TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

DIPLOMARBEIT

Comparison of Selected Segmentation Algorithms of 3D Point Clouds

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Geodäsie und Geoinformation

eingereicht von

Sebastian Flöry
Matrikelnummer 00826399

ausgeführt an der Forschungsgruppe Photogrammetrie (E120.7)
am Department für Geodäsie und Geoinformation
der Fakultät für Mathematik und Geoinformation an der technischen Universität Wien

Betreuung:
Univ.Prof. Dipl.-Ing. Dr.techn Norbert Pfeifer
Projektass. Dipl.-Ing. Dr.techn. Johannes Otepka-Schremmer

Wien, 12th September 2019 _____ _____

(Unterschrift Verfasser) (Unterschrift Betreuer)

Erklärung zur Verfassung der Arbeit

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht. Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Diplomarbeit eingereicht.

12th September 2019

(Sebastian Flöry)

Acknowledgements

First up a big thank you to Johannes Otepka-Schremmer and Norbert Pfeifer. You both supported me from the beginning when I approached you with the idea for this thesis, helped me at every point and you did not loose your patience when things took longer than expected. Although it might have seem to both of you that at some point I did loose focus, I always knew that, no matter what, I could count on your help. Thank you very much!

Of course without my parents supporting me at every point I would not have had the opportunity to study at an university in the first place. Thank you for giving me this opportunity. Besides my parents I also have to mention my brother Simon who voluntarily offered to proofread this thesis. I hope it wasn't too terrible. Thanks!

Through the various positions I held in different companies during my studies, I was able to experience many incredible moments and meet amazing people with whom friendships developed. Without this, I am sure, that I would never have completed my studies. So, without naming anyone personally, thank you for every conversation we had, brilliant ideas we came up with after some beers (and brilliant they were) and every hour of overtime we worked together. I do not want to miss any of it.

Last but absolutely not least I want to thank my girlfriend Lena. You supported me since the first day and had to put back in the recent months because beside my work and writing this thesis there was not much time for other things. Now I have more time again to get on your nerves. Promised!

It has been a long journey...

Abstract

Segmentation is an important step in the processing pipeline of 3D point clouds. It can be used to identify and extract individual objects or build the base for a subsequent classification. Within this work two methods, so far mainly used in the field of computer vision, are extended to the segmentation of 3D point clouds.

The first method, based on the minimum spanning tree of the neighborhood connectivity graph, shows promising properties enabling segmentation on an object-based level. The second method, SLIC (Super Linear Iterative Clustering), is designed to create a strong over-segmentation which can be used to reduce the amount of data in the first place.

To evaluate both approaches a new metric, termed completeness, is introduced. In contrast to other commonly used metrics like the number of segments or mean segment size, completeness measures the quality of the segmentation with respect to individual objects. The investigated datasets represent different scenes (urban/rural), show different point densities and contain objects of different size, shape and color.

All conducted tests show that both approaches are suitable for the segmentation of 3D point clouds. While the graph-based method improves segmentation in inhomogeneous regions, SLIC is an useful option for point clouds with higher point densities. Particularly their combination seems to be an interesting option for the segmentation of more complex scenes.

Kurzfassung

Segmentierung ist ein wichtger Schritt in der Analyse von Punktwolken. Sie wird sowohl zur Identifizierung und Extraktion von einzelnen Objekten, als auch als Basis für objektbasierte Klassifikation verwendet. In dieser Arbeit werden zwei Methoden, welche zu den meist zitierten Segmentierungsansätzen für Bilder zählen, auf 3D Punktwolken erweitert und evaluiert.

Der erste Ansatz, welcher auf dem Minimum Spanning Tree des Nachbarschaftsgraphes basiert, verfügt über interessante Eigenschaften welche ihn zu einem vielversprechenden Ansatz zur objektbasierten Segmentierung machen. Die zweite Methode, SLIC (Super Linear Iterative Clustering), resultiert in einer starken Übersegmentierung der Punktwolke. Die extrahierten Regionen (so genannte 'Superpixel') dienen zum einen zur Datenreduktion, können aber auch direkt für eine Klassifikation verwendet werden.

Für die Evaluierung werden ausgewählte Referenzdatensätze verwendet. Als Metrik wird ein neues Maß, 'completeness', eingeführt, welches auf den einzelnen Objekten basiert.

Alle durchgeführten Experimente zeigen, dass beide Ansätze für 3D Punktwolken vielversprechende Ergebnisse liefern. Vorallem mit dem graphenbasierten Ansatz können auch Regionen, welche inhomogenere Eigenschaften aufzeigen, segmentiert werden. Der zweite Ansatz, SLIC, scheint speziell für hoch aufgelöste Punktwolken eine interessante Option zu sein. Für komplexere Punktwolken führt vorallem die Kombination beider Methoden zu verbesserten Resultaten.

Contents

Class colour legend

- **GROUND (GRASS)**
- **GROUND (STREET)**
- **BUILDING (ROOF)**
- **BUILDING (WALL)**
- **VEGETATION**
- **CARS**
- **OBJECTS**

Nomenclature

ALS	Airborne Laser Scanning
DIM	Dense Image Matching
DSM	Digital Surface Model
DTM	Digital Terrain Model
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
MLS	Mobile Laser Scanning
MST	Minimum Spanning Tree
RAG	Region Adjacency Graph
SFM	Structure From Motion
SLIC	Super Linear Iterative Clusters
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle

1. Motivation

3D point clouds have gained an important role in many fields of work and research. With the increasing availability of high resolution 3D point clouds in numerous fields the necessity for automatic solutions to process, analyse and visualize point clouds has emerged even further.

Besides the derivation of digital terrain models (DTM) [Kraus and Pfeifer, 2011] which are important base products for hydrological, geomorphological and archaeological analysis, many applications require the localization and identification of individual objects within 3D point clouds. As an example the automatic identification of traffic signs, location of water hydrants or the extraction of streets themselves might be of interest within urban scenes. The process of grouping points sharing some property is called segmentation.

Most objects vary in shape, color or texture making the extraction of whole objects rather complex and error prone. Therefore many segmentation methods are designed to either extract specific objects (f.e. flat surfaces) [Rabbani et al., 2006], detect clusters in an arbitrary feature space [Melzer, 2007] or fit geometric primitives [Schnabel et al., 2007].

While some object-based segmentation approaches exist, they either depend on multiple individual steps conducted in sequence [Vosselman, 2013] or are limited to point clouds acquired from specific systems [Pu et al., 2011]. Hence it seems natural to search for a method, which does not depend on a priori knowledge, works with data captured from different sensors and is flexible enough to segment complete objects.

One approach originating from the field of computer vision shows promising properties regarding these criteria. The graph-based method proposed by [F. Felzenszwalb and P. Huttenlocher, 2004] seems to offer the possibility to conduct segmentation at an object-based level. While some work has been done already to extend this approach to 3D point clouds, either the results were note evaluated qualitatively ([Sima and Nuchter, 2013], [Strom et al., 2010]) or rastered RGB-D images were used rather than the original point cloud [Ben-Shabat et al., 2018]. Therefore a detailed discussion including data from various sensors is not available yet.

Besides the extraction of objects (or parts of it), segmentation can also be used as a data reduction step. Point clouds with hundreds of millions or even billions of points are no longer a rarity. In the field of computer vision such approaches are generally referred to as superpixel algorithms, which have been successfully applied to high resolution satellite imagery [Csillik, 2017].

Visually meaningful smaller regions ('superpixels') adhering well to object boundaries are extracted. These regions allow calculating robust features and may serve as the basis for a subsequent segmentation or classification. One of the most cited and efficient approaches is called SLIC (Super Linear Iterative Clustering) proposed by [Achanta et al., 2012]. Especially its simplicity makes it interesting for the segmentation of 3D point clouds.

Both methods are extended to 3D point clouds within this thesis. Their performance is evaluated on the basis of various datasets acquired from different acquisition systems. In order to measure the quality of the obtained segments a new metric, termed completeness, is introduced.

1.1 Implementation

Both methods have been implemented in Python 3.x making heavy use of optimized libraries like *Numpy* [Walt et al., 2011] or *pandas* [Mckinney, 2010]. Data structures and routines which could not be optimized with available solutions have been implemented in *Cython* [Behnel et al., 2011]. The nearest neighbor search is conducted with the kd-tree implementation offered in *scipy* [Jones et al., 2001].

Both methods are verified for correctness using the respective image equivalents included in the *scikit-image* package [van der Walt et al., 2014] respectively the original implementations if available. In order to access the supervoxel implementation the *pclpy* package, offering python bindings for most modules of the Point Cloud Library (PCL) [Rusu and Cousins, 2011], is used.

The file format to store, exchange and whatever someone does with point clouds is the LAS file format (especially version 1.4 which allows the definition of additional user defined attributes). Reading and writing .las files with Python is done with the *laspy* library.

The reference datasets used for evaluation are available in various different formats. For the conversion of the datasets and estimation of normal vectors *opals* ([Pfeifer et al., 2014], [Otepka et al., 2012]) has been used.

1.2 Structure of this work

The next chapter gives an general overview over 3D point cloud acquisition methods, available segmentation methods and the evaluation techniques used in order to quantify the quality of the segmentation results.

In chapter 3 both segmentation methods extended to 3D point clouds are described in detail, followed by a short description of the reference datasets in the subsequent chapter.

Chapter 5 in detail presents the results of the conducted experiments.

2. Overview

Throughout this thesis 3D point clouds are referred to as unstructured collections of points representing samples of earth's surface and objects (vegetation, buildings) on it. Each point stores it's coordinates (X,Y,Z) in an arbitrary chosen coordinate system and optional attributes (colour, intensity).

The 3D point clouds used within this work were obtained using airborne laser scanning (ALS), mobile laser scanning (MLS) and Structure from Motion (SfM). Therefore in the following the basic principles of these capturing methods are outlined.

2.1 Laser Scanning

Laser scanners capture the surface of objects in a systemic manner using a laser source. The distance to each point of reflectance is calculated either directly based on the time of flight using the well known equation

$$d = \frac{c \cdot t}{2} \tag{1}$$

with c being the speed of light and t the time passed between the emission and reception of the emitted pulse or based on phase measurements [Pfeifer and Briese, 2007]. Depending on whether the capturing platform is moving or spatially fixed, static (TLS) and kinematic (ALS, MLS) laser scanning are distinguished.

2.1.1 Terrestrial Laser Scanning

For terrestrial laser scanning the scanner is mounted on a static platform (generally a tripod) which does not move during the acquisition. Therefore all points obtained from one scan position have the same origin and orientation, generally referred to as exterior orientation.

Due to the limited field of view of the scanner itself and occlusions caused by the environment, in general it is necessary to scan objects from multiple positions. In order to combine point clouds obtained from different positions the relative orientation has to be established. This can either happen directly based on points in overlapping areas (e.g. ICP [Rusinkiewicz and Levoy, 2001]) or based on selected (artificial) tie points visible from multiple positions.

2.1.2 Airborne Laser Scanning

In the case of ALS the laser scanner is mounted on an airplane (helicopter or UAV). Due to the movement of the aircraft the exterior orientation of the laser scanner changes during the scanning process. Therefore it is necessary to obtain the position and orientation of the sensor platform over time. This can be achieved by using a combination of a global navigation satellite system (GNSS) and an inertial measurement unit (IMU).

Due to the acquisition geometry, recorded points are mainly located on the top surface pointing towards the sky leading to less information on vertical structures (Figure 2.1c, d). In forested areas ALS partly penetrates through the vegetation due to small gaps in the canopy capturing points within the vegetation and on the forest floor.

ALS is mainly used for large scale topographic data acquisition with applications ranging from forestry [Vauhkonen et al., 2014], archaeology [Doneus et al., 2008] to building extraction [Dorninger and Pfeifer, 2008].

2.1.3 Mobile Laser Scanning

While the basic principle for MLS is the same as for ALS, the main difference lies in the acquisition geometry. The laser scanner is mounted on a vehicle driving directly through the area of interest leading to a very high point density around the vehicle. In contrast to ALS, vertical structures of buildings (Figure 2.1a, b) are captured in higher detail.

MLS has been succesfully used for the monitoring of railways [Yang and Fang, 2014], in forestry [Liang et al., 2014] and for the identification and extraction of urban objects [Yang and Dong, 2013]. Especially the high point density combined with the acquisition geometry, allows, compared to ALS, the identification of much smaller objects in an urban scenery like street poles, curbs or traffic signs.

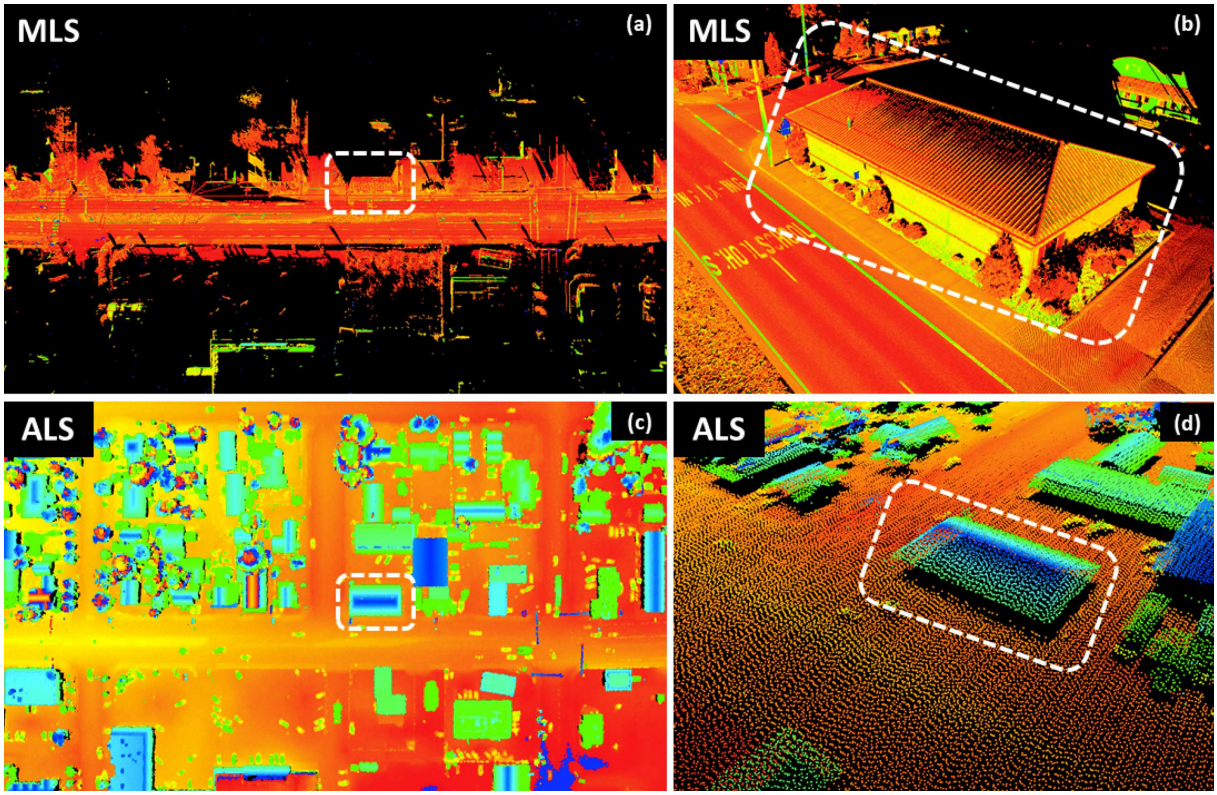

Figure 2.1: Comparison between MLS and ALS for the same area. Figure taken from [Che et al., 2019]

2.2 Structure from Motion

Structure from Motion is a highly automated method to reconstruct objects in 3D from multiple image sequences. Based on local feature points (e.g. SIFT [Lowe, 2004]) extracted in each image, the relative orientation between the images is established automatically by matching corresponding feature points. As the matching of the feature points is error prone, additional statistical (e.g. RANSAC [Fischler and Bolles, 1981]) and geometrical tests are used to remove outliers.

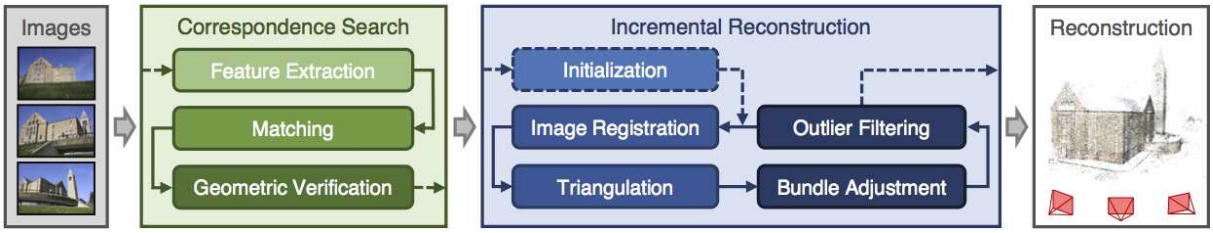

Figure 2.2: General pipeline for the 3D reconstruction of objects from images. Graphic taken from [Schonberger and Frahm, 2016]

In a subsequent dense image matching step (e.g. Semi Global Matching [Hirschmuller, 2008]) a dense 3D point cloud is generated. For each point radiometric information (e.g. colour) is extracted. This additional information is very useful for the identification of individual objects.

In case of aerial images in vegetated areas only the top canopy layer is visible. Therefore no points on the actual ground will be captured. Especially for the derivation of digital terrain models this represents a major limitation compared to ALS.

2.3 Attributes

Besides the 3D coordinates (X,Y,Z) each point can store additional attributes (features) describing properties of the object of reflectance. Depending on the acquisition method different attributes are available. [Otepka et al., 2013] propose 4 levels for the categorization of attributes:

Level 0: generated coordinates and other measurements directly recorded by the sensor
Level 1: improved coordinates from further georeferencing (e.g. strip adjustment)
Level 2: features computed within the neighbourhood of a point (e.g. normal vectors)
Level 3: features obtained by a combination with other data sources

Colour and normal vectors are important attributes used within both segmentation methods. While the availability of radiometric information is dependent on the acquisition method, normal vectors describing the local surface properties of each point can always be estimated from the points themselves.

Based on selected neighbours (according to a neighbourhood definition) of each point P_i the best fitting, in the sense of least squares, local tangent plane is estimated. Besides the normal vector n_i the standard deviation of the estimated plane expressed as s_0 is stored as well. For smooth surfaces like roofs s_0 will be small as the deviation of the points from the plane will be small. In vegetation or other rough areas, the strong variation of the points from a fitted plane lead to a higher s_0.

2.4 Neighbourhood Systems

The selection of spatial coherent points (further referred to as neighbours) in 3D is crucial for many calculations including the estimation of normal vectors or the definition of the neighbourhood connectivity graph. For images this spatial coherence is naturally defined by the pixels arranged on a grid (Figure 2.3a):

4-connected: all pixels sharing an edge with the central pixel
8-connected: all pixels sharing an edge or corner with central pixel

3D point clouds are generally unstructured leading to the definition of various neighbourhood systems. Especially point clouds obtained from terrestrial or mobile laser scanning, due to the acquisition geometry, show very inhomogeneous point distributions. Following [Filin and Pfeifer, 2005] an optimal neighbourhood should fulfil the following criteria:

1. rotation and translation invariant
2. keep the original structure
3. possibility to geometrically limit the neighbourhood

In the following commonly used neighbourhood systems for unstructured 3D point clouds are discussed and evaluated in respect to the criteria listed above.

2.4.1 Voxel Grid

Voxels can be seen as the extension of pixels to 3D. An unstructured point cloud is represented as a voxel grid by superimposing a regular 3D grid (with defined cell size) over the unstructured point cloud. For each cell a representative point from the point cloud (e.g. mean coordinates) is chosen. The usage of voxels leads to a definition of neighbourhood similar to the 2D case (Figure 2.3b):

6-connected: all voxels sharing a face with the central voxel
12-connected: all voxels sharing a face or edge with the central voxel
26-connected: all voxels sharing a face, edge or corner with the central voxel

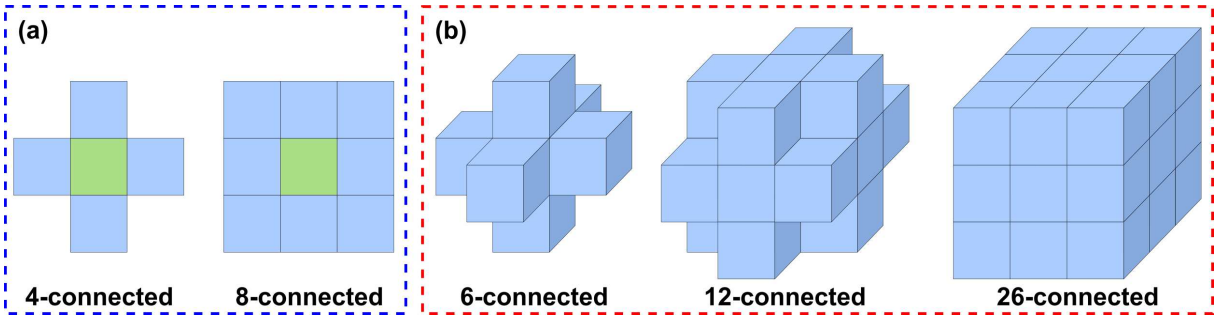

Figure 2.3: Neighbourhood definition for 2D (blue box) and 3D (red box) grid.

Voxel grids are neither invariant to translation nor rotation. Furthermore the original data structure is lost as the original points may be represented by a single representative point. If the original points contained additional attributes one needs to decide in which way the attribute of the representative point is calculated for each voxel.

2.4.2 Nearest Neighbours

2.4.2.1 K nearest neighbours

For each point the k nearest points are used as neighbours. No distance criterion limits the spatial distance. Especially for point clouds with varying point density (MLS, TLS) very distant points will be considered as neighbours although they might correspond to complete different objects. Furthermore neighborhoods based on the k nearest neighbours are not symmetrical per definition. In order to overcome these limitations a fixed distance neighbourhood might be considered.

2.4.2.2 Fixed distance neighbourhood

Only points within a defined spatial extent are used. As distance measure generally the Euclidean metric is used

$$d_{3D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{2}$$

$$d_{2D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{3}$$

Neighbourhood definitions based on nearest neighbours are rotation and translation invariant. The original data structure is preserved. By considering all points within a fixed distance the neighbourhood is symmetrical and geometrically limited.

Further spherical and cylindrical neighbourhoods are distinguished. For spherical neighbourhoods all points within a fixed distance (sphere) are considered as neighbours. Cylindrical neighbourhoods use two different distance thresholds for the horizontal and vertical direction:

$$d_{hor} = d_{2D} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4}$$

$$d_{ver} = h \tag{5}$$

with h being a user-defined value. On the contrary to the definition solely based on the fixed number of neighbours this definition will lead to a massive amount of neighbours for very dense point clouds. Therefore the combination of the fixed number with a limited search radius combines the strength of both individual definitions.

