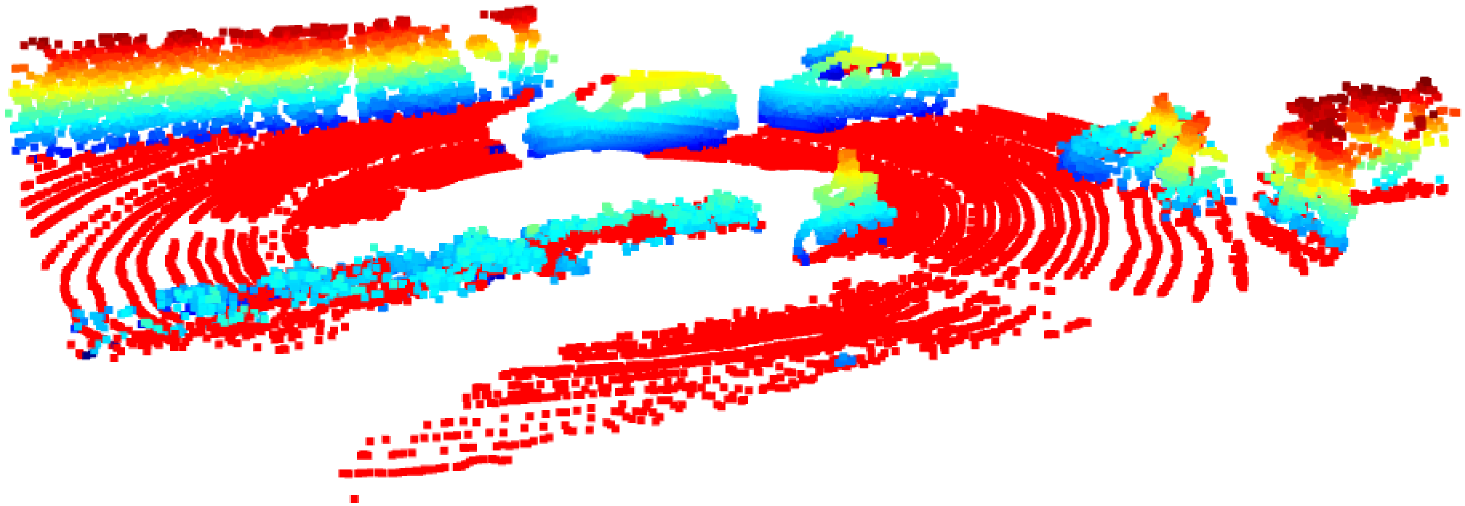
ELTE | FACULTY OF INFORMATICS

3D Point Cloud processing and analysis
Clustering and Point Cloud Features

Massinissa Aouragh:
Faculty of Informatics, Department of Artificial Intelligence
Robert Bosch Kft
m2j7au@inf.elte.hu

Data Clustering

Dividing unlabeled data into different clusters, or partitioning the data points into different classes according to a particular criterion, such that similar data points fall in the same class or cluster

