- they are robust to large number of variables and small samples.
- they can learn rather quickly also with few training samples.
- they belong to the non-parametric classification techniques, i.e., no parametric probability density functions are required.
- the kernel trick allows to find a linear classifier in some feature space.
- they deliver a unique solution.

3.3 Feature extraction

One important step in the classification procedure is the feature selection. The features are variables extracted from the raw data by the mean of linear or non-linear transformations and used to distinguish between the different classes. They must contain meaningful information which allows the separation of the data and provide accurate classification results with good predictive performances. The classification algorithms proposed in this work use geometrical features in the form of special attributes extracted from the available 3D coordinates of the points. For each point, a feature vector is determined based on the surrounding neighborhood points included in a restricted region around it. As a first step, the nearest neighbors within a specified location around each point have to be determined. In the next section, we discuss the methods used to find the nearest neighbors using a kD-Tree search algorithm.

3.3.1 Nearest neighborhood search

We define the nearest neighborhood N_j of a query point $\mathbf{x}_j$ as the set of all points which are found inside of a sphere of radius r centered at that point. This can be represented as:

$$N_j = \left\{ \mathbf{x}_i \mid ||\mathbf{x}_i - \mathbf{x}_j||_2 \leq r, i = 1, .., k \right\} \tag{3.16}$$

The radius r represents the maximum allowed distance from one neighbor to the query point.

In order to determine the closest points to a given point, all distances from that point to all the other points in the dataset have to be estimated and ordered [37]. The ones having a distance less then the threshold correspond to the set of the closest points. This is extremely costly. Therefore, the input dataset is divided into smaller chunks using a spatial decomposition technique such as $kD-trees$ [38]. kD-trees are data structures which are used to partition the k-dimensional space and store the points in the different partitions. An example of a kD-tree in the 2D space is shown in Figure 3.5. The closest point searches are then reduced to a limited space around the point of interest. In this

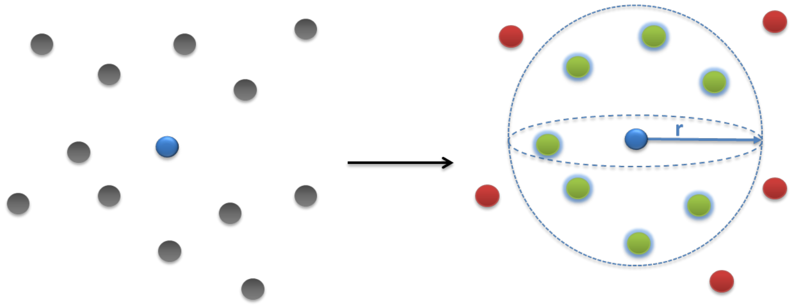

Figure 3.4: Illustration of the nearest neighborhood: the green points inside a sphere of radius r centered in the point of interested (in blue) are considered as the nearest neighbors to this point.

work, we deal with point clouds in $3D$. Therefore, a $3D-Tree$ is used to find the nearest neighbors of a specific point and to derive the neighbors within some radius specified by the user. The tree is obtained by partitioning the space with respect to the three dimensions.

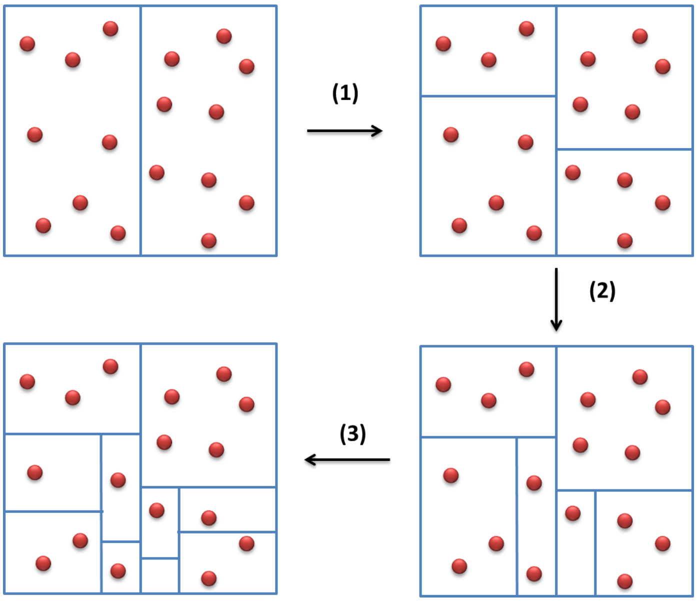

Figure 3.5: The partition of a set of points in the 2D space by a kD-tree: each split in one direction divides the space and the dataset into two distinct parts.

The obtained neighborhood is the base for the calculation of the features. In the literature, a multitude of features have been already used for 3D data classification. In this work, a set of features have been analyzed and selected to separate the LiDAR data. We report their contribution to the classification task.

In general, two of the most common geometric features are region covariance and estimated normals [39]. These two concepts have been used to derive our features which can be summarized into: eigenvalue based features, height based features and surface normals based features. They will be discussed in detail in the following sections.