2.4.2.3 Combination

Throughout this thesis the k-nearest neighbours in combination with a maximum search radius are used as neighbourhood definition.

By limiting the neighbourhood both spatially and with regard to the number of neighbours ensures a robust definition even for 3D point clouds with irregular point distributions. For very dense regions the number of neighbours limits the neighbourhood while for sparse areas the limited search radius prevents problematic neighbourhoods.

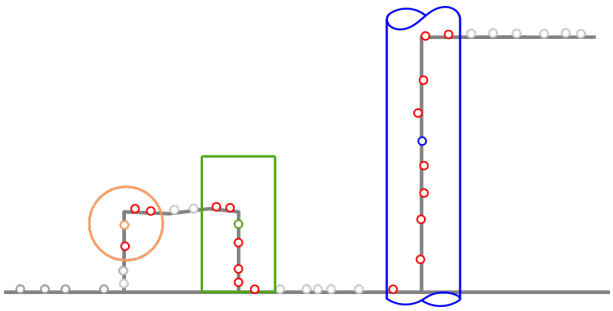

Figure 2.4: Spherical neighbourhood (orange); cylindrical neighbourhood with fixed height (green) and infinite height (blue)

2.5 Segmentation

In Figure 2.5 popular segmentation methods are shown roughly following the division made by [Nguyen and Le, 2013] and [Grilli et al., 2017]. As segmentation itself has a long tradition and research history in the area of computer vision many methods origin from image segmentation and have been adopted to the segmentation of 3D point clouds. Therefore in the following both the 2D (image) and 3D (point cloud) versions will be discussed.

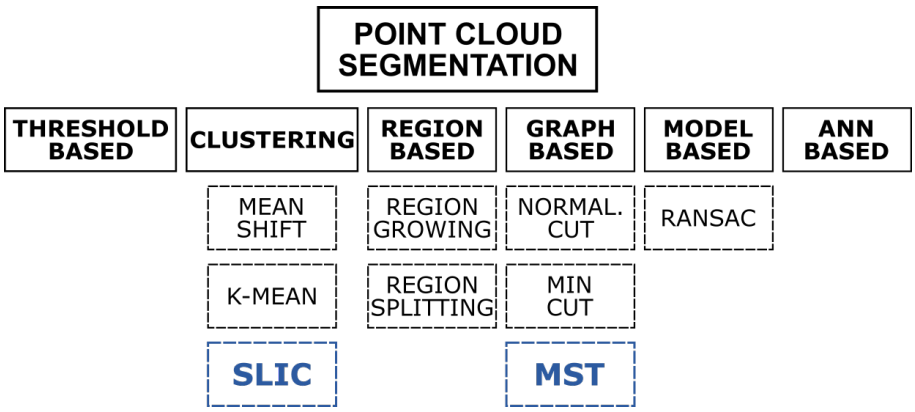

Figure 2.5: Schematic overview of 3D point cloud segmentation methods.

Before selected approaches are presented in detail, the terms segmentation and classification shall be discussed as both are often confused and used incorrectly.

Segmentation is the process of grouping spatial coherent points sharing some similarity. On the contrary classification is used to assign semantic information generally in the form of predefined classes (e.g. like whether a point represents ground, vegetation or buildings). Therefore solely based on segmentation no categorical information regarding the segments is available.

For the reference datasets the situation is different: These datasets contain a ground truth classification (each point was already assigned to a class) but no information regarding individual objects is available. Completeness, a metric used for evaluation, is based on the ground truth objects. Therefore it is necessary to extract, based on the ground truth classes, individual objects from the reference datasets. The method used is explained in detail together with the definition of completeness in the end of this chapter.

2.5.1 Region Based

Region based methods rely on the assumption that neighbouring points representing the same object or region will have similar attributes. Generally these methods can be divided into bottom-up (growing) and top-down (splitting) approaches.

2.5.1.1 Region Growing

For image segmentation, regions are grown from initial seeds adding neighbouring pixels to the same segment if some similarity criteria is fulfilled. This process is continued until all pixels have been visited. To prevent isolated small segments generally a minimum segment size is introduced. Region growing methods highly depend on the selection of the initial seeds. Rather unfortunate choices (e.g. on the edges of regions) affect the segmentation and will lead to segments not properly representing the regions of interest.

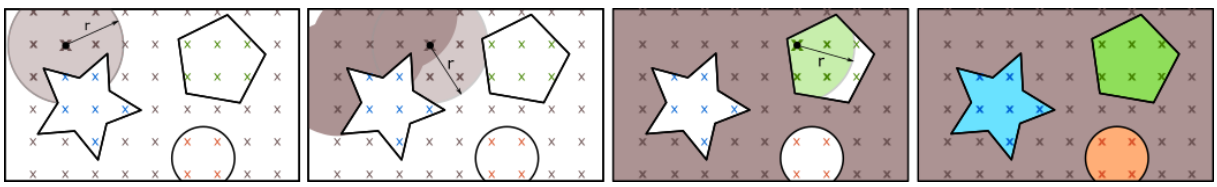

Figure 2.6: Seeded region growing taken from [Poechtrager, 2016]

Region growing methods can directly be extended to 3D point clouds. [Rabbani et al., 2006] proposed a method for extracting planar surfaces using normal vectors as smoothness constraint to avoid over-segmentation. [Poechtrager, 2016] implemented a highly efficient conditional region growing approach for 3D point clouds. Tiling of the original point cloud is used to conduct the region growing in parallel.

2.5.1.2 Region Splitting

In contrast to region growing methods this variant starts from one region representing the whole image or 3D point cloud. By iteratively subdividing regions into smaller subregions until some homogeneity criteria is fulfilled the final segments are obtained.

[Bruggisser et al., 2019] use an adaptive k-means approach to iteratively subdivide a 3D point cloud representing forested areas into homogeneous forest patches. A merging step after the actual partitioning is used to reduce over-segmentation.

2.5.2 Clustering

2.5.2.1 K-Means

K-Means is used to partition a selected feature space (colours, coordinates) into k clusters where each feature is assigned to the cluster with the closest mean, resulting in a partition of the feature space into Voronoi cells. The biggest disadvantage of k-means clustering is that the number of clusters must be known a priori, which is not the case for many applications.

Commonly starting from randomly selected cluster means the final clusters are computed in an iterative manner consisting of two main steps:

Assignment: Assign features to the cluster with the closest mean
Update: Calculate the new cluster means based on the new assignments

The iteration is stopped when no features are assigned to different clusters in two consecutive iterations.

K-Means is the core idea in the SLIC segmentation method proposed by [Achanta et al., 2012]. Starting from uniformly placed seeds every pixel of an image is assigned to the seed with the lowest distance in a combined features space (Eucldidean distance in colour and spatial domain). By limiting the search distance around each seed point this methods represents a highly efficient segmentation method, leading to a strong over-segmentation of the image (Figure 2.7).

2.5.2.2 Mean Shift

A non-parametric method for delineating arbitrarily shaped clusters was presented in [Comaniciu and Meer, 2002]. The feature space is interpreted as the empirical probability density function where dense regions correspond to local maxima. By finding the local maxima for each feature point using a technique called mean shift it is possible to delineate the clusters within the feature space. In contrast to k-means the number of clusters must not be known a priori.

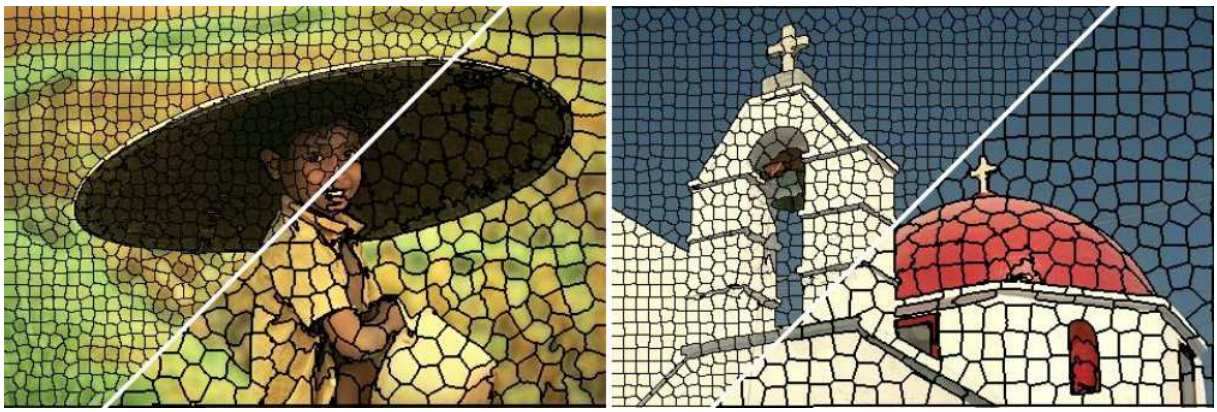

Figure 2.7: SLIC superpixels obtained from two different seed resolutions overlaid on two test images [Achanta et al., 2012]

[Melzer, 2007] applied mean shift segmentation to separate power cables from vegetation and the segmentation of an urban scene. [Xiao et al., 2019] used it to delineate individual trees in 3D point clouds derived from ALS.

2.5.3 Graph Based

Both images or 3D point clouds can be represented as a graph $G(V, E)$ where each pixel or point is represented as a node $v \in V$. Each edge $e(v_i, v_j) \in E$ of the graph G connects two adjacent nodes based on a selected neighbourhood system. If a weight $\omega(v_i, v_j)$ is assigned to each edge, G is an undirected weighted graph which serves as the basis for the graph based methods presented within this section.

The methods discussed belong to the more popular ones which have also been applied to the segmentation of 3D point clouds. A more complete discussion regarding graph based approaches to image segmentation can be found in [Peng et al., 2013].

2.5.3.1 Minimum Spanning Tree

Given an undirected weighted Graph $G(V, E)$ the minimum spanning tree is the path connecting all reachable nodes in the graph G with minimum weight. A widely used algorithm for finding minimum spannings trees was proposed by [Kruskal, 1956].

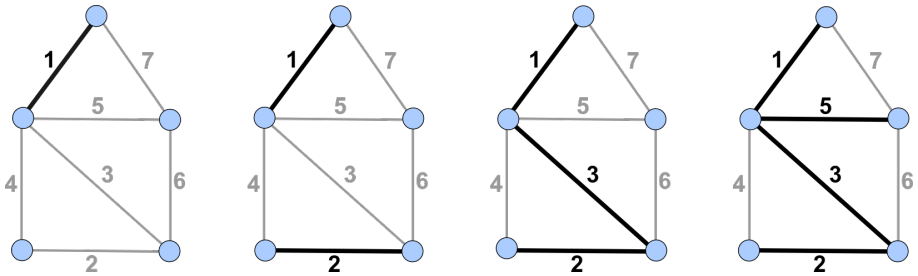

Figure 2.8: From left to right: Each step of the creation of the MST using Kruskal's algorithm

Starting with the edge having to lowest weight of G, Kruskal's algorithm greedily adds edges to the minimum spanning tree sorted in ascending order by weight. Only edges are considered whose inclusion won't create cycles within the minimum spanning tree.

The basic idea of the segmentation method proposed by [F. Felzenszwalb and P. Huttenlocher, 2004] builds upon minimum spanning trees using Kruskal's algorithm. Introducing a size-adaptive upper bound for the weight of edges added to the minimum spanning tree breaks it into multiple disconnected paths, each representing one segment of the image.

2.5.3.2 Graph Cuts

In graph theory a cut $C = (S, T)$ of a graph $G(V, E)$ is the partition of all nodes V into two subsets S and T. The cut set are the edges that need to be removed to separate the graph G into two separated connected components.

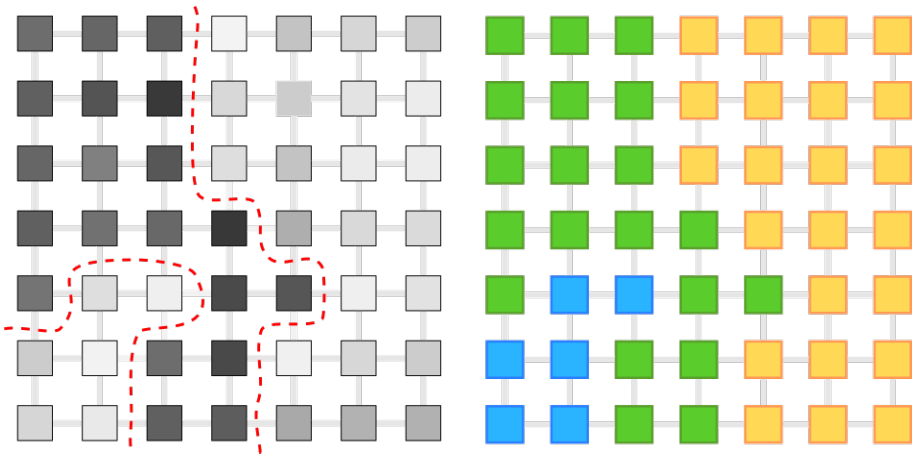

Figure 2.9: Image graph based on 4-connected neighbourhood; Tow possible cuts are illustrated as red dotted lines. Resulting image segmentation shown on the right.

The cut value is the sum of the weights ω of the cut set:

$$c(S, T) = \sum_{e \in C} \omega(e) \tag{6}$$

Finding the optimal cut set lead to the development of various methods:

Minimum Cuts

If the weights of an undirected weighted graph G represent the dissimilarity between the connected nodes the optimal cut will be the one that globally minimizes the weight of the cut set.

[Golovinskiy and Funkhouser, 2009] used minimum cuts to separate foreground objects (cars, traffic lights, street poles) from background objects (building) in mobile laser scanning data based on the approximate position of the object provided by the user.

Normalized Cuts

According to [Wu and M. Leahy, 1993] minimum cuts tend to cut small sets of nodes from a graph G. To overcome this problem [Shi and Malik, 2000] presented an improved method called 'normalized cuts'.

Instead of considering only the total weight of the cut itself, the cut value is related to all nodes in the graph:

$$N_{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{7}$$

$assoc(A, V)$ represents the weight of all edges connecting nodes in A to all nodes in G:

$$\text{assoc}(A, V) = \sum_{u \in A, t \in V} \omega(u, t) \tag{8}$$

[Hu et al., 2017] use normalized cuts to separate individual trees in a point cloud obtained using airborne laser scanning. To increase the efficiency of the segmentation the normalized cuts are calculated for the voxelized point cloud.

[Yao et al., 2009] combine mean shift with normalized cuts. Based on the segments extracted using mean shift, the region adjacency graph (RAG) is constructed. Normalized cuts are then calculated for the region adjacency graph.

2.6 Evaluation

For the evaluation of image segmentation algorithms multiple criteria have been developed. Among the most used are under-segmentation and boundary-recall. Developed for images, some of these metrics are not directly applicable to unstructured 3D point clouds due to the ambiguous neighbourhood definitions.

Boundary recall, for example, measures how well segment borders align with the borders of a ground truth segmentation. Using these metric for the evaluation of 3D point clouds fails, as borders highly depend on the used neighbourhood definition and are not as clearly defined as it is the case for images. In contrast, under-segmentation measures how much of a segment leaks across multiple ground truth segments and therefore is very well suited for 3D point clouds as well.

In the case of superpixel algorithms (e.g. SLIC) a common way to evaluate the performance of the segmentation is to plot either under-segmentation or boundary recall against the mean segment size or the number of segments. Both mean segment size and the number of segments do not include any information regarding how well (complete) ground truth segments are represented by the obtained segmentation. In order to overcome this, within this work an additional measure, termed 'completeness', is introduced.

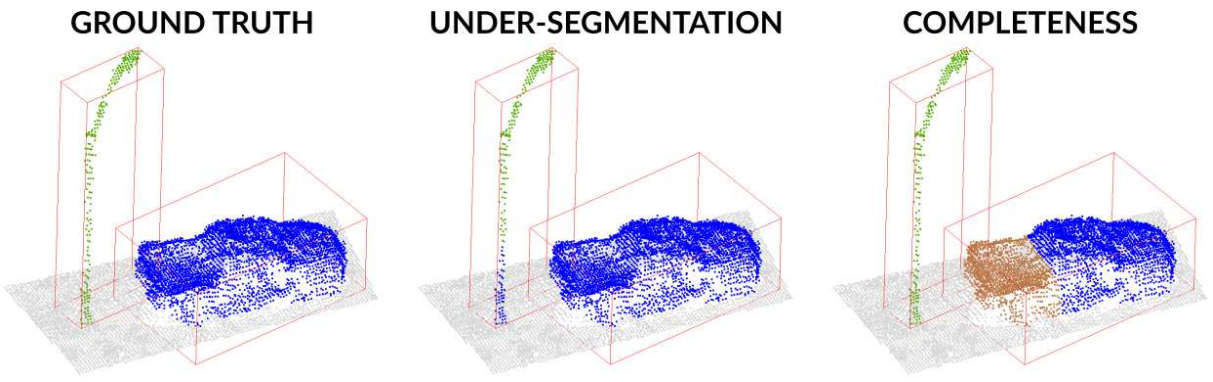

Figure 2.10: Examples of under-segmentation and completeness. Colours denote single segments; Individual objects are highlighted with red bounding boxes.

2.6.1 Under-segmentation

Under-segmentation measures the amount of points from segments crossing ground truth segments (Figure 2.10 middle). The blue segment includes points both from the car and street lamp. As more points belonging to the segment originate from the car, all points located on the street lamp will define the under-segmentation error for this segment. In general under-segmentation is defined as

$$\mathrm{UE} = \frac{1}{N}\left[\sum_{i=1}^{K} |P_{i,out}|\right] \tag{9}$$

with N being the number of all points in the point cloud and K the number of segments. $|P_{i,out}|$ are the points of the segment S_i that cross the ground truth boundary and is defined as

$$P_{i,out} = |S_i| - |P_{i,max(GT)}| \tag{10}$$

where $|S_i|$ is the size of the segment S_i and $P_{i,max(GT)}$ being the maximum number of points that come from one ground truth class within the segment.

If the segments shall serve as the basis for a subsequent classification step, under-segmentation error directly gives an upper-bound for the maximum achievable classification accuracy. Even if a classifier (SVM [Cortes and Vapnik, 1995], Random Forest [Breiman, 2001]) assigns the correct class labels to all segments, those points within each segment, which actually belong to a different class will be wrong.

2.6.2 Completeness

Besides evaluating if segments cross ground truth boundaries, the second metric represents how well segments describe individual objects in terms of completeness. Individual objects are defined within this work as connected points, based upon a neighbourhood graph, belonging to the same ground truth class. In order to extract those, all edges within the connectivity graph connecting points with different class labels are eliminated. By identifying all connected components within the graph the objects are extracted.

Figure 2.11: Extraction of individual objects. Left image shows the coloured point cloud; right image the individual objects extracted in individual colours.

The completeness for a single object O_i is defined as

$$C_i = \frac{|P_{i,max(SEG)}|}{|O_i|} \tag{11}$$

with $|O_i|$ being the size of the object and $|P_{i,max(SEG)}|$ being the number of points from the segment with maximum size contained in the object. Therefore a completeness of 1 indicates that all points within the object come from the same segment.

In order to derive the overall completeness the weighted mean is used where weights are based on the size of each object in relation to all points within each class (for the completeness for each class) and to all points within the point cloud (for the overall completeness). The motivation for the usage of the weighted mean shall be explained based on a simple example.

The sizes of objects contained in the Ankeny dataset vary strongly. While the largest component for the ground class has approximately 4 million points, smaller parts contain 50 points and less.

Lets assume there are 3 individual objects of different size: Object 1 has 100, Object 2 10000 and Object 3 1000000 points. Object 1 is perfectly segmented whereas Object 2 and Object 3 only by 50%. Using directly the arithmetic mean the overall completeness results in 66.6% and based on the weighted mean 50%. Therefore in order to account for the size of the objects as well the weighted mean must be used.

Relating the segment sizes to the individual objects is already a good measure for the quality of the segments. But in that way only the largest segment for each object is considered. This sometimes leads to the situation that completeness alone is not enough to describe the true quality of the segmentation. Therefore besides completeness later on also the number of segments and mean segment size are used as additional metrics.

3. Segmentation

Within this thesis two selected image segmentation algorithms are extended and evaluated for 3D point clouds. The first one is the 'Efficient Graph-Based Image Segmentation' proposed in [F. Felzenszwalb and P. Huttenlocher, 2004] further referred to as MST.

The second one, called 'SLIC' (Super Linear Iterative Clustering) was published by [Achanta et al., 2012]. Both methods are among the most cited image segmentation methods found in literature.

For each approach the original implementation is discussed, followed by a short review of related work conducted to extent the respective approach to 3D point clouds. Finally the methods implemented and evaluated within this work are presented.

3.1 MST

3.1.1 Original implementation

According to [F. Felzenszwalb and P. Huttenlocher, 2004] the main motivation for developing their method can be explained with the synthetic example shown in Figure 3.1. A human observer will distinguish three different regions in Figure 3.1 a:

R1: The large region on the left with a gradient fill.
R2: The smooth region on the right side with a hole.
R3: The noisy smaller region within R2.

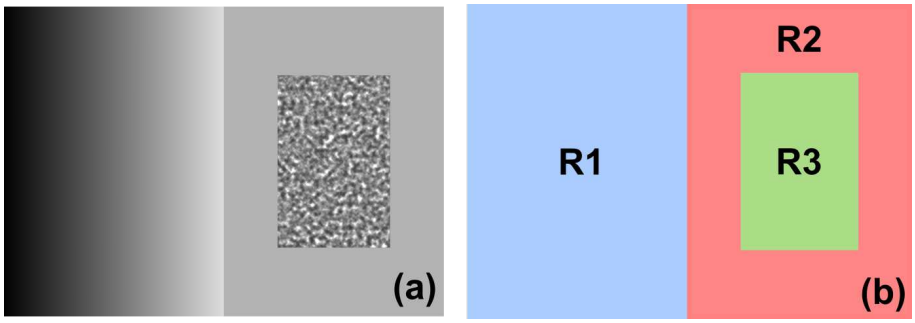

Figure 3.1: Synthetic example showing three distinct regions taken from [F. Felzenszwalb and P. Huttenlocher, 2004]

Especially the third region with its high local variance represents a major problem for most segmentation methods only considering local changes. As these methods only compare the differences in a pixel-wise manner (pixel by pixel), the strong fluctuations in the intensity of the gray values will most certainly lead to multiple small segments.

Based on the idea, that features within a segment shall be just more similar to each other than to features in other segments, two quantities are evaluated in each step of the segmentation:

$\mathbf{Int}(\mathbf{C_i})$ representing the variability within each segment, called **internal difference**.
$\mathbf{Dif}(\mathbf{C_i}, \mathbf{C_j})$ actually representing the difference across segments, called **difference between**.

In order to efficiently conduct the segmentation, the image is represented as a graph $G(V, E)$. Each pixel corresponds to a node $v \in V$. Each edge $e(v_i, v_j)$ connects two adjacent nodes (pixels). A weight $\omega(v_i, v_j)$ is assigned to each edge, representing the difference between them.