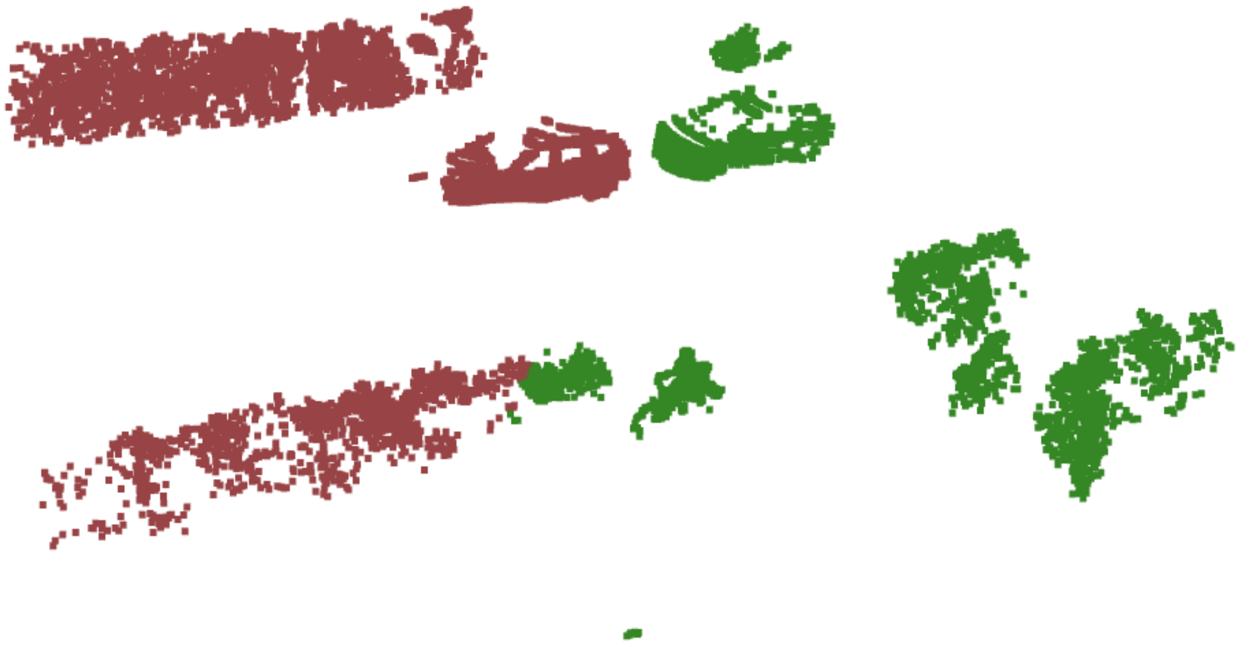


K-mean clustering

- K-Means Clustering is a widely used unsupervised machine learning algorithm
- Partition a set of n data points into k clusters, where k is a user-specified number
- The algorithm works by iteratively assigning each data point to one of the k clusters
- Minimizing the sum of squared distances between the data points and their respective cluster centroids
- The cluster centroids are calculated as the mean of the data points in each cluster and are updated after each iteration of the algorithm

