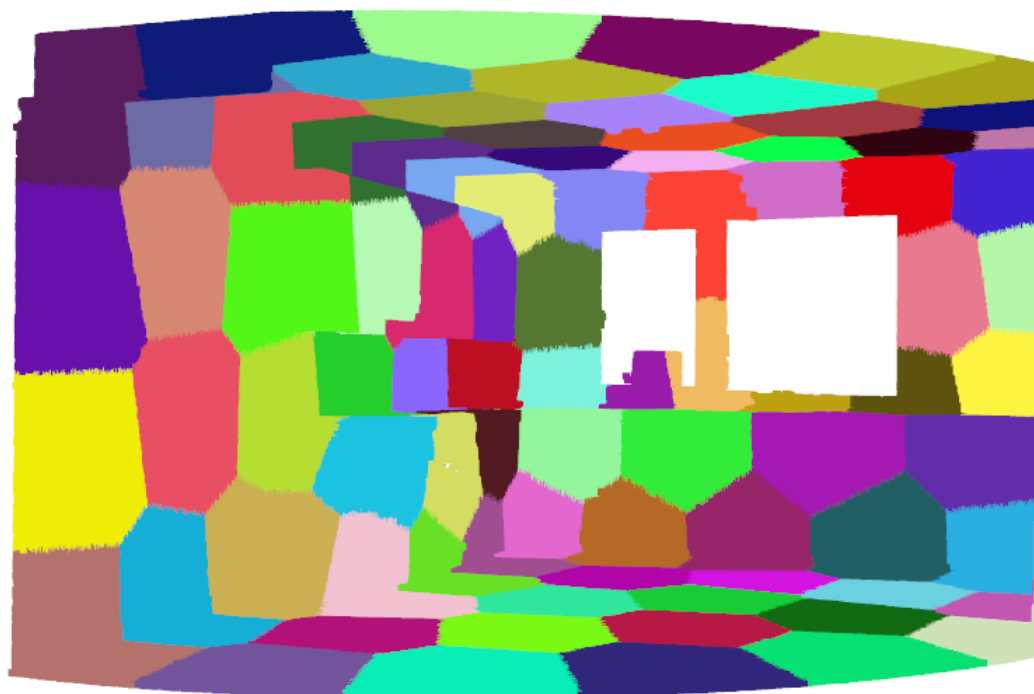
K-MEANS sme aj v tomto prípade nastavili na preurčený počet 10 zhukov, no taktiež pri DBSCAN sme dosiahli omnoho lepšiu zhukovaciú vlastnosť. Z toho dôvodu uvedieme ešte experiment, kde preurčený počet zhukov pre K-MEANS nastavíme na 90.

### 4.3 K-MEANS aplikovaný na dáta z Kinectu (90 zhlu- kov)



Vidíme že ani pri zvýšení počtu zhlu-  
kov z pôvodných 10 zhlu-  
kov na 90,  
sme nedosiahli výsledky blížiac sa ku algoritmu DBSCAN.

#### 4.4 K-MEANS aplikovaný na dáta z internetu (90 zhlukov)



K-MEANS aplikovaný na mračno bodov stiahnuté z internetu nepreukázal dostatočnú zhlukovacu schopnosť ani pri preurčenom počte 90 zhlukov.