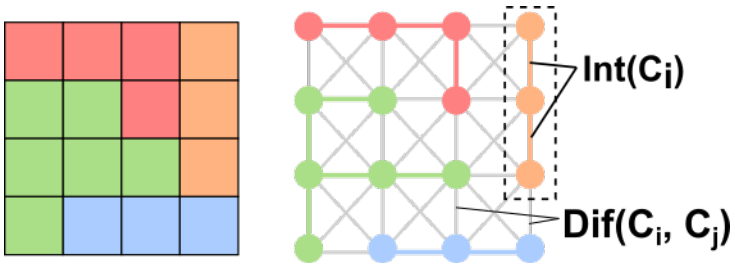

Figure 3.2: Graphical illustration of the internal difference and difference between. Left: segmented Image; Right: 8-connected graph representation; colours indicate individual segments

While for gray-scale images the intensity differences between adjacent pixels are used as weight,

$$\omega(a,b) = |I_a - I_b| \tag{12}$$

for colour images the Euclidean distance in the RGB colour space can be used:

$$\omega(a,b) = \sqrt{(r_a - r_b)^2 + (g_a - g_b)^2 + (b_a - b_b)^2} \tag{13}$$

Based on the graph G the internal difference (Int) of a segment is defined as the largest weight of its minimum spanning tree

$$\mathrm{Int}(C) = \max_{e \in MST} \omega(e) \tag{14}$$

Respectively the difference between (Dif) is defined as the weight of the edge with minimum weight connecting two segments

$$\mathrm{Dif}(C_i, C_j) = \min_{v_i \in C_i, v_j \in C_j} \omega(v_i, v_j) \tag{15}$$

In order to evaluate if two segments should be merged

$$D(C_i, C_j) = \begin{cases} true & \text{if } \mathrm{Dif}(C_i, C_j) \leq \mathrm{MInt}(C_i, C_j) \\ false & \text{otherwise} \end{cases} \tag{16}$$

is used. The minimum internal difference (MInt) is defined as the minimum of the internal difference plus some scale τ

$$\mathrm{MInt}(C_i, C_j) = \min(\mathrm{Int}(C_i + \tau(C_i), \mathrm{Int}(C_j + \tau(C_j)) \tag{17}$$

τ is used to control how much the internal difference (Int) and the difference between (Dif) must differ, in order to unite two segments. [F. Felzenszwalb and P. Huttenlocher, 2004] defined it as a function of the size of the segment $|C|$ and a scaling factor k:

$$\tau(C) = \frac{k}{|C|} \tag{18}$$

By defining τ based on the segment size $|C|$, $\mathrm{Dif}(C_i, C_j)$ can be larger for smaller regions. As the region grows, τ gets smaller and the regions have to be more similar. Introducing τ turns $\mathrm{MInt}(C_i, C_j)$ into an size adaptive criterion. Applying this size-adaptive criterion to the synthetic example shown in Figure 3.1 makes it possible that R3 is not split into multiple small segments.

Figure 3.3 shows the segmentation result for the MST approach using three different scales. Starting with a low scale (top right), medium scale (bottom left) and high scale (bottom right)

one can observe that segments in general strongly vary in shape and size. With increasing scale the number of segments is reduced as more different regions are merged. This can nicely be observed in the segmentation result for the highest scale where the gorilla is merged with the background whereas for the other scales the gorilla is preserved.

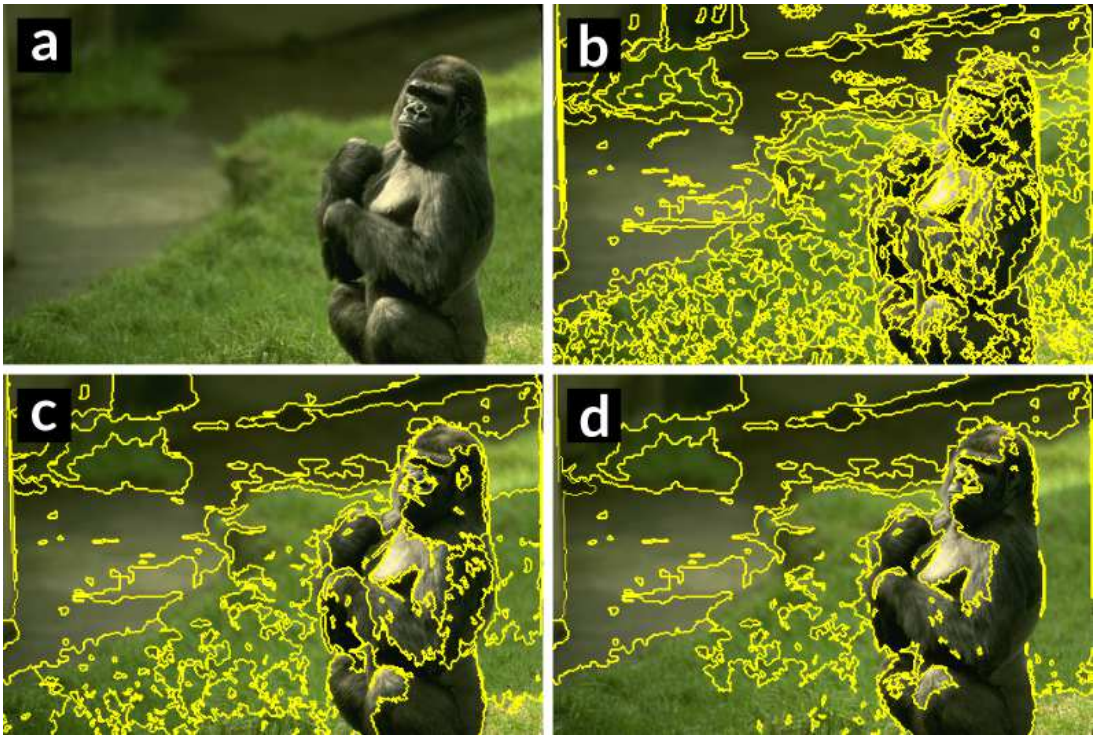

Figure 3.3: Original image (a) and segmentation results for different scales τ (b-d)

Algorithm 1: Felzenszwalb Segmentation

Result: Segmentation of the image

```
// actual segmentation
for each edge(v_i, v_j) in ascending-order do
    ω ← weight of current edge
    C_i ← component containing v_i
    C_j ← component containing v_j
    if C_i ≠ C_j then
        if ω ≤ MInt(C_i, C_j) then
            merge(C_i, C_j)
        end
    end
end
// remove segments smaller than a min_size (optional)
for each edge(v_i, v_j) in ascending-order do
    |C_i| ← size of component containing v_i
    |C_j| ← size of component containing v_j
    if |C_i| < min_size or |C_j| < min_size then
        merge(C_i, C_j)
    end
end
```

3.1.1.1 Post-Processing

The segmentation is directly based on the neighbourhood connectivity graph. This guarantees spatial connectivity of all pixels or points belonging to the same segment already during the segmentation. As segments vary strongly in shape and size in some situations small or isolated segments occur. To further remove these small isolated segments an optional post processing step is conducted after the initial segmentation.

Instead of comparing the weights of the edges, only the segment size is regarded. Any segment below a given minimum segment size will be merged no matter how large the difference between the adjacent segment is. In order to reduce the error introduced during this step, again the edges are sorted in ascending order by weight making sure that the segments will be merged with the most similar ones.

3.1.1.2 Related work

[Sima and Nuchter, 2013] segmented an indoor and outdoor scene obtained using a terrestrial laser scanner. The k-nearest neighbours were used to build the connectivity graph from the unstructured 3D point cloud. As weight they combined the Euclidean distance with the reflectance difference

$$\omega(a,b) = \underbrace{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}}_{\text{Euclidean distance}} + \underbrace{|I(a) - I(b)|}_{\text{reflectance difference}} \tag{19}$$

Combining two different metrics (in this case the Euclidean distance and reflectance difference) into one combined measure generally seems problematic. While the authors claim to obtain satisfying results, their judgement only relies on visual inspection of the segmentation results as no quantitative analysis was conducted.

In order to avoid the combination of two measures into one [Strom et al., 2010] used two separate weights during the segmentation:

$$\omega_{colour}(a,b) = \sqrt{(r_a - r_b)^2 + (g_a - g_b)^2 + (b_a - b_b)^2} \tag{20}$$

$$\omega_{normals}(a,b) = \operatorname{acos}(n_i \cdot n_j) \tag{21}$$

Only if both weights are below given thresholds, segments are merged. While this avoids combination of two completely different attributes, one has to decide which attribute shall be used for sorting the weights within the neighbourhood graph. As in the case of [Sima and Nuchter, 2013] no quantitative evaluation was conducted.

[Schoenberg et al., 2010] used a weighted combination of Euclidean distance, intensity and surface normals

$$\omega_{colour}(a,b) = \underbrace{k_e \cdot ||X_a - X_b||^2}_{\text{Euclidan distance}} + \underbrace{k_i \cdot ||I_a - I_b||^2}_{\text{intensity difference}} + \underbrace{k_n \cdot (1 - n_i \cdot n_j)}_{\text{normal difference}} \tag{22}$$

Besides combining different features into one measure, introducing additional weights rather over complicates than improves the segmentation. The additional weights need to be user defined, making it even harder finding the optimal set of values for the segmentation.

While all the presented works evaluated the quality of the segmentation results only visually, recently [Ben-Shabat et al., 2018] gave a more complete discussion regarding the choice of neighbourhoods and combination of attributes.

Various neighbourhoods and weight combinations are evaluated in terms of under-segmentation and boundary-recall on the NYU Depth V2 dataset [Silberman et al., 2012]. Their segmentation, referred to als PLCV, is based on the k-nearest neighbours and uses indiviudal weights similar to [Strom et al., 2010]:

$$\omega_{colour}(a,b) = \frac{\sqrt{(r_a - r_b)^2 + (g_a - g_b)^2 + (b_a - b_b)^2}}{\sqrt{3}} \tag{23}$$

$$\omega_{euklid}(a,b) = \frac{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} - d_{min}}{d_{max} - d_{min}} \tag{24}$$

$$\omega_{normals}(a,b) = 1 - n_a \cdot n_b \tag{25}$$

3.1.2 MST-3D

3.1.2.1 Neighbourhood

Based on the general considerations on neighbourhood systems outlined previously and the evaluation in [Ben-Shabat et al., 2018] the k-nearest neighbours within a maximum radius for the construction of the neighbourhood connectivity graph are used. The maximum radius is selected according to the average point density of the respective point cloud.

3.1.2.2 Merge Criterion

Additionally to the original 'size-adaptive' merge criterion as suggested by [F. Felzenszwalb and P. Huttenlocher, 2004] a fixed scale independent from the segment size is used.

If the edge weight $\omega(a,b)$ is below a user defined scale, the segments are merged. This fixed merging criteria, primarily used for comparison purposes, turns the MST-3D approach into a 'classic' region growing approach as used in [Rabbani et al., 2006] or [Poechtrager, 2016].

3.1.2.3 Weights

Additional to the Euclidean distance in the RGB and LAB space and angle between the normal vectors a new weight based on the normal vectors is introduced. The angle between two normal vectors generally is defined as the scalar product

$$\vec{n_1} \cdot \vec{n_2} = ||\vec{n_1}|| \cdot ||\vec{n_2}|| \cdot cos(\theta) \tag{26}$$

If n_1 and n_2 are similar, θ will be small. Sorting the edges in the connectivity graph in ascending order based on θ will guarantee, that points with similar normal vectors will be merged first.

Figure 3.4 shall illustrate an example where a weight based on the normal vectors might be a problematic choice. A 3D point cloud including a parked car (blue) and the surrounding street (gray) is shown. P corresponds to a point on the ground, Q and T are points on the car. v_1 and v_2 are the vectors connecting PQ and QT. As the roof of the car is approximately horizontal, the normal vectors of $\vec{n_1}, \vec{n_2}, \vec{n_3}$ will be similar. As the edges of the connectivity graph are sorted in ascending order, it might happen that P and Q are merged into the same segment.

While their normal vectors are similar, they actually correspond to completely different surfaces. It is not possible, based on the scalar product, to distinguish such a case. Therefore a measure is needed, which not only captures the angle between normal vectors, but also the relative position of the points themselves with respect to the normal vectors. We refer to such a measure as orthogonal distance.

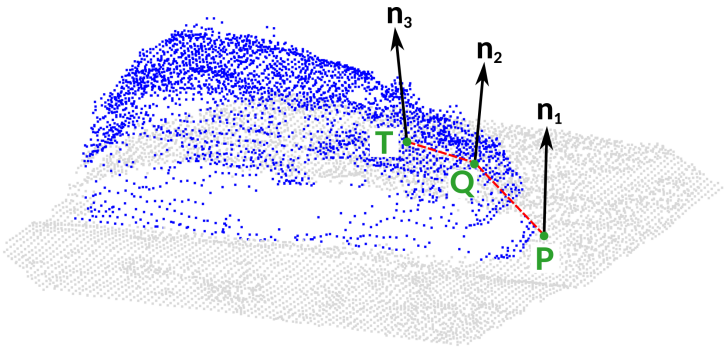

Figure 3.4: Example showing the problem when using the scalar product as weight between two normal vectors

In figure 3.4 $\vec{n_1}, \vec{n_2}$ represent the normal vectors for the points P and Q. $\vec{v_1}$ is the vector $\vec{PQ}$ and $\vec{v_2}$ the vector $\vec{QT}$. The orthogonal distance is defined as

$$dist_{ortho} = \|\vec{v_1}\| \cdot \cos(\theta) = \vec{v_1} \cdot \frac{\vec{n_1}}{\|\vec{n_1}\|} \tag{27}$$

With $\|\vec{n_i}\| = 1$ (27) simplifies to

$$dist_{ortho} = \vec{v_1} \cdot \vec{n_1} \tag{28}$$

(28) projects $\vec{v_1}$ onto $\vec{n_1}$. In general $\vec{n_1}, \vec{n_2}$ are not parallel (Figure 3.4) causing

$$\vec{v} \cdot \vec{n_1} \neq -\vec{v} \cdot \vec{n_2} \tag{29}$$

To account for this asymmetry the orthogonal distance is defined as

$$dist_{ortho} = \max\left\{|\vec{v} \cdot \vec{n_1}|, |-\vec{v} \cdot \vec{n_2}|\right\} \tag{30}$$

3.2 SLIC

3.2.1 Original implementation

SLIC as proposed in [Achanta et al., 2012] is an adaptation of the k-means clustering approach, producing regularly shaped superpixels in an efficient and simple manner.

Based on regularly positioned seeds (highlighted pixels in Figure 3.5) the distance in a selected feature space is calculated to every other pixel within a defined search region. While the search region can be defined arbitrarily, [Achanta et al., 2012] used two times the seed resolution (2S), resulting in an high overlap of the search regions. Using two times the seed resolution guarantees that each pixel can be observed from 4 seeds (except for those pixels at the image border).

For each pixel the closest seed based on the selected feature distance is obtained. After all pixels have been assigned to the respective closest seed, the seeds are updated. This steps are iteratively repeated until the change of seed points in two consecutive iterations is smaller than a certain threshold or a maximum number of iterations is reached. According to [Achanta et al., 2012] the algorithm normally converges within 10 iterations.

In the original implementation a weighted combination of spatial and colour proximity is used as distance measure in order to assign pixels to their closest seed:

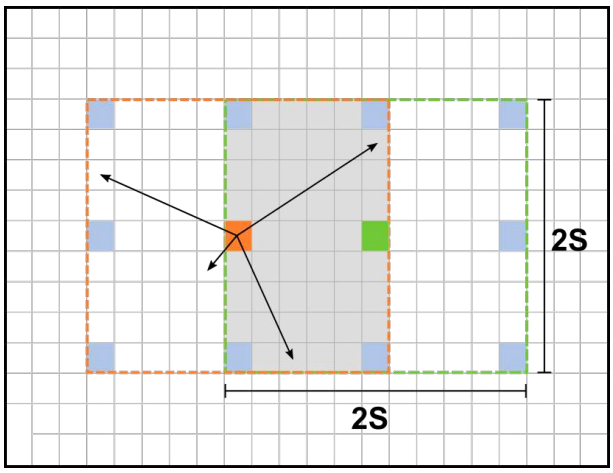

Figure 3.5: Initial slic seeds; search regions for two selected seeds are highlighted; overlapping region is highlighted

$$d_{spatial} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{31}$$

$$d_{colour} = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \tag{32}$$

Both $d_{spatial}$ and d_{colour} are combined into

$$D = \sqrt{\left(\frac{d_{spatial}}{N_s}\right)^2 + \left(\frac{d_{colour}}{N_c}\right)^2} \tag{33}$$

using two normalization factors N_s and N_c. While N_s is defined by the size of the search region, N_c can not be defined in such a way. Therefore N_c is replaced with an user defined constant m referred to as the compactness factor. Using m the final formulation of the distance measure used for the k-means clustering is obtained:

$$D = \sqrt{\left(\frac{d_{spatial}}{N_s}\right)^2 + \left(\frac{d_{colour}}{m}\right)^2} \tag{34}$$

A high compactness factor m results in compact clusters giving more weight to spatial proximity (Figure 3.6 b) at the cost of boundary adherence. On the contrary choosing a small m the resulting segments will better adhere to boundaries loosing their regular shape (Figure 3.6 c,d).

In contrast to the graph-based segmentation introduced in the previous section the number of segments can directly be defined based on the seed resolution. Due to the regular positioned seeds in combination with a medium compactness factor the resulting segments have similar compact shapes and their size varies to a lesser extent compared with the graph-based method.

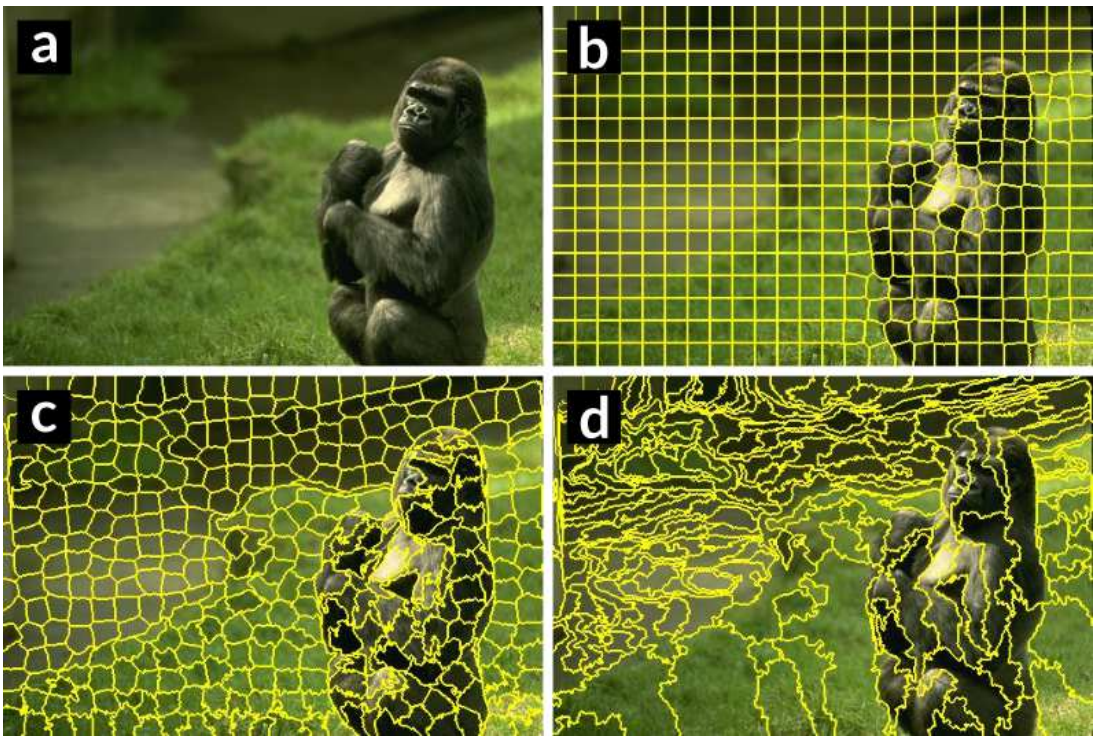

Figure 3.6: Original image (a) and the results of the SLIC segmentation for varying compactness factors (b - d)

Algorithm 2: SLIC-3D segmentation

Result: Segmented point cloud

`// Initial seeds`
Obtain initial seeds based on voxel grid and calculate initial attributes

`// actual segmentation`
$iter = 1$
$d_i = \infty$
$l_i = -1$

while $iter \leq max$ **do**
  for *each seed point* S_k **do**
    for *each point* P_i *in search region around* S_i **do**
      compute distance D between S_i and P_i
      if $D \leq d_i$ **then**
        $d_i = D$ `// set min distance to the current distance`
        $l_i = l_k$ `// set label of point to the label of seed`
      end
    end
  end
  update seed points
  $iter+ = 1$
end

3.2.1.1 Related work

While there exist some super-voxel methods ([Lin et al., 2018], [Gao et al., 2017]) only one method can be interpreted as a direct extension of SLIC to (unstructured) 3D point clouds. The implementation reffered to as 'Voxel Cloud Connectivity Segmentation' (VCCS) from [Papon et al., 2013] is available in the open source software PCL [Rusu and Cousins, 2011].

This approach does not directly segment the original 3D point cloud itself, but the voxelization of it. As already mentioned previously using voxels allows the definition of neighbourhood similar to images and reduces computation time as the amount of points is reduced.

A 39 dimensional feature space based on spatial coordinates, LAB colour and local geometric features (FPFH) [Rusu et al., 2009] are combined into the distance measure D:

$$D = \sqrt{\frac{\lambda \cdot D_c^2}{m^2} + \frac{\mu \cdot D_s^2}{3 \cdot R_{seed}^2} + \epsilon \cdot D_f^2} \tag{35}$$

In contrast to the original SLIC approach, where the segmentation is conducted in an iterative manner, VCCS is based on a breadth-first search of the adjacency graph. Due to this spatial connectivity is enforced already during segmentation without the need of a post processing step.

While using a voxelized point cloud instead of the original point cloud increases efficiency due to the simplification (all points within a voxel cell are represented by one representative point) information might be lost reducing the quality of the segmentation result.

3.2.2 SLIC-3D

Compared to the VCCS the implementation developed within this work directly operates on the original 3D point cloud.

3.2.2.1 Seed points

SLIC with its fixed search radius requires regular 'gridded' seed points. While this regular seed points can precisely be defined for the 2D image case, for 3D point clouds this regularity is not clearly defined and ambiguous.

Within the implementation developed throughout this thesis a voxel grid of user-defined resolution is superimposed onto the 3D point cloud. After elimination of all voxels containing no points, the initial seeds are obtained based on the points contained within each voxel.

As in the original implementation, in the beginning just the center coordinates of the voxels were used as initial seed positions. This led to problems in some rare cases, where no points were found within the search radius of a seed point in the subsequent segmentation. To overcome this problem, the initial location of each seed point S_i is obtained by calculating the mean coordinates of all points located within the voxel V_i.