K-mean

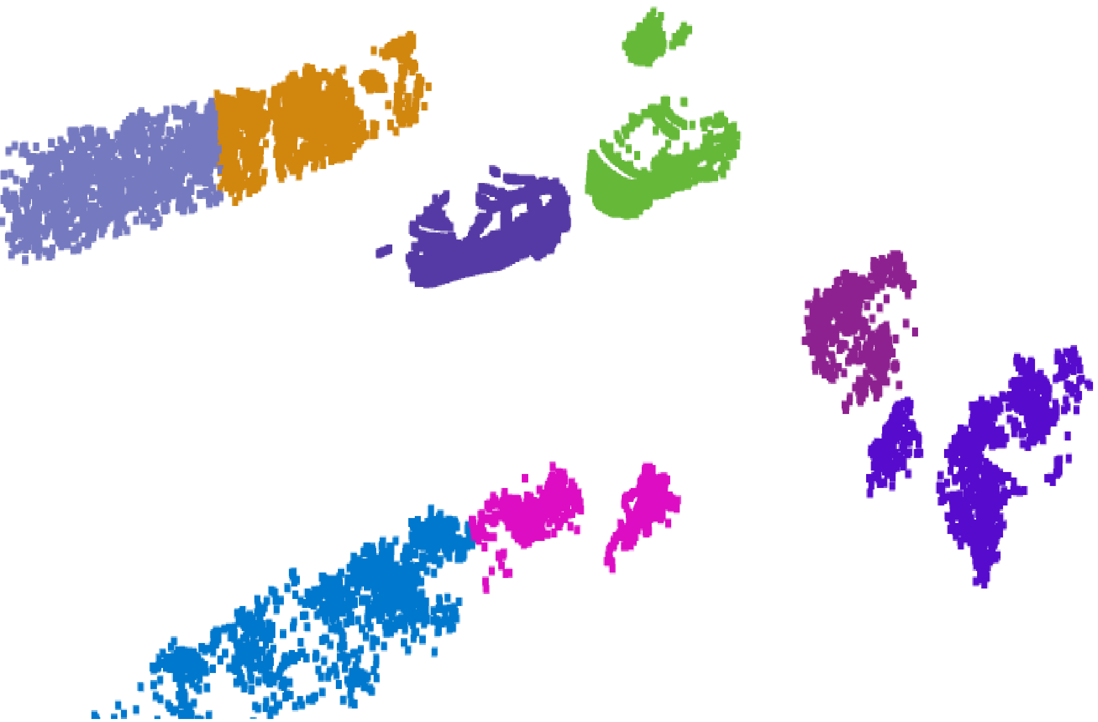
K=2



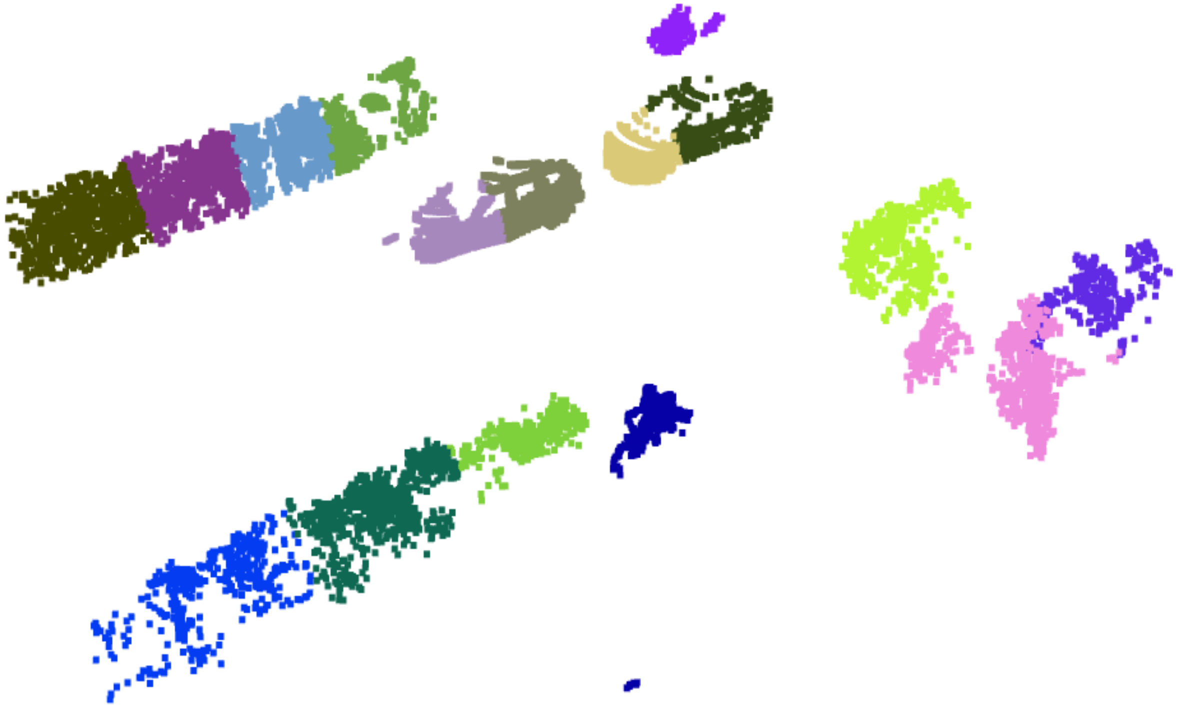
K=4

K-mean

K=8



K=16



Hierarchical clustering

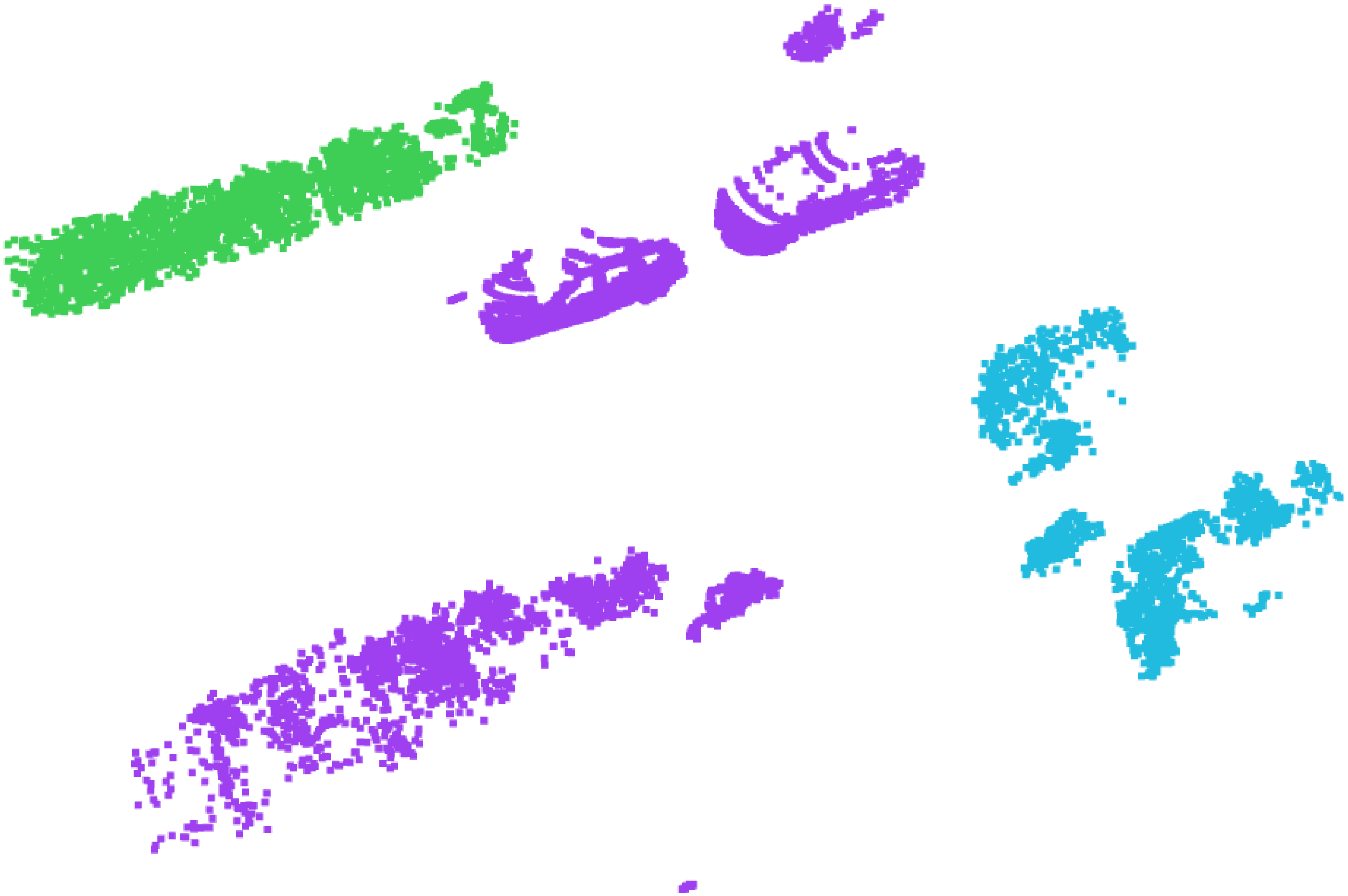
- Hierarchical clustering is a type of unsupervised learning that is used to group data points into a hierarchy of clusters
- It builds a tree-like structure that represents the hierarchical relationships between the clusters (dendrogram)

Hierarchical clustering

- Agglomerative hierarchical clustering, start with each data point as its own cluster and then merge the closest pairs of clusters until we reach a desired number of clusters, or the desired stopping criterion is met.
- Divisive hierarchical clustering, we start with all data points in a single cluster and then repeatedly split the clusters until each data point is in its own cluster or the desired stopping criterion is met.