3.3.2 Eigenvalue-based features

The spatial distribution of the neighbor points around a point of interest represents a good feature for the classification. This property can be expressed through the covariance matrix between the x, y, and z coordinates of the points in a local neighborhood. For a set of 3D points $\{\mathbf{x}_i, i = 1, ..., k\}$, the covariance matrix is a 3×3 matrix of the given set of points defined as follows:

$$M = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{3.17}$$

Where:

n: is the number of points considered in the neighborhood of the point
$\bar{\mathbf{x}}$: represents the 3D centroid of the nearest neighbors defined as:

$$M = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{3.18}$$

The decomposition of the covariance matrix into principal components results in a set of eigenvalues ordered in a decreasing order as follows: $\lambda_{\text{min}} < \lambda_{\text{mid}} < \lambda_{\text{max}}$.
If all three eigenvalues are nearly equal ($\lambda_{\text{min}} \approx \lambda_{\text{mid}} \approx \lambda_{\text{max}}$), the points have no structure and are spread out in 3D space with no dominant direction. This can be the case of vegetation.
If the two big eigenvalues have large values ($\lambda_{\text{max}} \approx \lambda_{\text{mid}} >> \lambda_{\text{min}}$), the points in the region are on a same plane with a normal parallel to the eigenvector with the smallest eigenvalue. This is the case for the roofs of the buildings and the flat part of the ground.
If the biggest eigenvalue is very large compared to the two others ($\lambda_{\text{max}} >> \lambda_{\text{mid}} \approx \lambda_{\text{min}}$), the points in this region are located on a line. This is the case for tree trunks or roof boundaries. A graphical representation of the signification of the eigenvalues is shown in Figure 3.6.

The following eigenvalue-based features can be derived [22]:

- *Anisotropy*:

$$A_\lambda = \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}}} \tag{3.19}$$

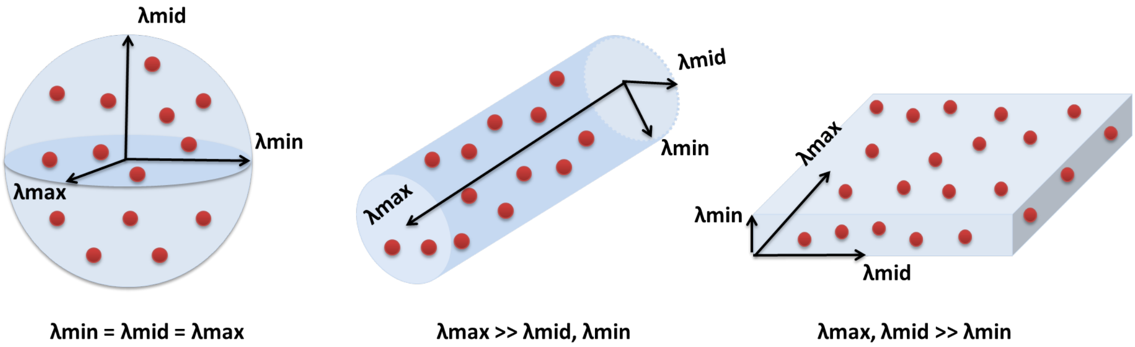

Figure 3.6: Illustration of the eigenvalues for the case of scatter, linear and planar points.

This represents the homogeneity of point distribution in three arbitrary perpendicular axes. This feature is useful to separate structures such as vegetation.

- *Linearity*:

$$L_\lambda = \frac{\lambda_{\text{max}} - \lambda_{\text{mid}}}{\lambda_{\text{max}}} \tag{3.20}$$

This permit to detect linear structures such as roof's bounding and building's edges.

- *Planarity*:

$$P_\lambda = \frac{\lambda_{\text{mid}} - \lambda_{\text{min}}}{\lambda_{\text{max}}} \tag{3.21}$$

This feature has high values for ground and building's roofs.

- *Sphericity*:

$$S_\lambda = \frac{\lambda_{\text{min}}}{\lambda_{\text{max}}} \tag{3.22}$$

It is used to detect points which are equally distributed in all three directions for points. This feature has big values for vegetation.

3.3.3 Height-based features

This kind of features is related to the altitude of the 3D points. We define two height-based features:

- $\Delta_z min$: is the height difference between the given point and the lowest point in the neighborhood. This feature has small values for ground points as they are more likely to be the lowest points in the cloud.

- Δ_z*min-max*: is the height difference between the highest and the lowest points in the neighborhood of a given data point. This feature has high values the vegetation points.
- σ_z: the height variance of the points in the neighborhood defined as:

$$\sigma_z^2 = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(z_i - \bar{z})^2} \tag{3.23}$$

Where:

n: is the number of points considered in the neighborhood of the point
$\bar{z}$: is the mean height in the neighborhood.

This feature has high values for vegetation.

3.3.4 Local curvature

The local curvature describes the rate of change in the shape and the variation of the surface at a point. The points belonging to planar surfaces such as the middle of a building's roof have a curvature value close to zero. In contrast, trees have higher curvature values as they belong to wildly varying surfaces. The surface variation around a given point **x** can be derived from the eigenvalues λ_j of the covariance matrix C as follows:

$$curvature = \frac{\lambda_{\text{min}}}{\lambda_{\text{min}} + \lambda_{\text{mid}} + \lambda_{\text{max}}} \tag{3.24}$$

In the next two chapters, the two classification approaches which have been developed are described and analyzed. Chapter 4 deals with the point-based classification approach. Chapter 5 discusses the object-based classification method used to improve the performances of the first one. Finally, the performances of the two methods is evaluated and a comparison between them is done.