To obtain the initial colour attributes of the seed points, as for the coordinates, the mean value of the respective colour components from all points located within the current voxel are used.

In the case of the normal vectors a different method is used. As outlined earlier, the method used for the estimation of the normal vectors also provides a quality parameter s_0, describing the standard deviation of the fitted plane. Therefore the normal vector with the smallest s_0 is selected.

3.2.2.2 Search Region

Instead of using a rectangular region around each seed point as for the 2D case, a spherical neighbourhood is used. The radius of the search sphere equals the resolution of the voxel grid.

3.2.2.3 Distance measure

As for the MST-3D different combinations of the distance measure D are evaluated and compared against in terms of under-segmentation. The individual feature distances are

Spatial Distance: Expressed as the Euclidean distance of the coordinates of the seed point and the points within the search radius:

$$D_{spatial} = \sqrt{(x_S - x_P)^2 + (y_S - y_P)^2 + (z_S - z_P)^2} \tag{36}$$

Colour Distance: Depending on the colour system either the Euclidean metric in the LAB or RGB space is calculated:

$$D_{lab} = \sqrt{(l_S - l_P)^2 + (a_S - a_P)^2 + (b_S - b_P)^2} \tag{37}$$

$$D_{rgb} = \sqrt{(r_S - r_P)^2 + (g_S - g_P)^2 + (b_S - b_P)^2} \tag{38}$$

Geometrical Distance: Either the angle between normal vectors or orthogonal distance are used with v being the vector between the seed and respective point:

$$D_{normal} = n_S \cdot n_P \tag{39}$$

$$D_{ortho} = \max\left\{|\vec{v} \cdot \vec{n_P}|, |\vec{-v} \cdot \vec{n_S}|\right\} \tag{40}$$

Each feature distance is divided by an user defined compactness factor in order to control the weighting of the individual features.

3.2.2.4 Post-processing

Due to the overlapping search regions and iterative procedure no spatial connectivity is enforced during the segmentation leading to small isolated segments. In order to identify isolated segments the region adjacency graph (RAG) based on the neighbourhood connectivity graph is used. By identifying all connected components within the graph after removing the edges connecting two nodes with different segments all individual segments are obtained. Afterwards all segments smaller than a selected minimum segment size will be merged with the most similar connected segment similar to the post-processing step conducted in the MST-3D.

4. Reference data

For the selection of the reference datasets different aspects have been considered:

1. Point clouds should represent different scenes including objects of different types, shape and size.
2. Each dataset must contain human labelled ground truth classes which can be used to verify the quality of the segmentation methods.
3. Different acquisition methods should have been used resulting in different point densities in order to show the generalizability and robustness of the segmentation methods.
4. At least one dataset should contain colour information.

Based on these criteria the following datasets have been used:

	Pix4D	**MA41**		**ParisTech**
	Ankeny	**AOI 10**	**AOI 12**	**Lille**
Method	SFM	ALS	ALS	MLS
Scene	rural	park	urban	urban
Points (total)	9 000 000	800 000	250 000	8 000 000
Points per m^2	160	90	23	2000
		Classes (%)		
	Ankeny	**AOI 10**	**AOI 12**	**Lille**
Ground	-	84.0	27.7	48.3
Ground (concrete)	8.3	-	-	-
Ground (grass)	68.6	-	-	-
Vegetation	11.3	10.6	0.1	2.1
Building	9.7	-	-	44.0
Building (roof)	-	-	51.8	-
Building (wall)	-	-	11.5	-
Cars	0.8	5.3[1]	8.9[1]	4.4
Objects	1.2	-	-	2.1

Table 1: Reference data statistics

4.1 Pix4D

The Ankeny dataset, published by Pix4D[2] , includes colour information and manually assigned ground truth labels. Based on aerial images acquired using an UAV the respective dense point cloud was obtained. Unfortunately no information about the used camera and other information regarding the data acquisition are available.

Showing an rural scene, the point cloud includes vegetation, buildings, cars and other smaller man made structures. The ground is separated into an artificial (street) and natural (grassland) class. For buildings no distinction between the walls and roofs have been made.

[1]For the MA41 datasets objects like traffic signs, poles etc. are not separated from cars.
[2]`https://www.pix4d.com/research`

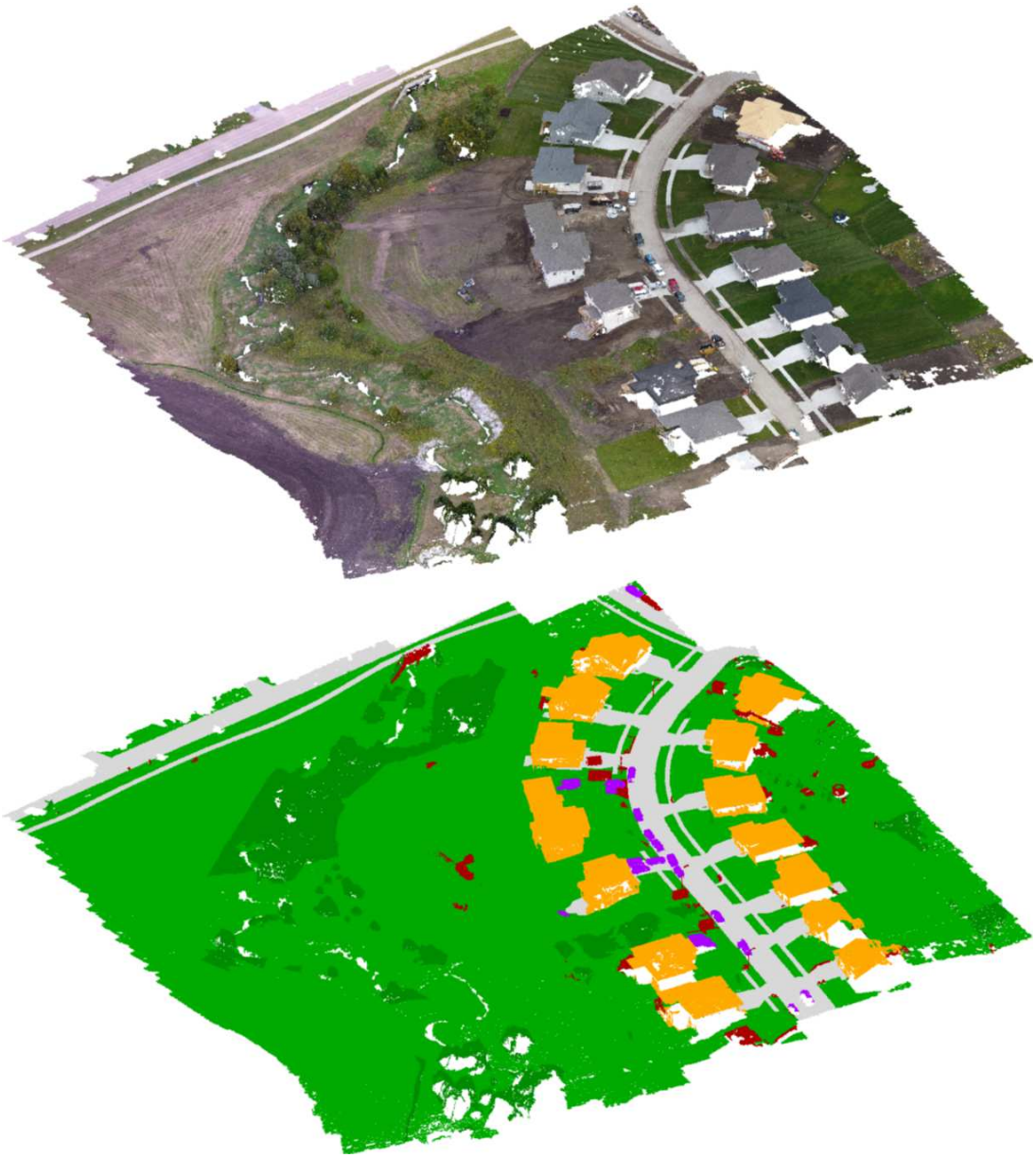

Figure 4.1: Ankeny top view with true colour (top) and ground truth classes (bottom)

4.2 MA41

This dataset, provided by the city of Vienna, includes data from an ALS mission carried out in 2006/2007. Using a RIEGL LMS-Q560 fullwaveform laserscanner and an average flight height of approximately 500m results in a mean point density of 15 - 20 $\frac{pts}{m^2}$ [Rutzinger et al., 2008].

Two tiles representing two different urban scenes have been selected from this dataset:

AOI 10: This tile covers the Heldenplatz, a public square in the centre of Vienna (Figure 4.2 top). Next to the park with individual trees runs a street with cars parked along its entire length. No buildings are present within this tile.

AOI 12: AOI 12 (Figure 4.2 bottom) is a typical urban scenery mainly dominated by high buildings and cars parked along the street. Only one tree in the top left corner is included. Along the vertical structures of the buildings the point density is significantly lower compared to the horizontal surfaces / structures.

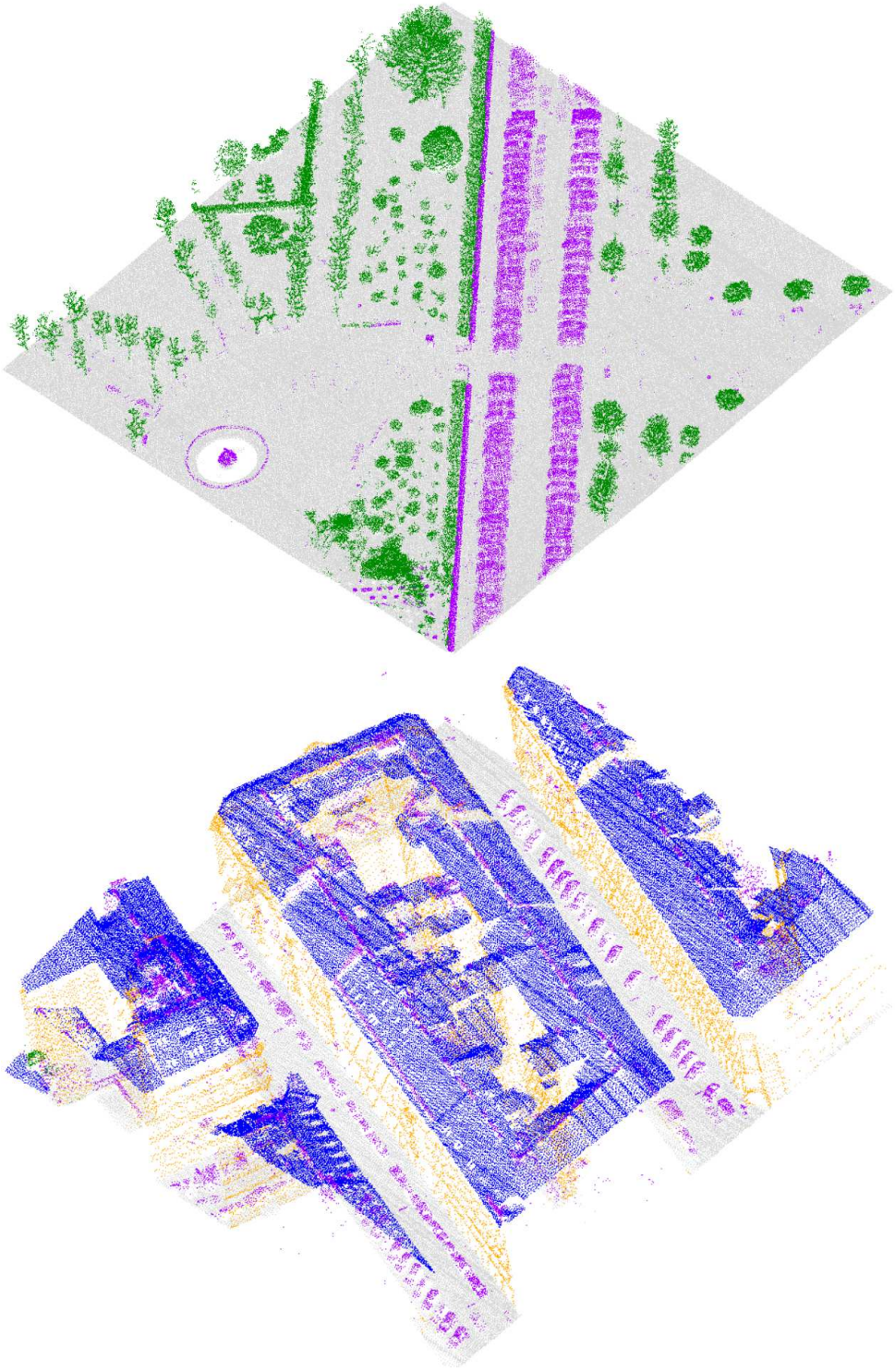

Figure 4.2: Tiles used from the MA41 dataset: AOI 10 (top) and AOI 12 (bottom). Points are coloured by ground truth class.

4.3 Lille

The point cloud has been acquired using a mobile mapping system developed at the center for robotics of Mines ParisTech[1] equipped with a Velodyne HDL32E LiDAR laser scanner. Compared to all other datasets this dataset has the highest point density.

It shows an urban scene in Lille (France). While the complete dataset is 2 km long and contains roughly 150 millions points, a 150m long section (Figure 4.3 orange bounding box) with approximately 8 million points was used.

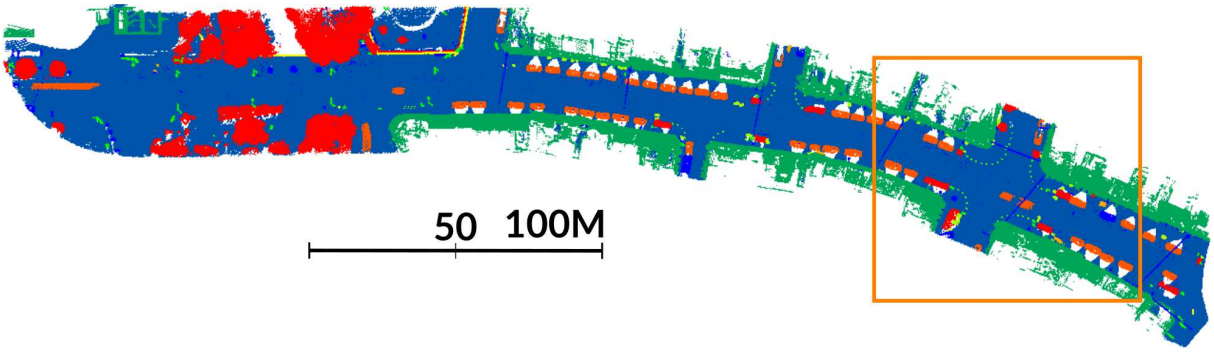

Figure 4.3: Lille dataset with the selected area of interest (orange bounding box)

The section was chosen to include all relevant object classes namely street, cars, building, vegetation and pole like objects.

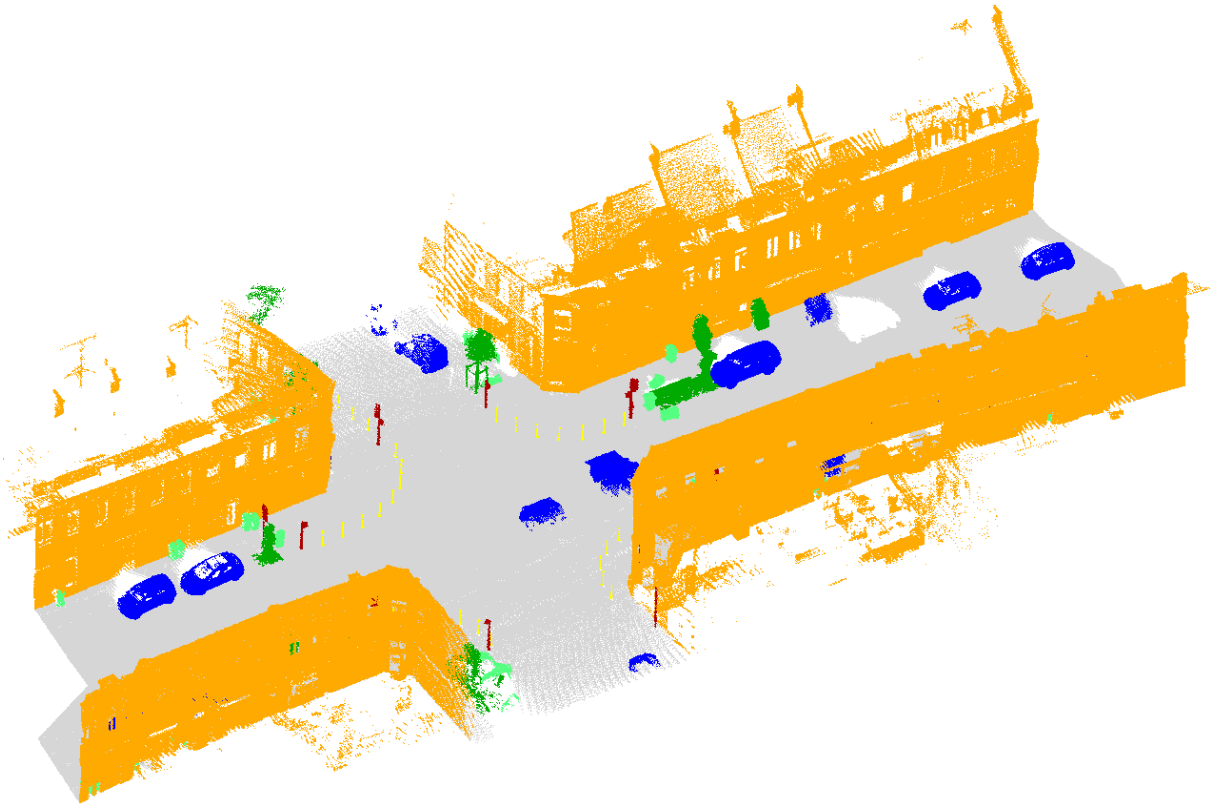

Figure 4.4: Detailed view of the selected area of the Lille dataset. Points are coloured by their class.

[1] http://npm3d.fr/paris-lille-3d

5. Experiments

All tests were run on a notebook with 16GB of RAM and a Intel Core i5-7200U CPU with 2.50GHz.

5.1 MST-3D

5.1.1 Examination for correctness

To examine the correctness of the implementation different tests have been conducted.

First the segmentation results for images are compared with the results obtained using the original implementation by [F. Felzenszwalb and P. Huttenlocher, 2004]. To use exactly the same routines and data structures as for 3D point clouds, the row and column indices of the pixels are interpreted as coordinates in meters. The connectivity graph is build using a kd-tree with 9 nearest neighbours (as each point to itself is its closest neighbour and will be removed afterwards) and a maximum search radius of 1.5 (the maximum distance between two pixels is $\sqrt{1^2 + 1^2} \approx 1.42$) leading to a 8-connected neighbourhood.

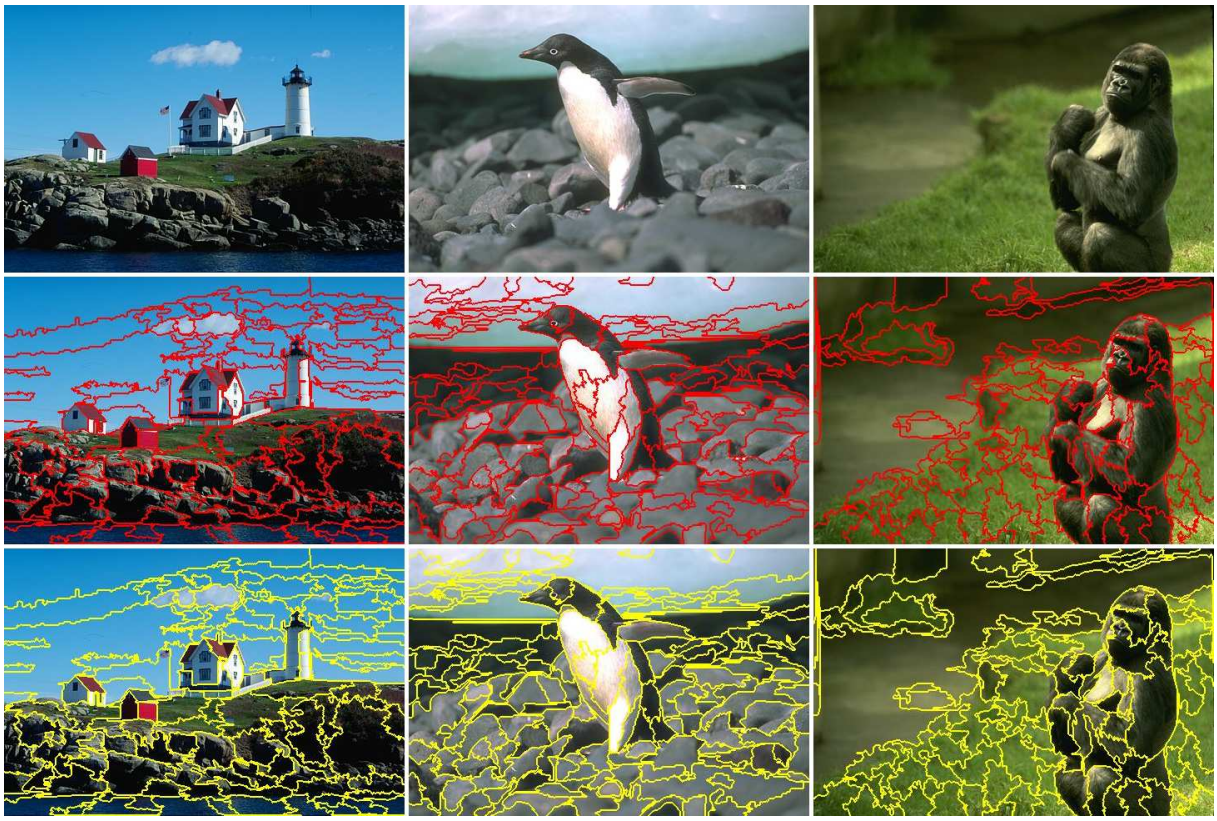

Figure 5.1: Segmentation results for the original implementation (middle row) and the MST-3D implementation (bottom row)

Various images were segmented with different scales τ in both colour spaces (RGB, LAB) and equality of the segment borders checked. All tests proofed that both implementations return exactly the same segmentation results.

To further verify the segmentation of 3D point clouds the MST-3D segmentation (using a fixed scale, no post processing step and angle between normal vectors as weight) was compared with the results from opalsSegmentation[1]. The results were evaluated point by point

proofing that they return the same results. For visual comparison both results are shown in Figure 5.2.