Agglomerative hierarchical clustering

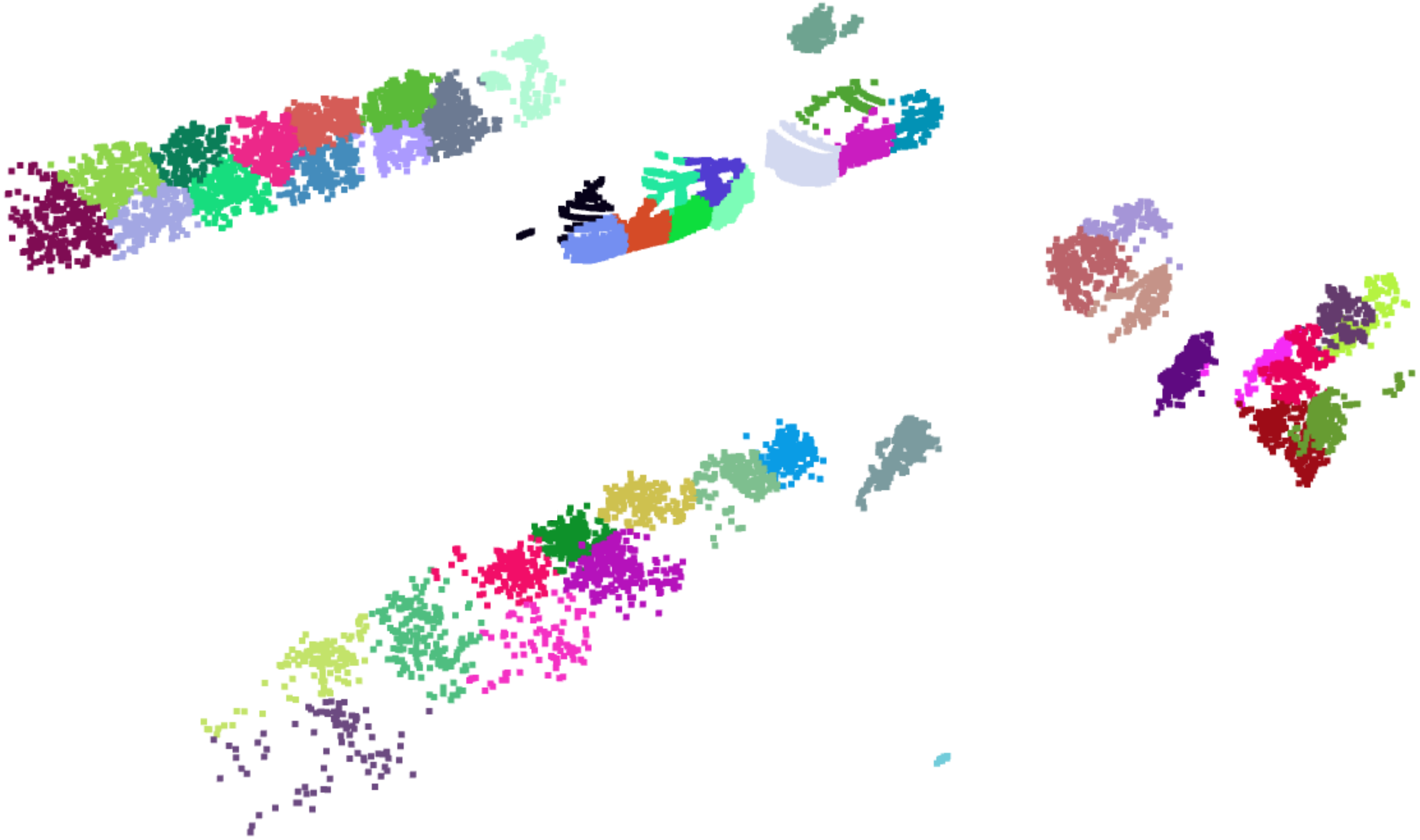
Stop at K=3



Affinity propagation

- Affinity Propagation is a clustering algorithm used to cluster data points into multiple groups based on their similarity. It was introduced by Brendan J. Frey and Delbert Dueck in the paper “Clustering by Passing Messages Between Data Points” in 2007
- Does not require the number of clusters to be specified, it iteratively adjust the responsibilities and availabilities between data points to determine the number of clusters
- Computationally expensive

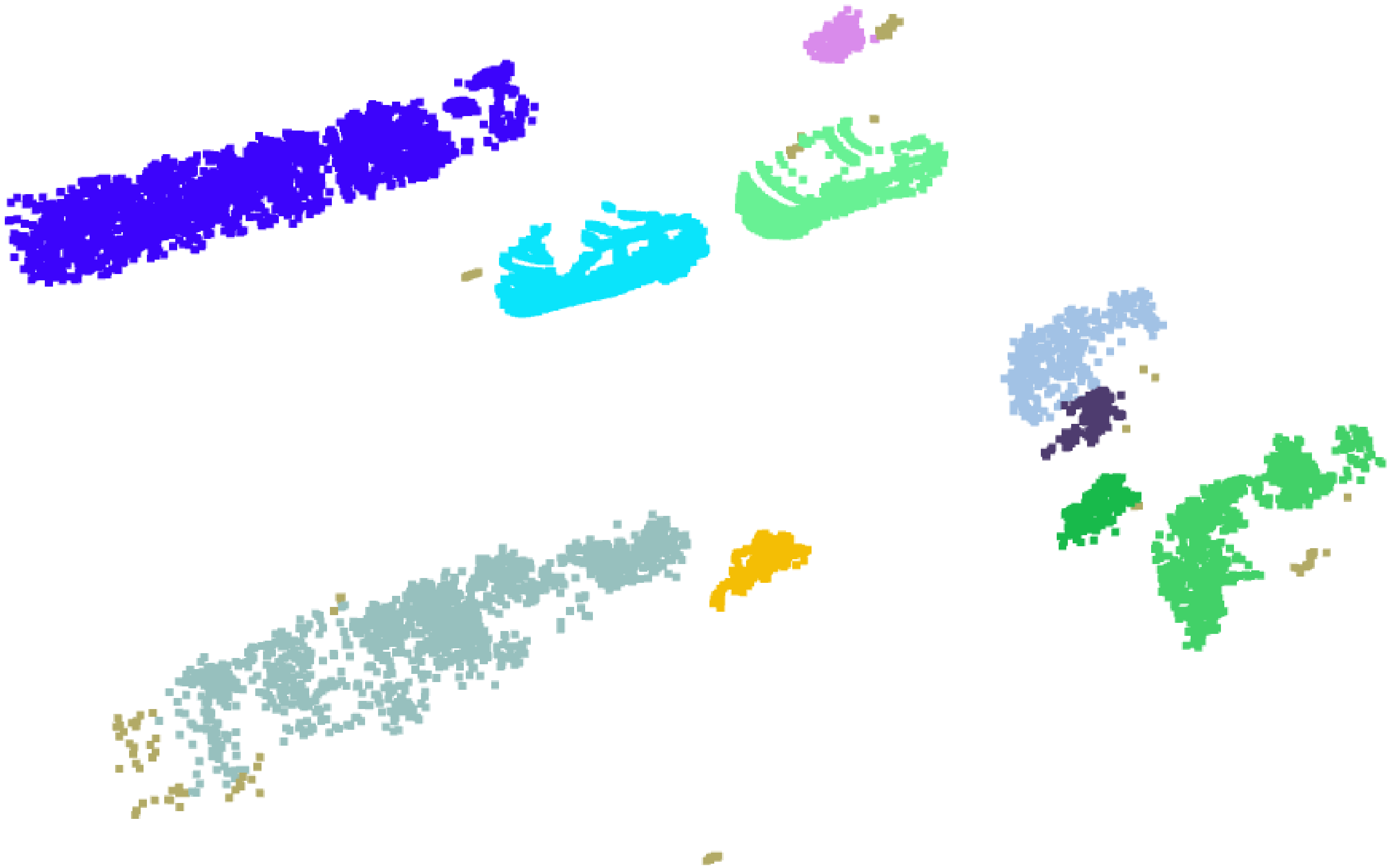
Affinity propagation



DBSCAN

- Density-Based Spatial Clustering of Applications with Noise is a density-based clustering algorithm
- Unlike other clustering algorithms, such as K-Means or Hierarchical Clustering, DBSCAN does not require the number of clusters to be specified beforehand
- It uses a density-based approach to identify clusters of similar data points
- It starts by selecting a random data point and then finding all data points within a specified radius