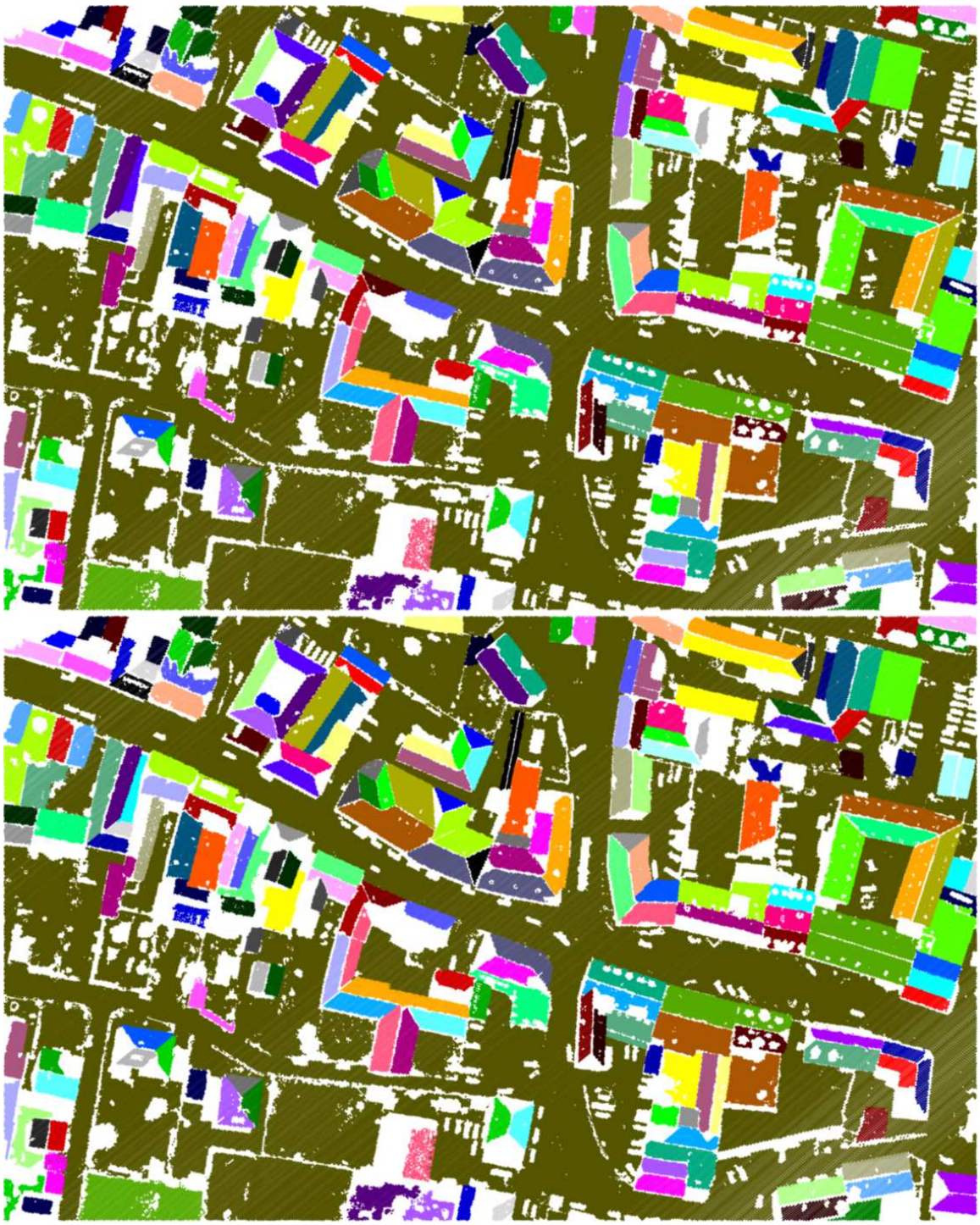

Figure 5.2: Segmentation results using opalsSegmentation (top) and MST-3D segmentation with fixed scale (bottom); colours represent different segments

[1]https://opals.geo.tuwien.ac.at/html/stable/ModuleSegmentation.html

5.1.2 Connectivity

The segmentation has been conducted for each dataset with various numbers (4, 8, 16, 32) of neighbours for the construction of the neighbourhood connectivity graph. The search radius of 1.5m was chosen to make sure, that it is not the limiting factor for the selection of the number of points. The completeness metric, as defined in section 2 of this thesis, is based on the connectivity graph, it is not suitable for comparison in this case. Instead the mean segment size is used.

The results for the Ankeny dataset are shown in Figure 5.3. For all attributes an increase in the nearest neighbours leads to a smaller under segmentation with respect to the mean segment size. The difference generally increases with increasing scale. The influence of the number of neighbours for smaller scales (small segment sizes) is only minor.

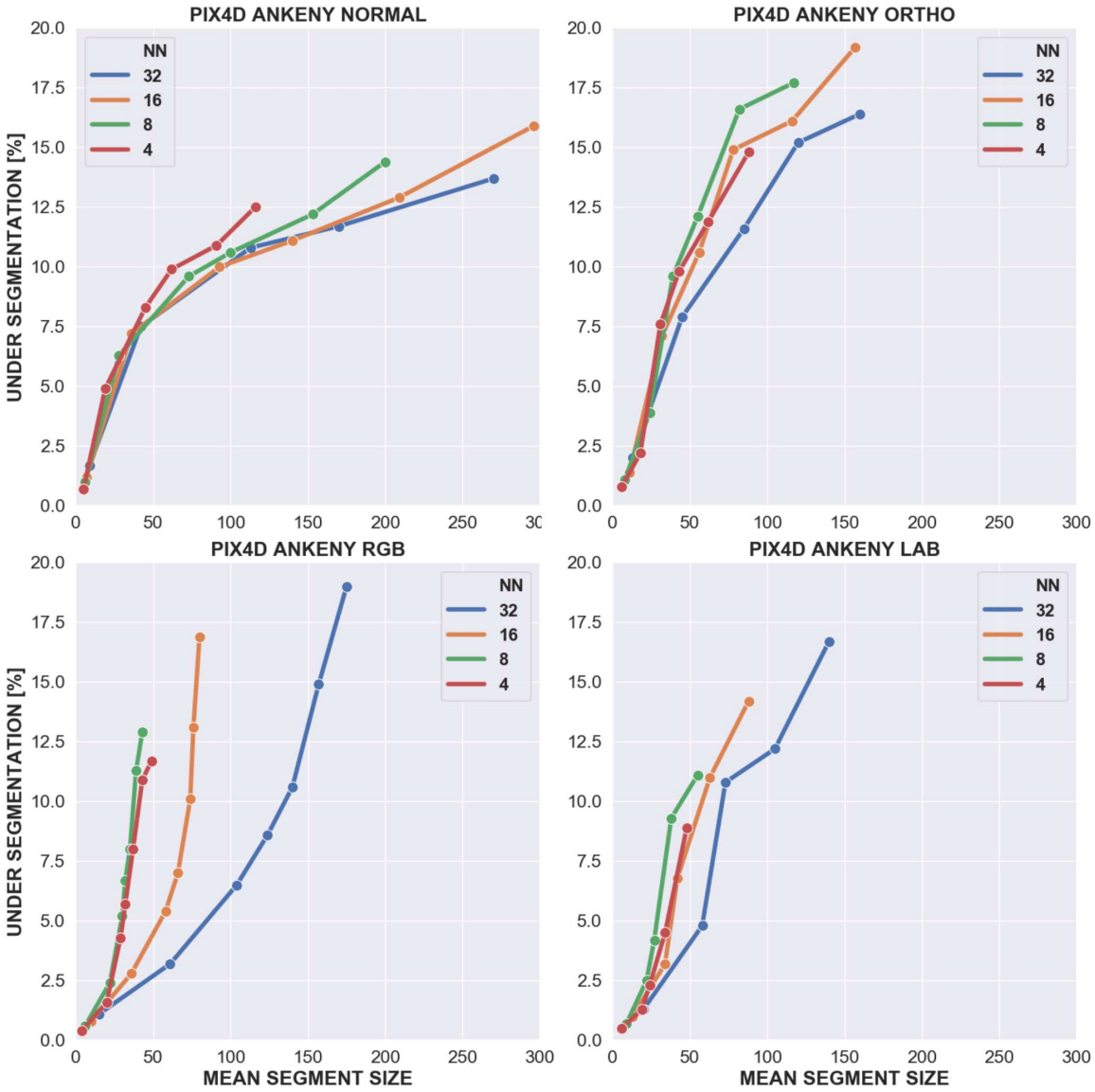

Figure 5.3: Influence of number of nearest neighbours for the Ankeny dataset.

While the results are only presented for the Ankeny dataset, the same tests have been conducted for the Lille dataset where the same tendency can be observed.

On the contrary for the MA41 datasets an increase in the number of neighbours partly leads

to an increased under segmentation error (Figure 5.4). As the point density of these datasets is lower compared to the Ankeny and Lille data, more distant points are considered as neighbours which might already belong to a different object/class resulting in an increased under-segmentation error.

The same behaviour might be observable for the Ankeny and Lille datasets if one would further increase the number of nearest neighbours. Unfortunately, due to performance reason (more than 16GB of RAM would be necessary) this is not possible with the current implementation of the MST-3D.

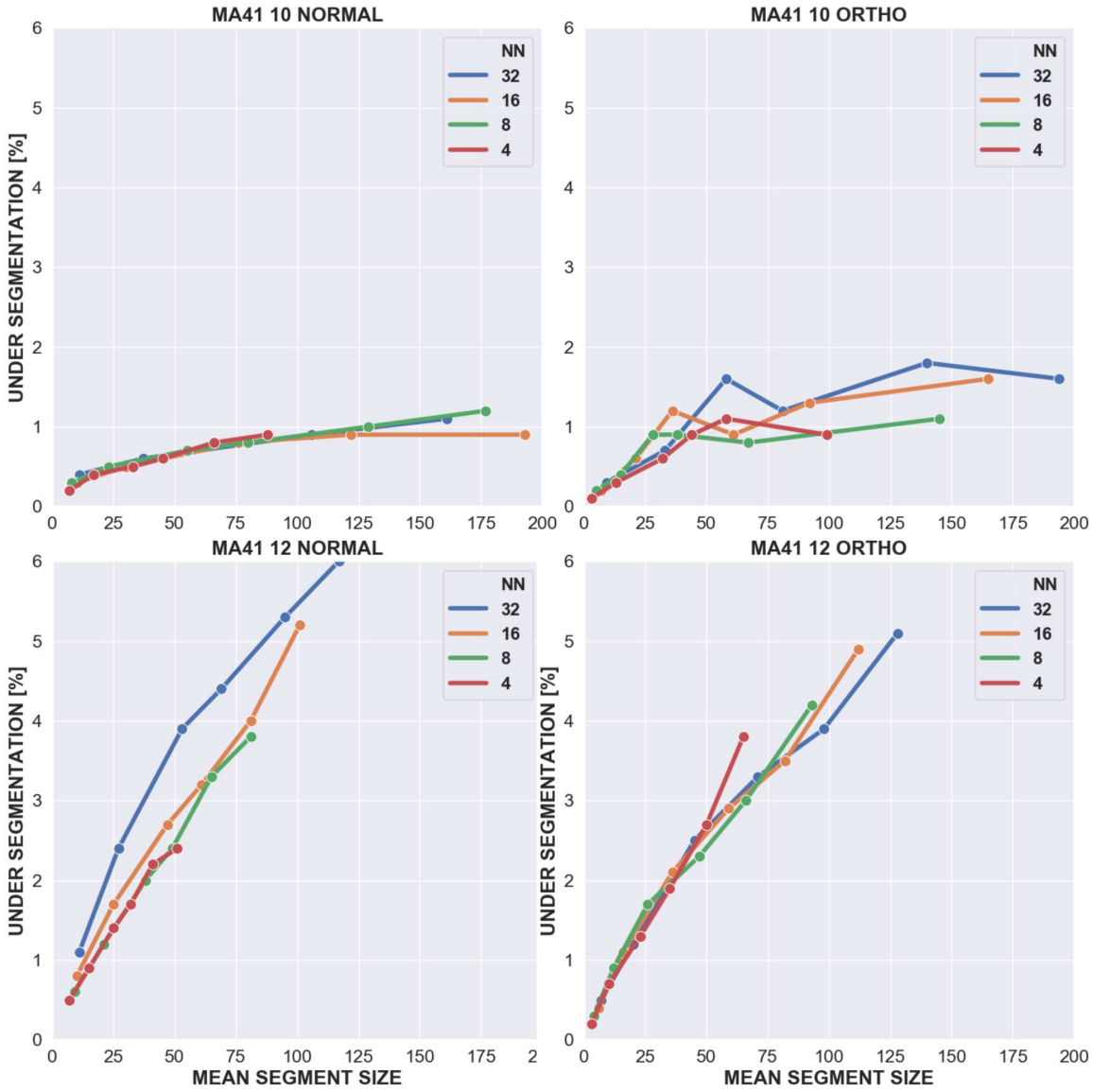

Figure 5.4: Influence of number of nearest neighbours for both MA41 datasets.

5.1.3 Attributes

After evaluating the influence of the number of nearest neighbours onto the segmentation result, within this section the choice of attributes shall be analysed in some detail. For both colour and geometrical attributes two different weights have been implemented:

Geometrical: Angle between normal vectors and orthogonal distance

Colour: Euclidean distance in the RGB and LAB colour space

Besides comparing the different attributes, the size-adaptive criterion as defined by [F. Felzenszwalb and P. Huttenlocher, 2004] is compared with its fixed scale equivalent. For none of the tests a minimum segment size is enforced in order to compare the 'raw' segments without any influence of this optional post-processing step.

For all figures within this section, dashed lines correspond to the size-adaptive criterion labelled in the plots with 'FELS'. The solid lines correspond to the results obtained using the fixed scale, labelled 'FIX'.

Aoi-10

For smaller scales the size-adaptive method results in many small segments (Figure 5.6 a) whereas for the fixed scale ground is already well segmented (Figure 5.6 c). This is well reflected by the overall completeness in Figure 5.5 (left).

Regarding the choice of attribute none is clearly superior in terms of completeness as in all cases almost identical results are obtained. In the case of the fixed approach the number of segments is lower across all scales using the angle between normal vectors as weight (Figure 5.5 - right).

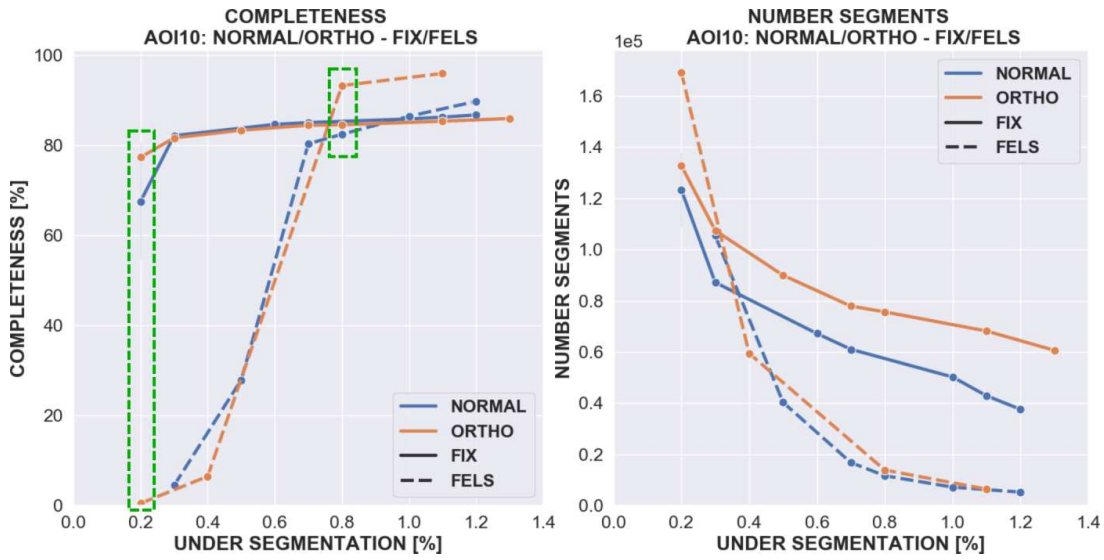

Figure 5.5: Completeness and number of segments for the Aoi-10 tile.

An increase of the scale has a different effect on each approach. While for the fixed method completeness remains more or less constant, for the size-adaptive method it increases from around 0% to more than 95%.

This can also be observed visually in Figure 5.6. The segmentation results for the fixed approach (Figure 5.6 c, d) using a lower (c) and higher (d) scale are more or less the same. Figure 5.6 (a) and (b) correspond to the segmentation using the size-adaptive method. While for the lower scale (a) segments are small and no structures are recognizable, increasing the scale both the ground but especially also vegetation and cars are well segmented (Figure 5.6 b).

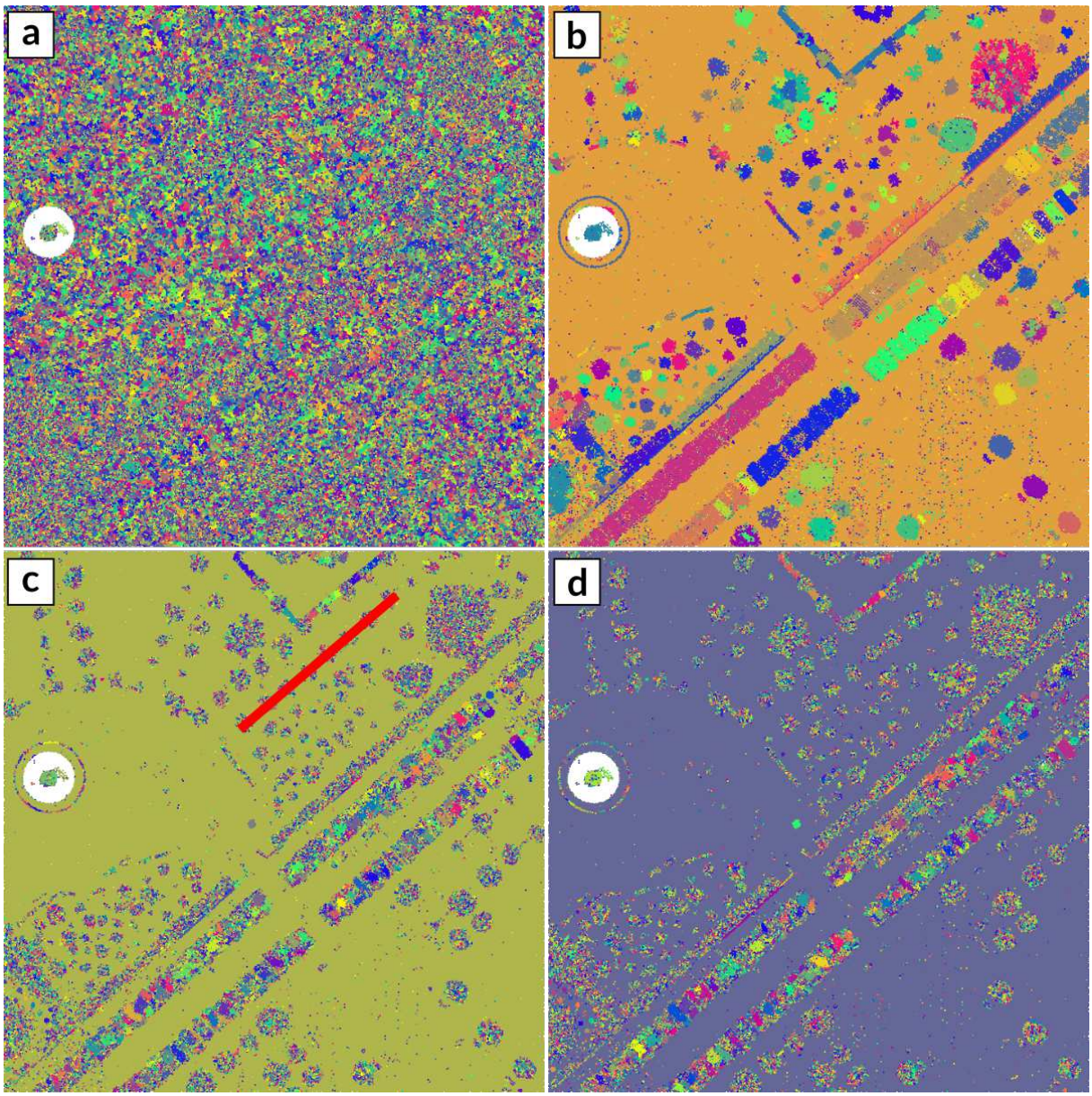

Figure 5.6: Segmentation result for fixed (c, d) and size-adaptive (a, b) method. Red line in (c) indicates the location of the profile shown in Figure 5.8.

A closer look at the completeness for each class (Figure 5.7) reveals that the major gain of completeness is achieved especially for vegetation, cars and objects. This is not surprising, as for these classes the normal vectors are quite inhomogeneous. In order to achieve similar completeness using the fixed approach it would be necessary to increase the scale significantly, leading to large under-segmentation across all classes. This demonstrates the big advantage of the size-adaptive criterion: While for homogeneous regions (ground) the performance is similar to the fixed scale, for inhomogeneous classes the performance is clearly superior.

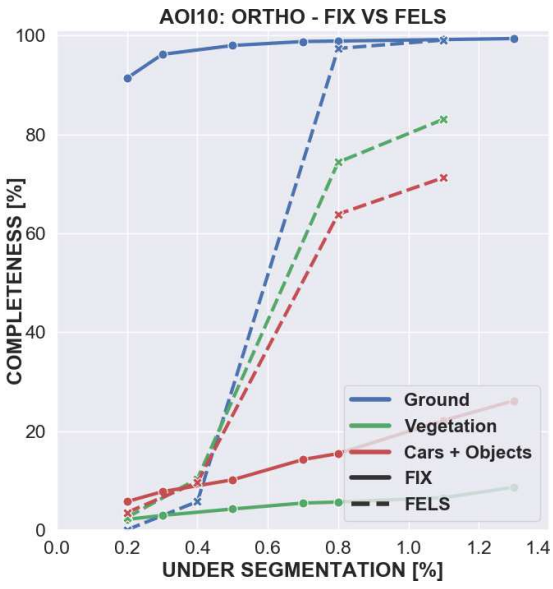

Figure 5.7: Completeness for each class.

A profile section through a series of individual trees (Figure 5.8) further confirms this observation. In both cases the ground is well separated. While for the size-adaptive method larger segments are found within the trees, for the fixed scale individual trees are scattered across numerous segments with sizes below 5 points.