DBSCAN



Further clustering algos

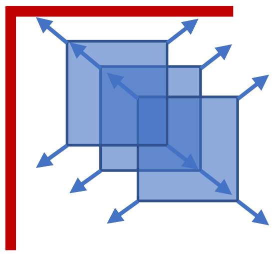
- Gaussian Mixture model
- Mean shift clustering
- Spectral clustering
- BIRCH Balanced Iterative Reducing and Clustering using Hierarchies
- OPTICS Ordering Points To Identify the Clustering Structure

Features

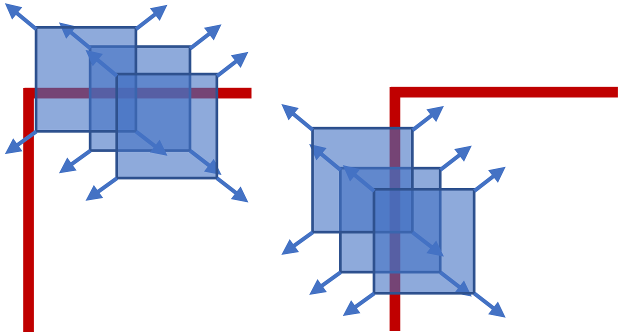
- Elements that provide information about the content of an image or a certain region of an image
- May be edges, corners or flat regions

Harris Corner Detector

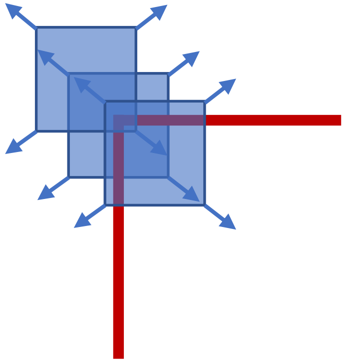
- Introduced by Chris Harris and Mike Stephens in 1988
- Harris' corner detector takes the differential of the corner score into account with reference to direction directly



Flat region no change in all directions



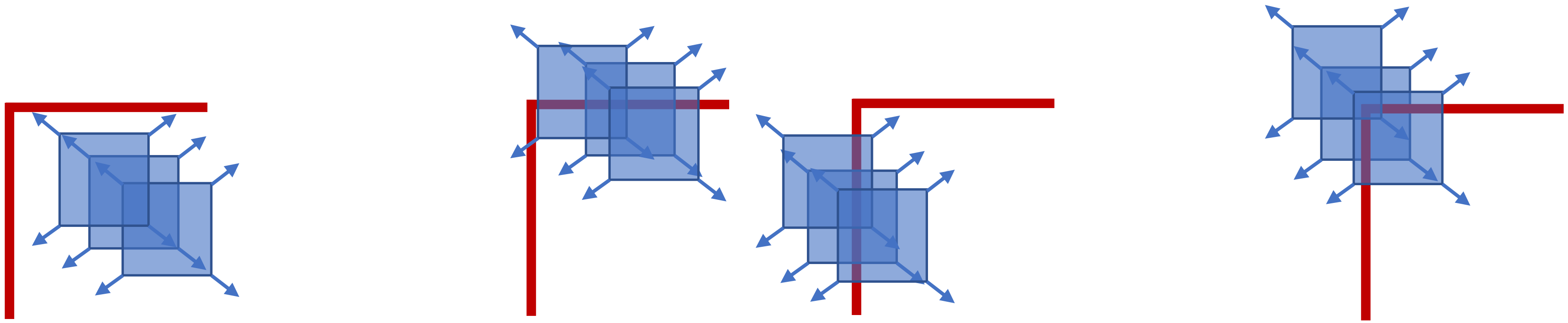
Edge no change in edge direction



Corner change in all directions

Harris Corner Detector

- We create a shifting window around pixel
- We calculate the sum squared difference of the pixel values before and after the shift



Flat region no change in all directions

Edge no change in edge direction

Corner change in all directions

Harris Corner Detector

- Sum squared difference error (SSD)
- The goal is to maximize the error for corner detection

$$E(u,v) = \sum_{(x,y)\in W} [I(x+u, y+v) - I(x,y)]^2$$

$$I(x+u, y+v) \approx I(x,y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Harris Corner Detector

$$E(u,v) \approx \sum_{(x,y)\in W} [I_x u + I_y v]^2$$

$$\approx Au^2 + 2Buv + Cv^2$$

$$E(u,v) \approx \sum_{(x,y)\in W} [I_x u + I_y v]^2$$

$$\approx Au^2 + 2Buv + Cv^2$$

$$A = \sum_{(x,y)\in W} I_x^2 \qquad B = \sum_{(x,y)\in W} I_x I_y \qquad C = \sum_{(x,y)\in W} I_y^2$$