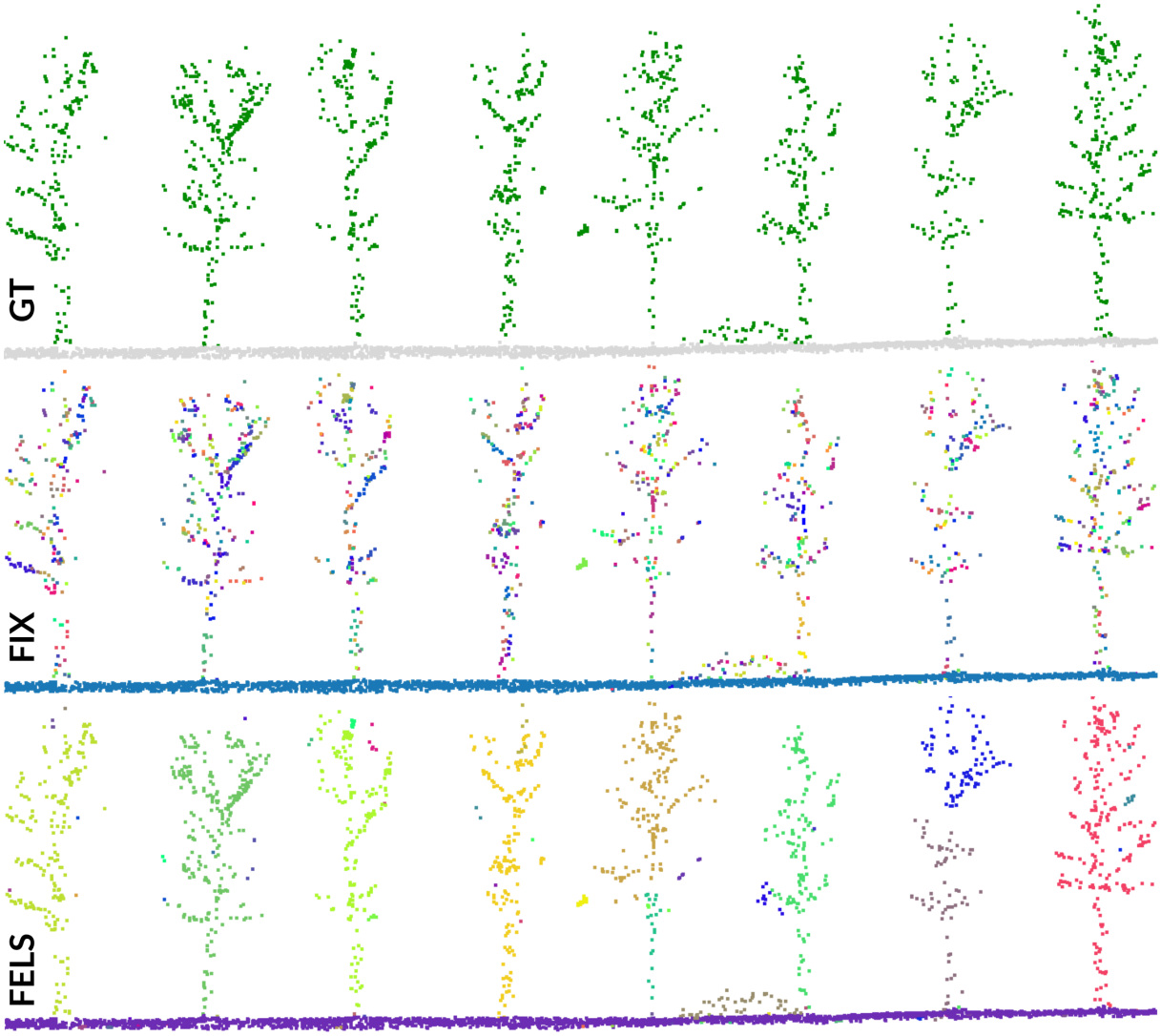

Figure 5.8: Tree profile. Top: ground truth classes; middle: fixed scale; bottom: size-adaptive

In Figure 5.9 (top) the segmentation result based on the size-adaptive method using the orthogonal distance is shown for which the overall completeness of 93% is reached. In the middle all points which define the under-segmentation error for this segmentation are highlighted in red. The bottom figure shows the under-segmentation for the fixed method resulting in the same under-segmentation error. Visualizing the under-segmentation error using the point cloud itself further allows analyzing problematic areas. For visualization purposes the red points are enlarged in size.

Within the areas highlighted by the blue rectangles in Figure 5.9 many points belong either to tree roots or partly to the bottom of cars while they have been added to the ground segment. Visually most striking is the apparent line of red points running diagonally across the tile. In the area enclosed by the green rectangle those points actually belong to some kind of cable running within the vegetation. In the pink region points are added to vegetation while they actually correspond to the ground.

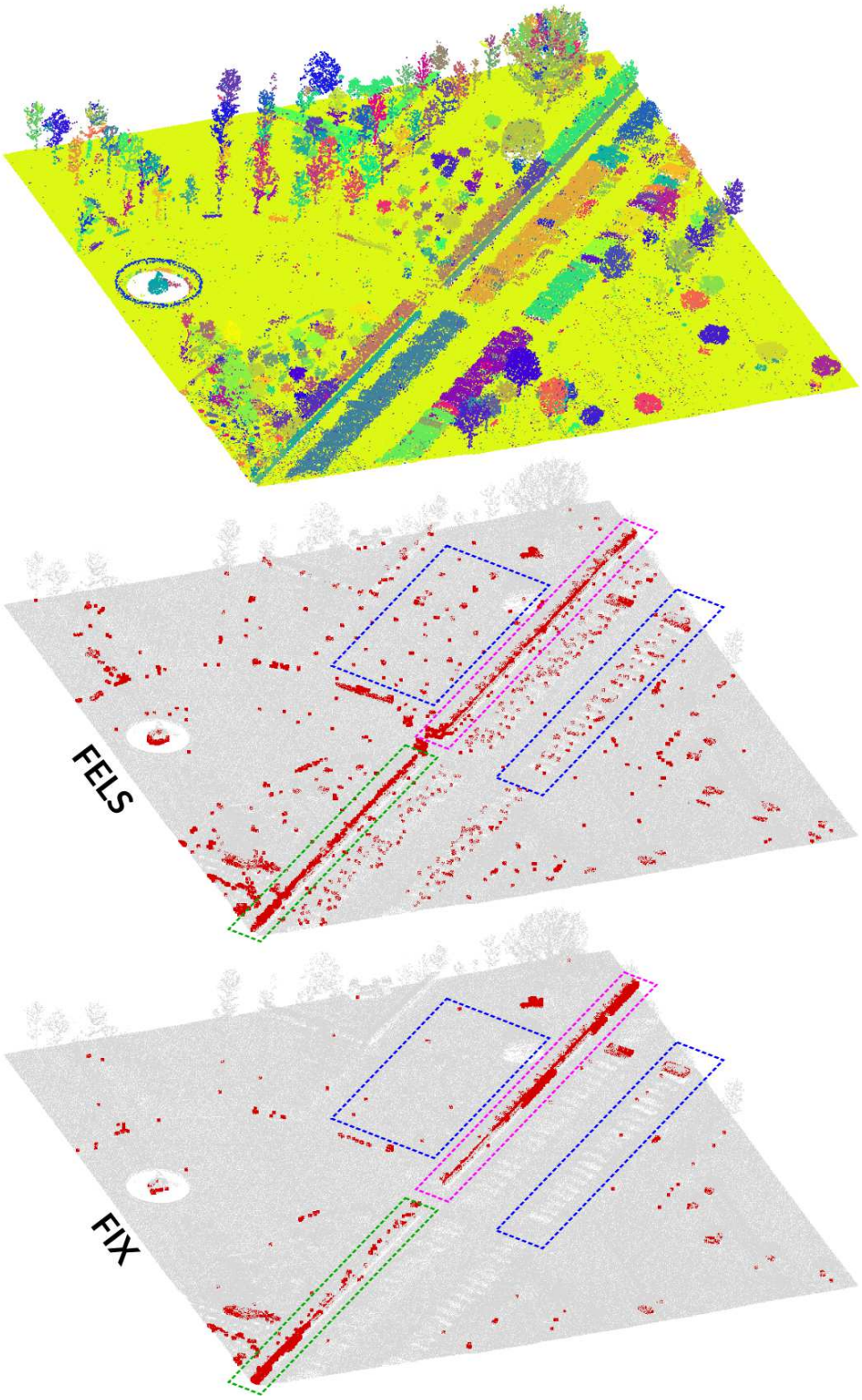

Figure 5.9: Segmentation result (top) and points defining under-segmentation highlighted in red (middle) for the size-adaptive method. The Bottom figure shows under-segmentation for the fixed method.

For the fixed method less points from cars or tree roots are added to the ground segment as one can see from Figure 5.9.

A detailed view of the region highlighted with the pink rectangle is shown in Figure 5.10. A look at the vegetation clearly shows the advantage of the size-adaptive method. Also the small wall below the vegetation (highlighted by the red rectangle) is well segmented. For the fixed method parts of it are merged with the segment representing ground. This is also well reflected by the under-segmentation value.

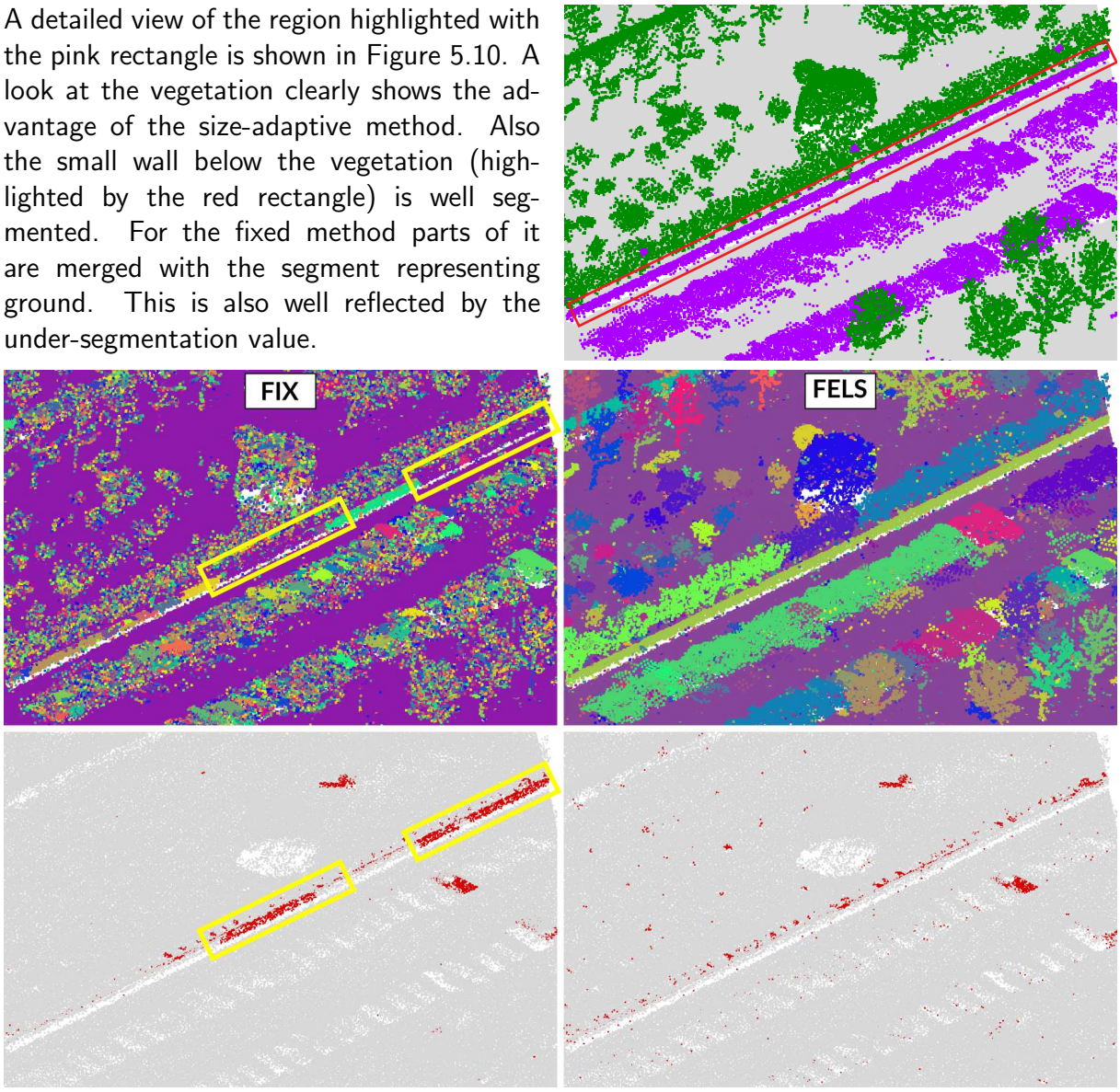

Figure 5.10: Detailed view of the region highlighted by the pink rectangle in Figure 5.9

Aoi-12

The Aoi-12 tile is a more complex urban scene mainly dominated by high buildings. Most of the points correspond to smoother surfaces like roofs, walls or ground. As one could observe for the previous dataset, the size-adaptive method is superior especially for inhomogeneous objects like vegetation. Due to the lack of these (only one tree is included within this scene) the fixed scale approach results in a higher completeness compared to the size-adaptive method across all scales.

Regarding the number of segments the difference between the size-adaptive and fixed scale is only minor. Especially the different weights produce more or less exactly the same amount of segments.

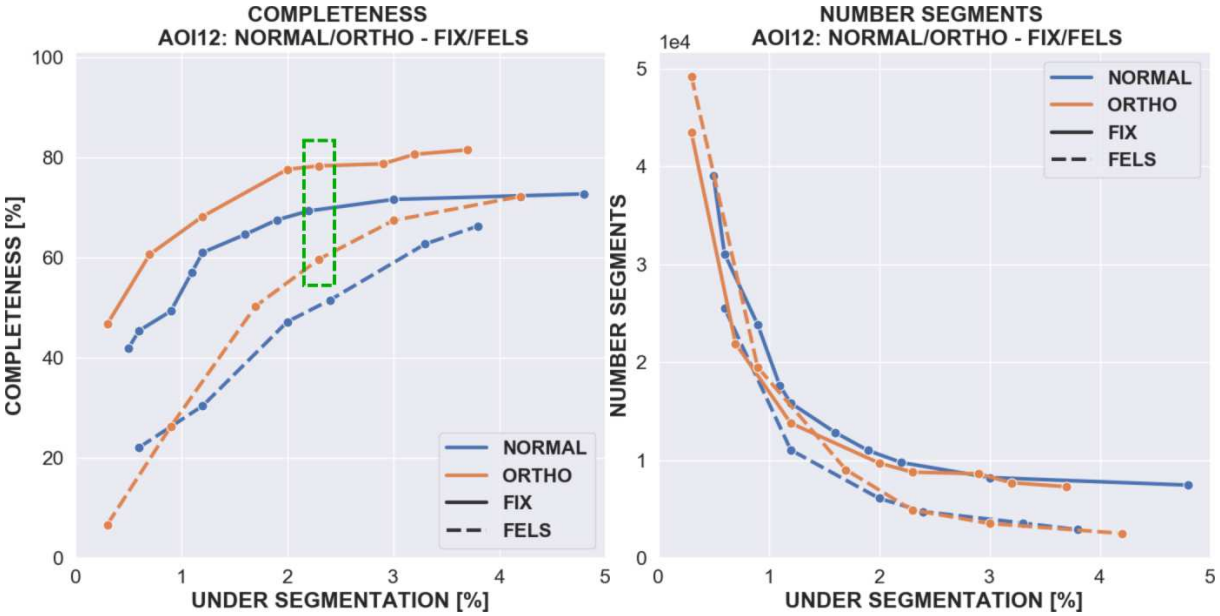

Figure 5.11: Comparison of the fixed and size-adaptive merging criteria for the Aoi-12 dataset. left: overall completeness; Right: number of segments.

By looking again at the completeness for each ground interesting details are revealed.

As for the Aoi-10 tile completeness for the ground class increases most with increasing scale for the size-adaptive method. The biggest difference between size-adaptive and fixed approach occurs for the building (roofs) class. For this class across all scales the fixed method results in an improved completeness rate of approximately 30 - 40%. Vegetation is the only class where the size-adaptive method is superior but as already mentioned the result is not really meaningful as only one tree is included within this scene.

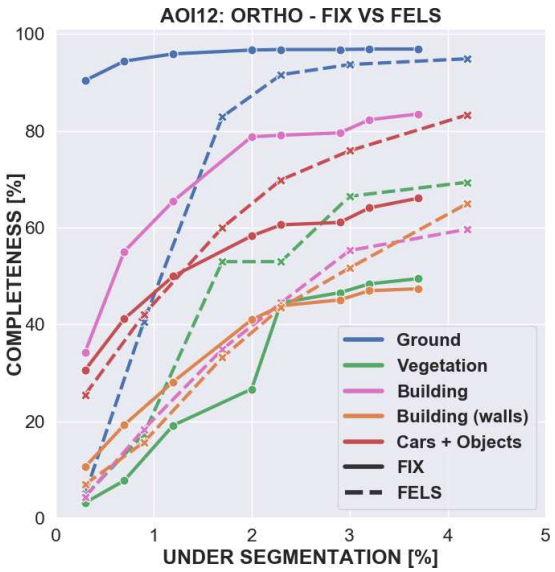

Figure 5.12: Completeness for each class.

In Figure 5.13 the results for the segmentation highlighted with the green dashed rectangle in Figure 5.11 (left) is shown. For the size-adaptive method individual roof parts are individual

segments, for the fixed scale bigger parts of the roof are within one segment. While this is not false, the whole roof is one connected component within the ground truth data. Therefore completeness for the size-adaptive method is smaller. Besides that no major differences can be observed.

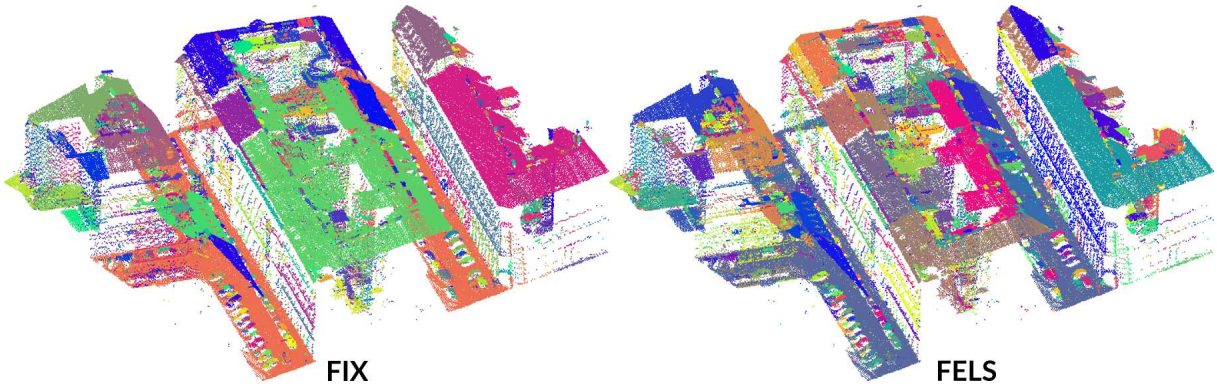

Figure 5.13: Comparison of the fixed and size-adaptive merging criteria for the Aoi-12 dataset.

Lille

In the MA41 dataset segmentations the orthogonal distance was slightly superior or no major difference between the weights was observed. In contrast, for the Lille dataset the angle between normal vectors results in higher completeness both for the fixed and size-adaptive method. (Figure 5.14 left).

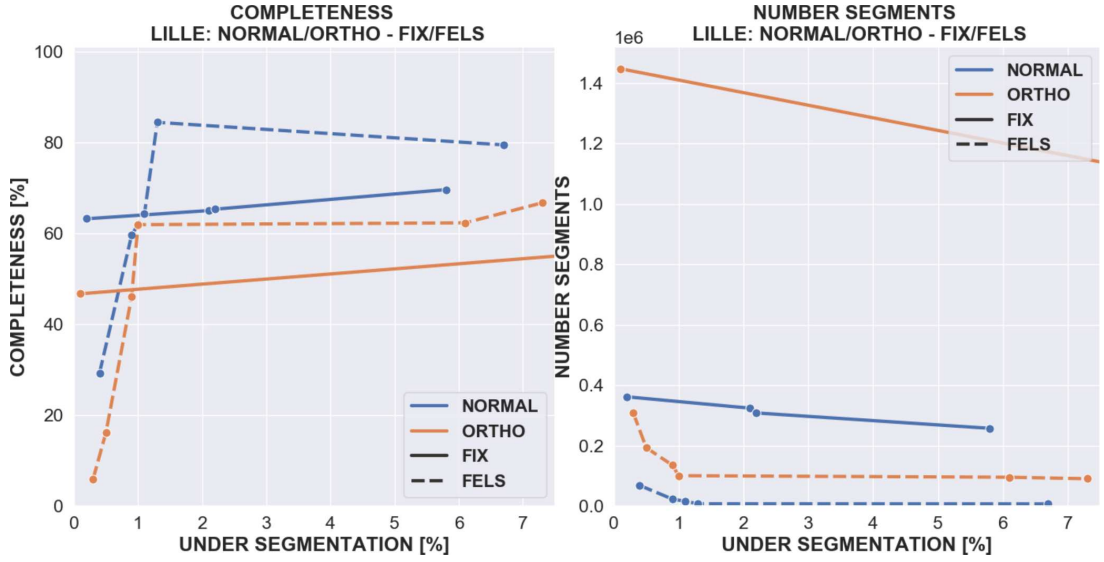

Figure 5.14: Comparison of the fixed and size-adaptive merging criteria for the Lille dataset. Left: overall completeness; Right: number of segments.

The MA41 datasets were acquired using ALS leading to a lower point density compared to this dataset obtained by MLS. Hence less details are captured. Surfaces and objects which appear smooth in ALS, in reality are much rougher and have stronger surface variations. Since the orthogonal distance is a stricter metric compared to the angle between normal vectors, for lower point densities it improves the segmentation results.

Besides the higher point density the scan quality of the MLS itself might be worse compared to ALS. While this would also influence the result, this was not further analysed.

Especially in Figure 5.14 (right) this becomes clear. For the fixed scale using the orthogonal distance results in more than 1.2 million segments whereas for the difference in normal vectors the number of segments is below 400 000.

Regarding completeness the situation is quite similar to the Aoi-10 tile. While for small scales the size-adaptive approach segments the ground class poorly in terms of completeness, the fixed scale already achieves closely 100%.

Increasing scales for the size-adaptive approach, completeness increases significantly while for the fixed method it remains constant. For the size-adaptive method a maximum completeness is reached at approximately 1.2% under-segmentation error. Above this completeness remains constant except for the building class where it drops from 75% to below 60%. This indicates that larger parts of buildings are added to segments representing different classes for example ground.

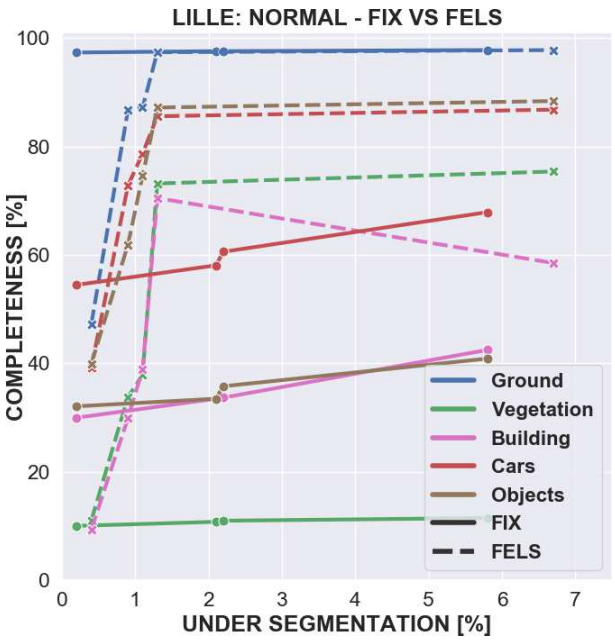

Figure 5.15: Completeness for each class.

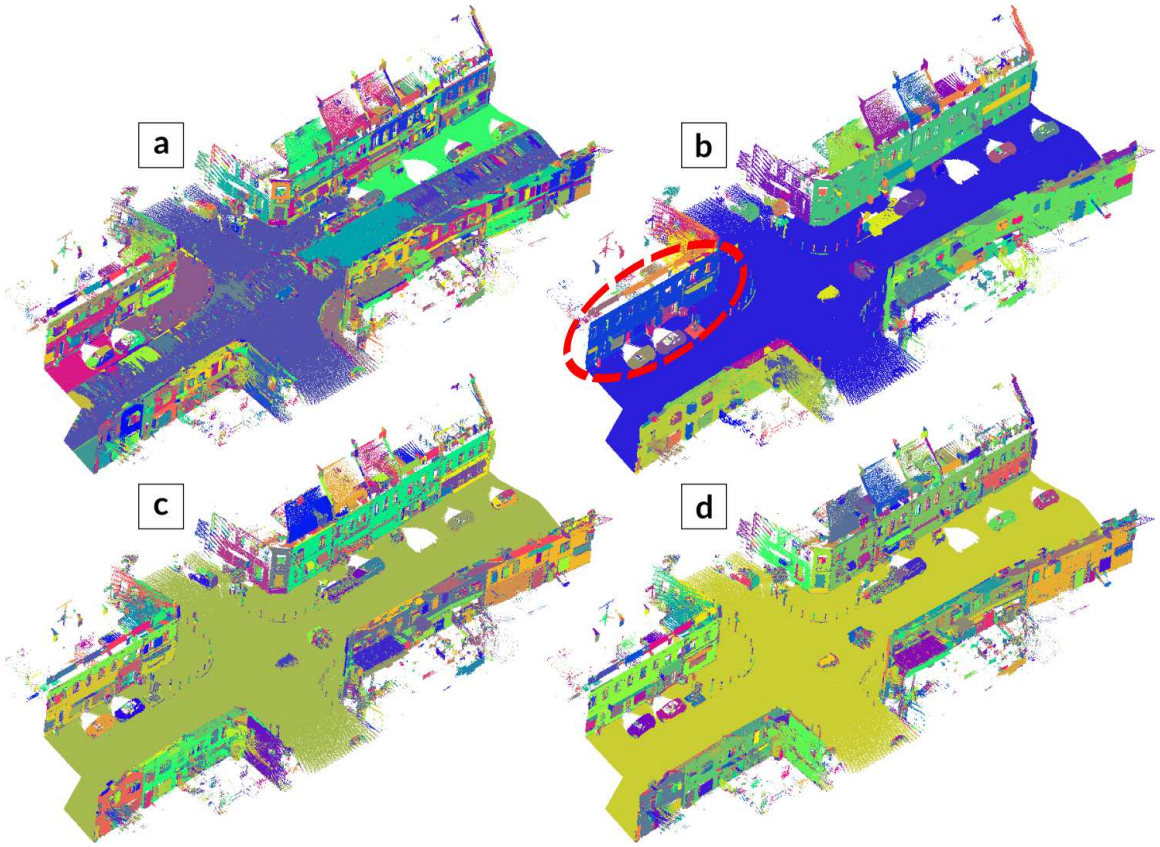

Figure 5.16: Size-adaptive (a, b) and fixed (c, d) method. First column corresponds to low scales, second to large ones. The building part within the red ellipse looks like it belongs to the ground but this is only due to colouring.

As for the Aoi-10 tile, the final result and under-segmentation error for the Lille dataset are shown in Figure 5.17. Obviously none of the side curbs have been properly segmented as most of them have been added to the street segment. Additionally mainly the bottom of buildings and ground parts of street poles are wrongly segmented as they have been added to the ground segment.

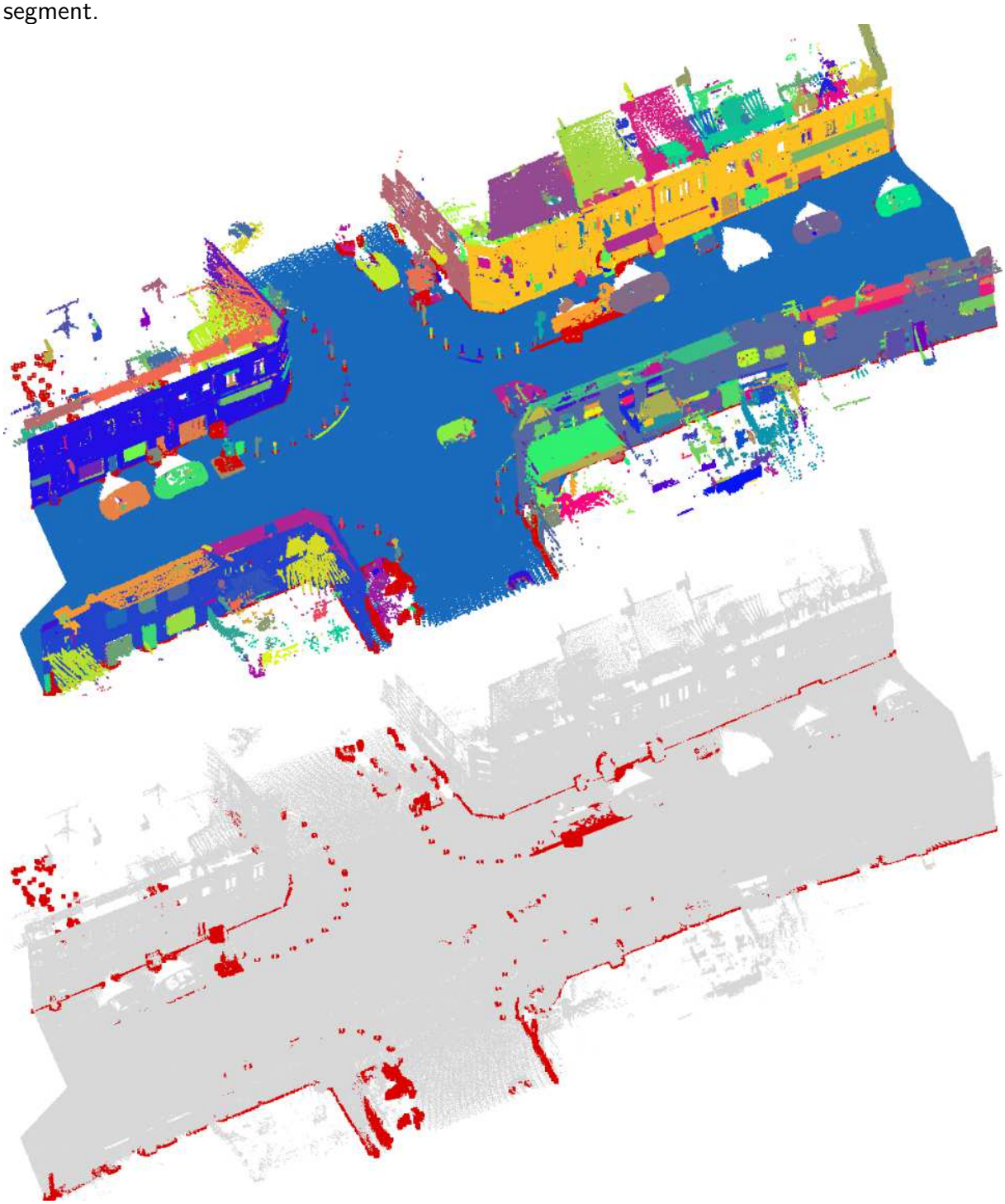

Figure 5.17: Segmentation result and under-segmentation error for the Lille dataset.

Ankeny

The Ankeny dataset is the only dataset used within this work containing additional colour information. Therefore not only orthogonal distance and normal difference but also two different metrics based on two different colour spaces, namely RGB and LAB, are compared.

For the geometrical weights, the fixed method seems to be the superior choice regarding overall completeness (Figure 5.18 left). While for the smallest scale both methods result in a completeness below 5%, for higher scales completeness rates for the fixed method are always above the ones obtained with the size-adaptive approach. In contrast to the presented results so far under-segmentation rates are much higher reaching up to 16%.

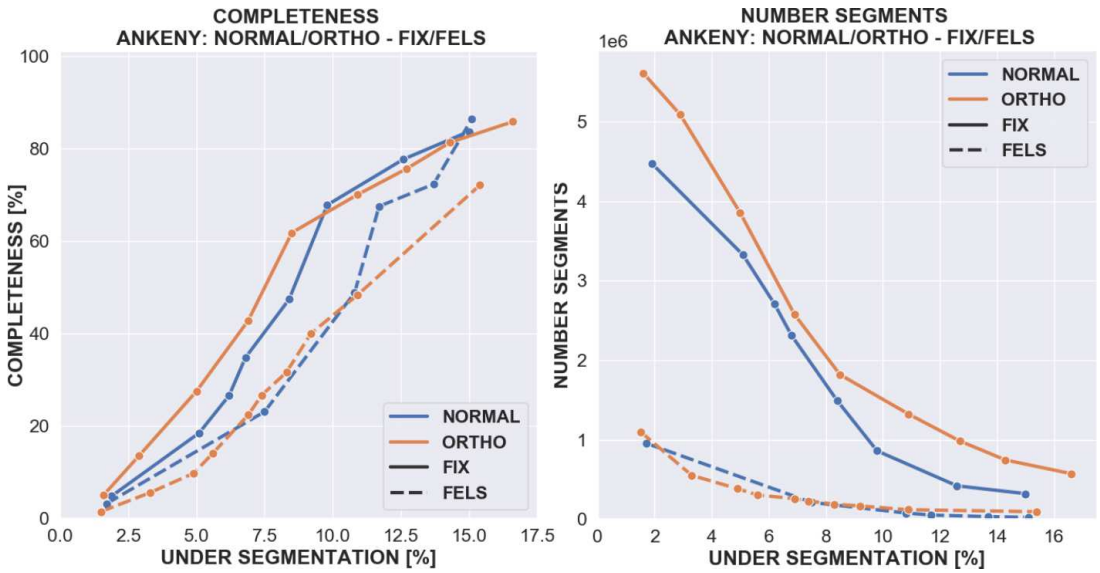

Figure 5.18: Comparison of the fixed and size-adaptive merging criteria for the Ankeny dataset. Left: overall completeness; Right: number of segments.

On the contrary looking at the number of segments one can observe that the size-adaptive method results in much less segments as the fixed approach. For the fixed method, the lowest scale in combination with the orthogonal distance results in approximately 5 millions segments whereas for the size-adaptive only in 1 million.

The original dataset contains around 8 million points indicating that for the fixed method there are a few larger segments whereas most of the remaining points are all in their own individual segment. Using the size-adaptive method it is possible to reduce the amount of data to 12.5% of the original size while only introducing an under-segmentation error of 1.5%.

Compared to the datasets presented so far two major differences exist:

1. All datasets showed urban sceneries whereas within this scene the dominant class is natural ground including grassland partly covered by vegetation.
2. All datasets have been acquired using an active sensor. This dataset is based on Dense Image Matching. Therefore, as already outlined in Section 2, only the top surface in overgrown vegetated areas will be captured.

Due to the higher resolution of the point cloud derived from DIM, more details are captured. In order to achieve completeness rates comparable to the other datasets would require increasing the scales significantly. But as it can be seen from Figure 5.18 at the same time the under-segmentation error increases significantly as well.

Therefore due to the nature of the point cloud itself in combination with the acquisition method, compared to all previously discussed datasets, both approaches using only geometrical weights are not well suited in order to obtain higher completeness rates. At least the size-adaptive method can be used to extract smaller homogeneous patches in order to reduce

the amount of features which might be useful for a subsequent classification step.

Looking at the results for both colour spaces more or less the same tendency can be seen (Figure 5.19). For smaller scales using the fixed method achieves higher completeness rates compared to the usage of geometrical weights. Besides that the size-adaptive method again results in lower completeness rates across all scales whereas the number of segments again is significantly lower compared to the fixed method.

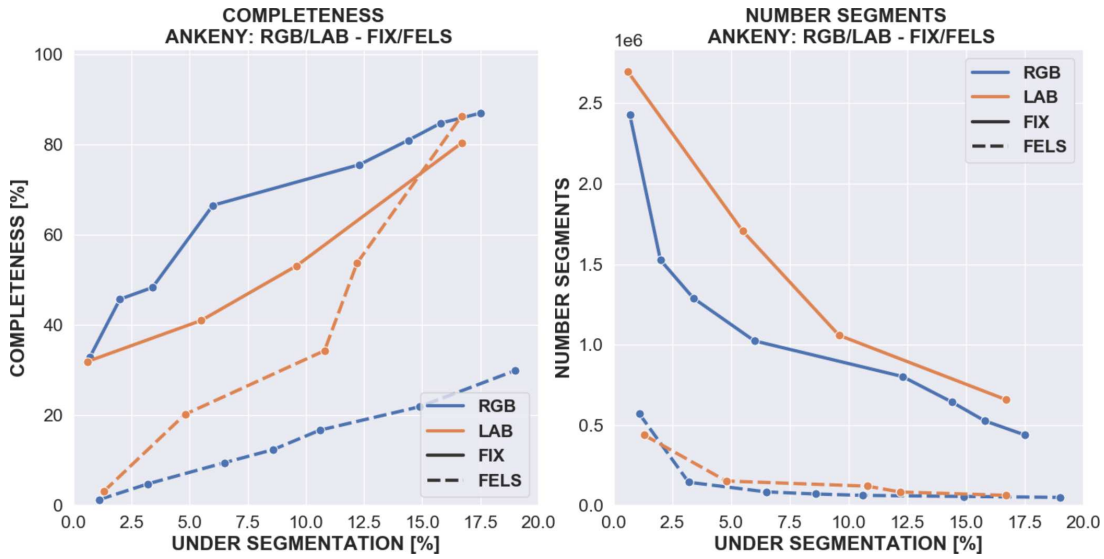

Figure 5.19: Comparison of the fixed and size-adaptive merging criteria for the Ankeny dataset.

While some of the higher under-segmentation for both colour and geometrical weights can be explained with the point cloud itself, another reason stems from the definition of the ground truth classes. So far no distinction within the ground was made, whereas within this dataset a distinction between street and ground solely based on colour is made. Therefore using only a geometrical attribute will not be able to distinguish those classes (Figure 5.20 - blue circle). Also the other way around, using only colour, white trucks parked along the street will be difficult to identify and separate (Figure 5.20 - red circle).

Figure 5.20: Ankeny dataset showing the problem using only colour or geometrical attributes for segmentation.

Therefore for both attributes high under-segmentation errors can be expected and, in order to be able to capture all ground truth classes properly, both geometrical and colour attributes need to be combined. Based on the previously conducted tests, for the geometrical weight the angle between normal vectors and for colour the Euclidean distance in the LAB colour space are used. Both weights show higher completeness rates compared to their respective alternative for the size-adaptive method.

As outlined in section 3.1.1.2 various methods exist to combine multiple attributes. In the following two approaches are evaluated:

INDIVIDUAL: This method conducts the segmentation individually for each attribute and the final segmentation is obtained by intersecting the individual results. This was suggested by [F. Felzenszwalb and P. Huttenlocher, 2004] for the segmentation of colour images. In Figure 5.21 this is referred to as 'INDIVIDUAL'.

COMBINED: The second approach introduces an individual weight for each attribute within a combined segmentation and only if both weights are below the respective scale, segments are merged. For this approach one has to decide, which weight is used to build the MST. In Figure 5.21 'NORMAL-LAB' represents the segmentation results where edges are sorted based on the angle between normal vectors and for 'LAB-NORMAL' the edges are sorted based on the Euclidean difference in the LAB colour space.

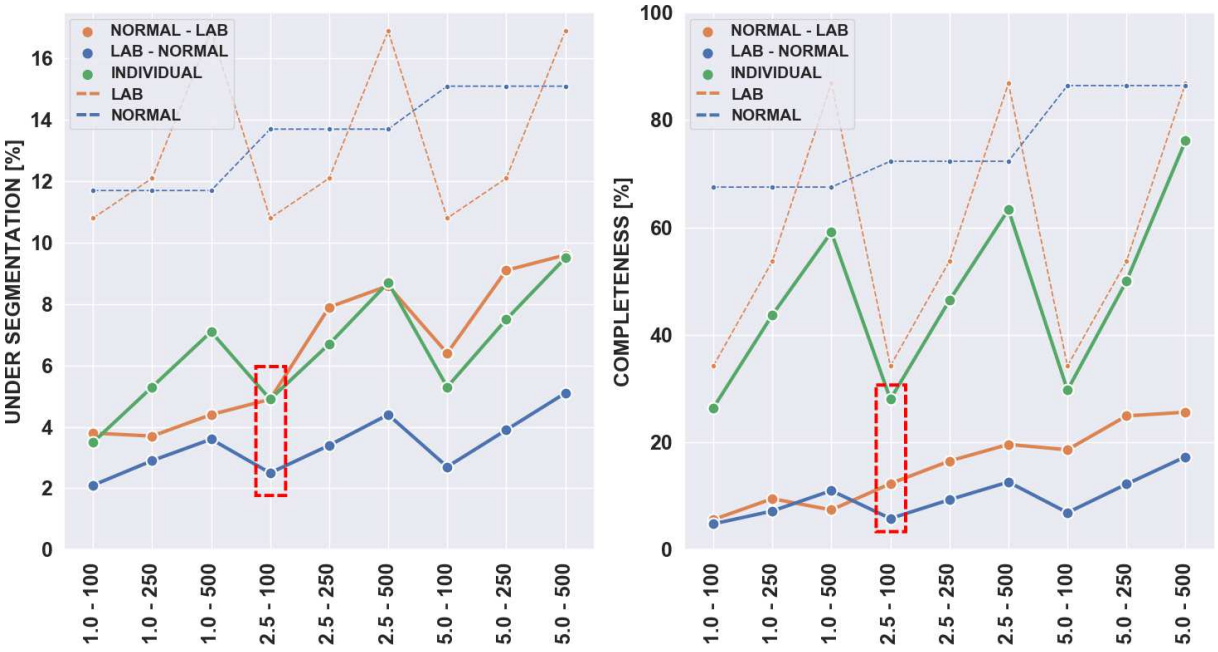

Figure 5.21: Under-segmentation and completeness for all possible attribute combination methods.

Figure 5.21 shows that the segmentation result with edges sorted based on the difference in colour space has across all scales the lowest under-segmentation but also the lowest completeness rates. The result with the individual segmentations shows similar under-segmentation errors as the segmentation with edges sorted based on the angle between normal vectors but with significantly higher completeness. The dashed lines represent the segmentation results if both the difference in normal vectors (blue) and difference in colour space (orange) are used completely individually. Comparing it with the result where both results are intersected (green) one can see that the combination of both is clearly superior.

A visual comparison of the results highlighted by the red rectangle in Figure 5.21 is shown in Figure 5.22. The result for the individual segmentation is already quite promising. Larger segments on the ground are recognizable, buildings and other smaller structures are separated. In contrast to both other approaches only small scattered segments have been extracted.

Looking at the under-segmentation one can observe that the major increase for the individual segmentation can mainly be explained by the larger areas within vegetation highlighted by the blue polygon. For buildings, street and objects the result is similar to the other approaches. Despite the larger under-segmentation error the individual approach, based on the visual analysis, seems to be the best method regarding the combination of colour and geometrical

attributes for the Ankeny dataset.

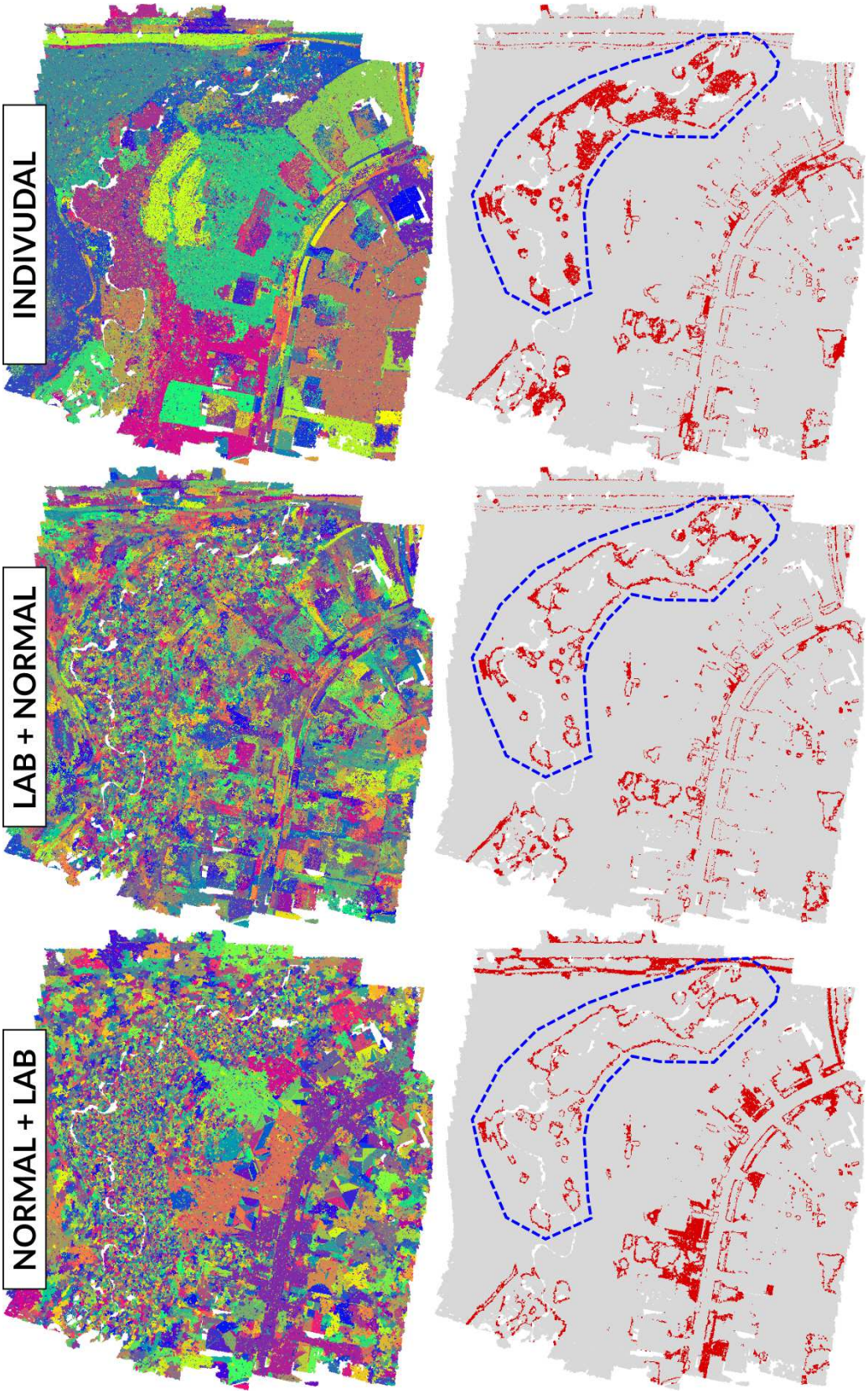

Figure 5.22: Segments and under segmentation for the different colour combination methods.