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$E(u,v) \approx [u \quad v] \left(\sum \begin{bmatrix} I_x{}^2 & I_x I_y \\ I_x I_y & I_y{}^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Corner Detector

$$M = \sum_{x,y \in \mathrm{W}} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- Flat region:

$$M = \sum_{x,y \in \mathrm{W}} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- Vertical or horizontal edge:

$M = \sum_{x,y \in \mathrm{W}} \begin{bmatrix} 0 & 0 \\ 0 & I_y^2 \end{bmatrix}$ or $M = \sum_{x,y \in \mathrm{W}} \begin{bmatrix} I_x^2 & 0 \\ 0 & 0 \end{bmatrix}$

Harris Corner Detector

- The eigenvectors of M show the direction of the fastest changes
- λ_1, λ_2 are both small for a flat path
- λ_1, λ_2 are both large for corner
- $\lambda_1 \gg \lambda_2$ for a vertical edges $\lambda_2 \gg \lambda_1$ for a horizontal edge

Harris Corner Detector

- Response function R is calculated for each pixel :

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(M) - k * tr(M)^2$$

- Where $k \in [0.04, 0.06]$ is set empirically
- After normalization, pixels with R > τ are corners.

Harris 3D

- An extension for processing point clouds with/without intensity

$$E(u, v, w) = \sum_{x,y,z \in W} [I(x+u, y+v, z+w) - I(x, y, z)]^2$$

$$M = \sum_{x,y,z \in W} \begin{matrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{matrix}$$

Harris 3D

- Point cloud without intensity

$$E(u,v,w) = \sum_{x,y,z \in W} [f(x+u, y+v, z+w) - f(x,y,z)]^2$$

- Define a neighborhood of points (e.g.: KNN)
- Translate the point cloud so that the centroid of this neighborhood is placed at the origin
- Compute best fitting plane of the neighborhood (e.g.: PCA) Local surface $f(x,yz) \approx ax + by + cz + d = 0$

Harris 3D

- $f(x+u, y+v, z+w) \approx a(x+u) + b(y+v) + c(z+w) + d = 0$
$$= [x+u, y+v, z+w]n$$
- The first order approximation for E(u,v,w) is:
$$E(u,v,w) \approx [u,v,w]M \begin{bmatrix} u \\ v \\ w \end{bmatrix}, M = \sum_{x,y,z \in W} \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_x n_y & n_y^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix}$$
- $R = \det(M) - k * tr(M)^2$

Harris 6D

$$M = \sum_{x,y,z \in \Omega} \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z & I_x n_x & I_x n_y & I_x n_z \\ I_x I_y & I_y^2 & I_y I_z & I_y n_x & I_y n_y & I_y n_z \\ I_x I_z & I_y I_z & I_z^2 & I_z n_x & I_z n_y & I_z n_z \\ n_x I_x & n_x I_y & n_x I_z & n_x^2 & n_x n_y & n_x n_z \\ n_y I_x & n_y I_y & n_y I_z & n_x n_y & n_y^2 & n_y n_z \\ n_z I_x & n_z I_y & n_z I_z & n_x n_z & n_y n_z & n_z^2 \end{bmatrix}$$

Harris 3D: A robust extension of the Harris operator for interest point detection on 3D meshes

- Check the paper attached in PDF
- They improve the process for calculating the Harris operator for 3D meshes, making it robust to noise, change of tessellations and other transformations which deform the mesh structure such as local scaling, shot noise, presence of holes, just to name a few.
- They propose a novel method to define the neighborhood size of a vertex, depending on its surrounding structure.
- They give several options to select a few interest points using the information that the Harris operator provides.
- They present a comprehensive experimentation, trying to investigate the effect of different parameters choices and comparing our method with effective methods in the state of the art.