5.1.4 Minimum segment size

So far for none of the tests a minimum segment size has been enforced using the optional post-processing step. Therefore the influence of this step shall be evaluated as well.

While this is generally possible for any segmentation algorithm, using the minimum spanning tree as the basis for the segmentation comes with two big advantages regarding this step:

1. Segments found previously are guaranteed to be most homogeneous in respect to the used attribute.
2. Using the MST guarantees that in this post processing step each segment will be merged with the most similar adjacent segment as the edges are sorted by ascending weights.

Both points reduce the under-segmentation error introduced during this merging step. For the tests shown in Figure 5.23 the angle between normal vectors in combination with a medium scale are used. While the results are only shown for this combination, similar tendencies can be observed across all scales and attributes.

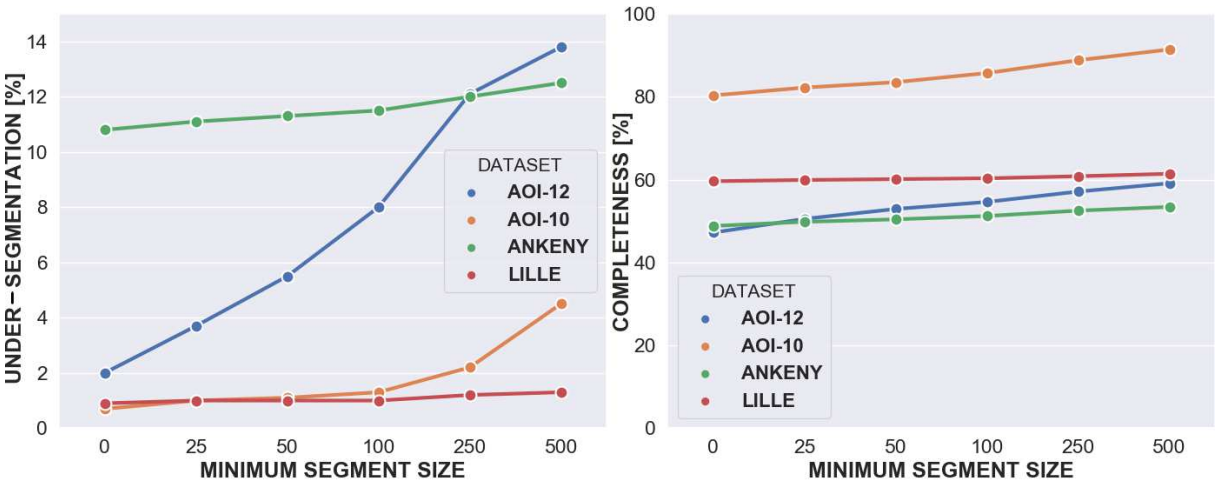

Figure 5.23: Influence of the minimum segment size on the segmentation result (using the angle between normal vectors)

The Aoi-10 dataset mainly contains individual trees, cars and flat terrain leading to a small under-segmentation error and already without an enforced minimum segment size an overall completeness of roughly 80%. While increasing the minimum segment size slightly improves completeness, especially for a minimum segment size above 100 points the under-segmentation error increases significantly.

Some objects like trees, due to the acquisition method and lower point density, are represented with less than 500 points. If such a large minimum segment size is chosen those objects are merged with the 'most' similar adjacent segment which in many cases is the ground segment (Figure 5.24).

A similar situation can be observed for the Aoi-12 dataset although already a minimum segment size of 25 significantly increases the under-segmentation error. In Figure 5.25 the roof structure from one of the building complexes from Aoi-12 is shown. In the point cloud coloured based on the ground truth classes (Figure 5.25 top left) one can see, that along the center of the roof (blue) smaller build up structures (violet) are located.

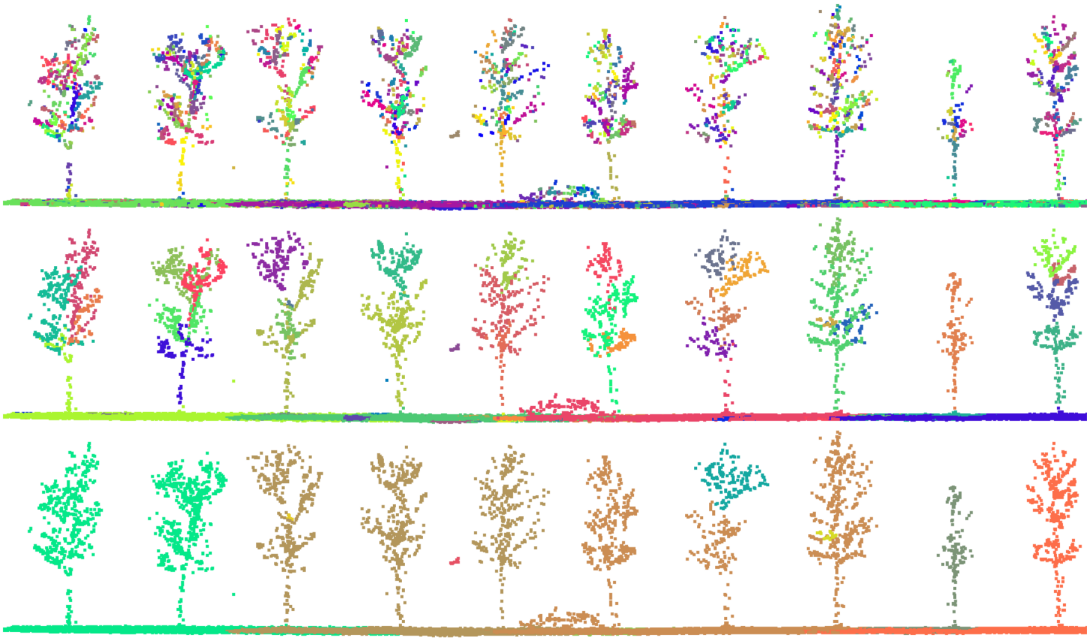

Figure 5.24: Influence of minimum segment size on vegetation in the Aoi-10 tile. Top: no minimum segment size; Middle: Minimum segment size of 50; Bottom: Minimum segment size of 500

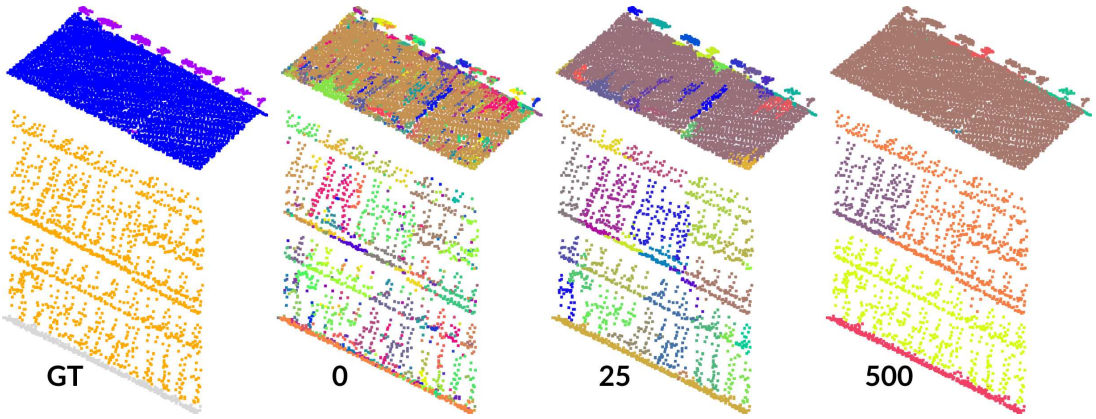

Figure 5.25: Influence of the minimum segment size on roof structures for the Aoi-12 dataset.

For the segmentation result without any enforced minimum segment size one can see many scattered single segments across all structures. Increasing the minimum segment size those scattered segments dissolve but already with an minimum segment size of 25 some of the small objects on the roof are merged with the adjacent roof structure. For the extreme case of a minimum segment size of 500 all of the build up structures are merged with the adjacent roof.

Also in the Lille dataset smaller objects like poles arranged along the street are merged with the ground segment if the segment size is chosen to large (Figure 5.26).

While the possibility to enforce an minimum segment size seems to be an useful option to remove smaller isolated segments (especially within vegetation) the minimum segment size has to be chosen carefully. It can be observed that too large minimum segment sizes result in an increased under-segmentation error due to objects having less points than the chosen threshold.

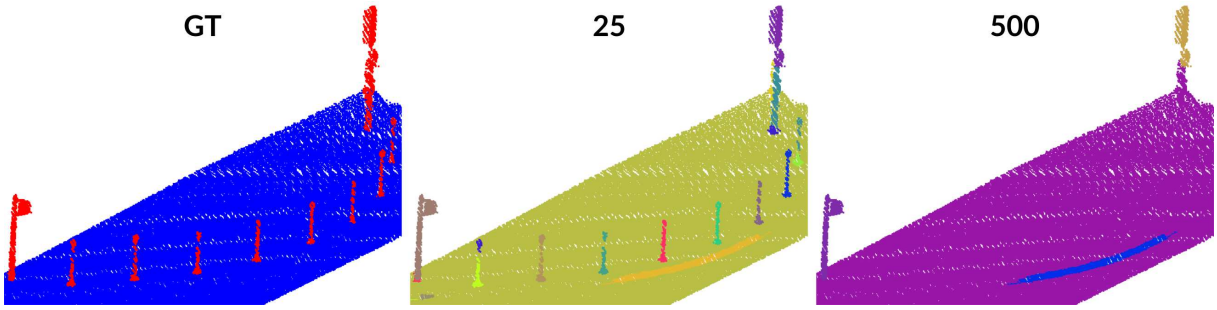

Figure 5.26: Using larger minimum segment sizes for the Lille dataset, poles along the street are merged with the street segment.

5.2 SLIC-3D

5.2.1 Examination for correctness

As for MST-3D the correctness of the implementation is tested against the scikit-image SLIC implementation using several images. In order to exclude disagreements caused by different seeding strategies, the scikit-image implementation was adjusted in order to allow providing initial seed positions. This guarantees that both implementations start from the same initial seeds.

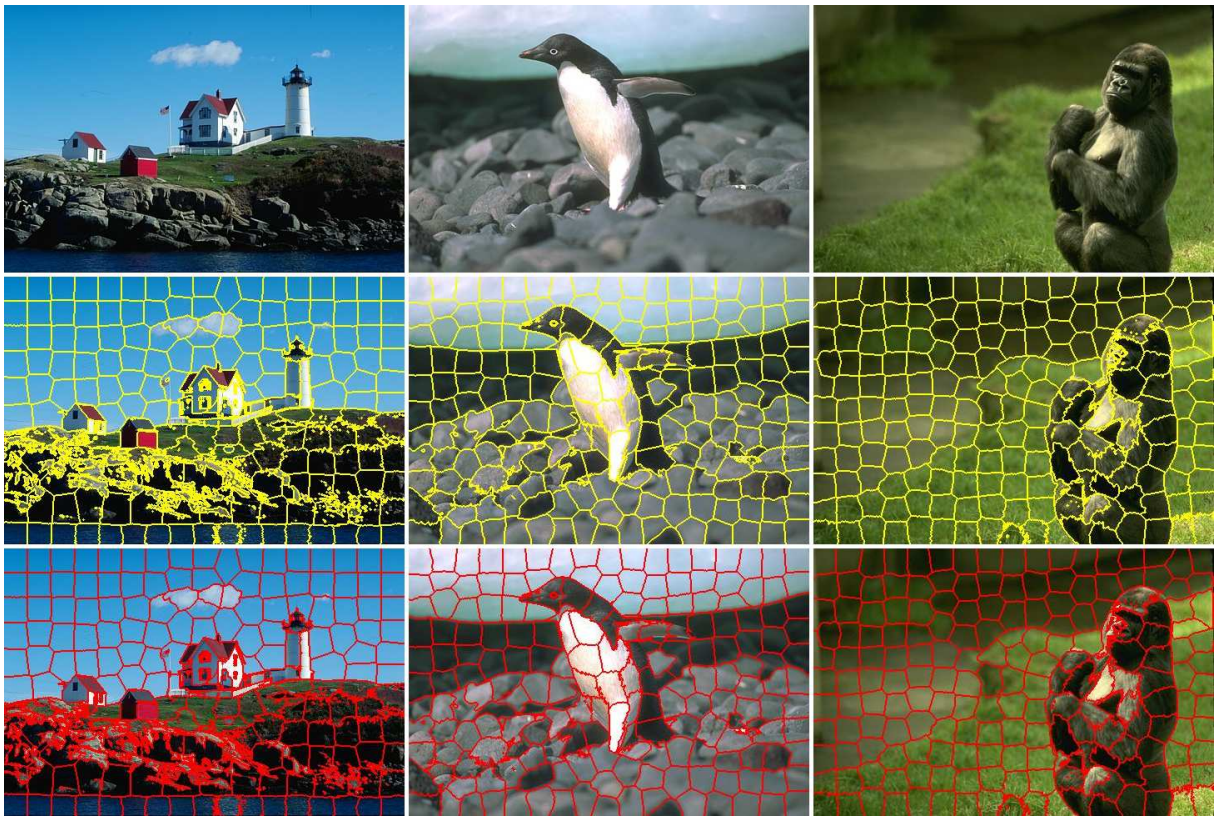

Figure 5.27: Segmentation result for the scikit-image (middle row) and SLIC-3D (bottom row) implementation.

All tests have been conducted with different numbers of segments, various compactness factors and both colour spaces. All tests showed, as for MST-3D by comparing the obtained segment borders, that both implementations produce exactly the same segments.

5.2.2 Iterations

Being an iterative method, the optimal number of iterations, if possible, should be determined in order to prevent unnecessary iterations optimizing runtime of the algorithm. As already mentioned earlier, [Achanta et al., 2012] state that for images in most cases 10 iterations suffice until the positions of the seeds converge.

Based on the current position of each seed, for each iteration step the Euclidean difference in consecutive iterations is calculated. In theory with each iteration step the overall mean distance should get smaller as each seed point converges to its final position. Both for the segmentation using the angle between normal vectors and the orthogonal distance a similar tendency is observable which can be seen in Figure 5.28.

The same tests have been conducted for all available reference datasets with different seed resolutions, distance combinations and compactness factors indicating that in all cases the seed positions converge within 15 to 20 steps. Due to the varying point densities of 3D point clouds no general stopping criterion based on the mean distance could be established.

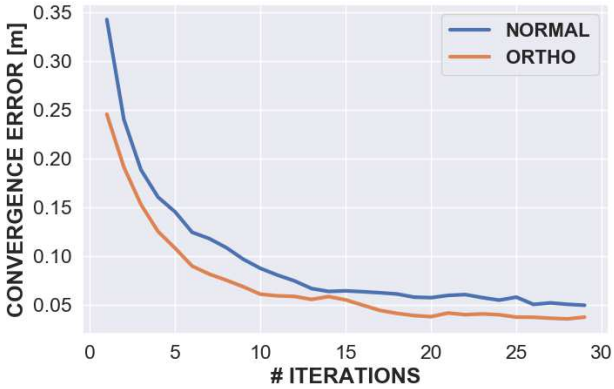

Figure 5.28: Mean angle between consecutive seed positions for the Ankeny dataset.

Another option is to count the number of points which in consecutive iterations different labels have been assigned. In theory this number should reach zero. Tests showed that this assumption does not hold hole true due to points at segment boundaries, that may get different labels assigned in each iteration. As for the mean seed difference the amount of points varies strongly with the number of segments and point density. Therefore again no general stopping criteria could be defined.

5.2.3 Post-processing

For MST-3D connectivity within segments is already enforced as the segmentation is directly based on the connectivity graph. In the case of SLIC-3D, due to the iterative procedure, segments are split into multiple smaller parts especially if small compactness values are chosen. Therefore it is necessary to conduct a post-processing step, where smaller parts are merged with adjacent regions.

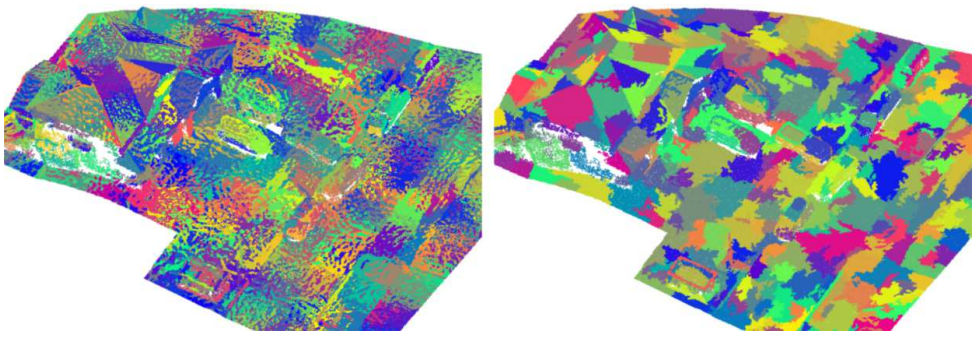

Figure 5.29: Raw segments (left) and after the post-processing (right)

As for the post-processing step in the MST-3D implementation, again the RAG is created based on the raw segments obtained from the initial SLIC-3D segmentation. Segments being smaller than a certain size threshold are merged with the most similar adjacent region. Figure 5.29 shows an extreme example. On the left the initial segmentation is shown. One can observe that segments are massively cluttered and no clear structures are recognizable. The segmentation on the right shows the result after the post-processing step where segments are again homogeneous patches.

5.2.4 Comparison with VCCS

According to [Papon et al., 2013] the chi squared distance of the fast point feature histograms (FPFH) is used as geometrical distance. But the implementation available in the pcl library uses the angle between normal vectors as geometrical distance. Due to this fact it is possible to directly compare VCCS with SLIC-3D to verify the influence of the voxelization in terms of under-segmentation.

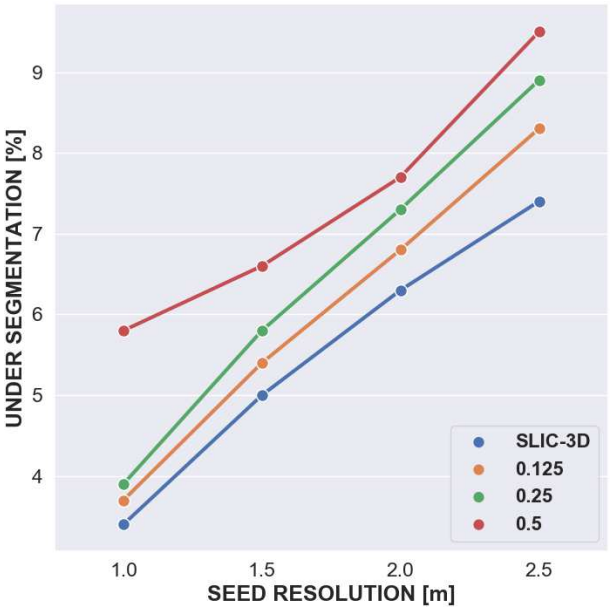

Figure 5.30: Comparison between VCCS and SLIC-3D.

While the same definitions for spatial, colour and geometrical distances are used, two minor differences exist that have an influence onto the result of the comparison.

The first one concerns the initial seeds themselves. Unfortunately it is not possible to supply initial seed positions to the VCCS algorithm. Therefore the initial 'raw' seed positions might be different to the ones used within SLIC-3D.

The second difference occurs due to the way the seeds and there attributes are updated. In both implementations, for the coordinates and colour the respective mean of all points belonging to each seed are used. Regarding the normal vector in SLIC-3D the normal vector with the lowest s_0 is used as reference. In VCCS it seems that the mean normal vector is calculated.

Besides these differences, Figure 5.30 shows the under-segmentation error with respect to the seed resolution and method. For VCCS the results for different voxel resolutions (0.125m, 0.25m and 0.5m) are shown. In all cases SLIC-3D shows the lowest under-segmentation error. While for smaller voxel and seed resolutions the difference is smaller, as expected the differences increase significantly with increasing resolutions.

The influence of the voxelization for a small part of the Ankeny dataset is shown in Figure 5.31. The scene includes an area where cars are parked in front of a building. For both segmentation results a seed resolution of 2.5m was used. The voxel resolution in the left figure was set to 50cm. While for the SLIC-3D result (right) segments nicely adhere to the bodies of the cars, for the VCCS result, as highlighted with the black circles, segments contain points from the cars and ground.

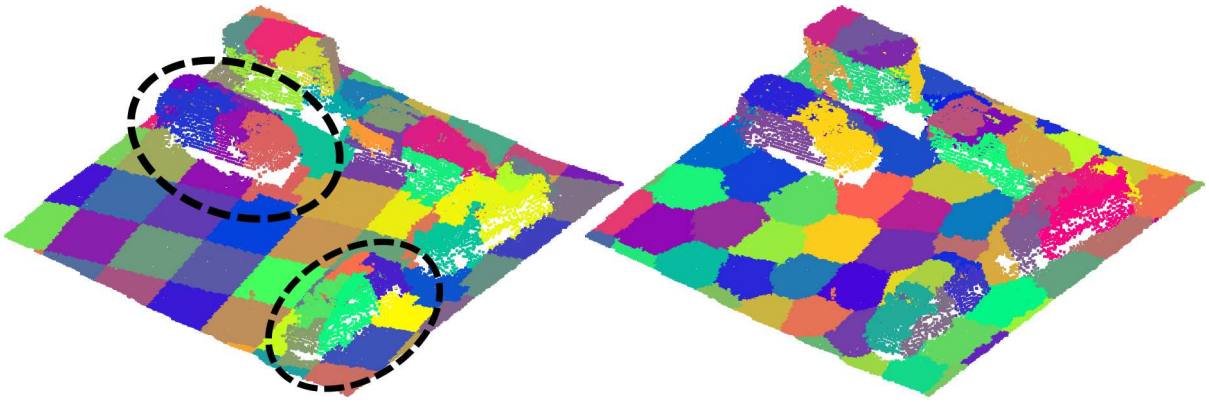

Figure 5.31: VCCS with 50cm voxel resolution (left) and SLIC-3D (right).

5.2.5 Resolution

Due to the design of SLIC-3D the seed resolution not only defines the number of segments but also the achievable level of detail regarding the segmentation. While using a smaller seed resolution (25 or 50cm) enables the detection of smaller objects, smoother surfaces or larger objects are divided into many small segments as well. Increasing the resolution reduces the amount of segments while at the same time loosing the possibility to detect smaller structures.

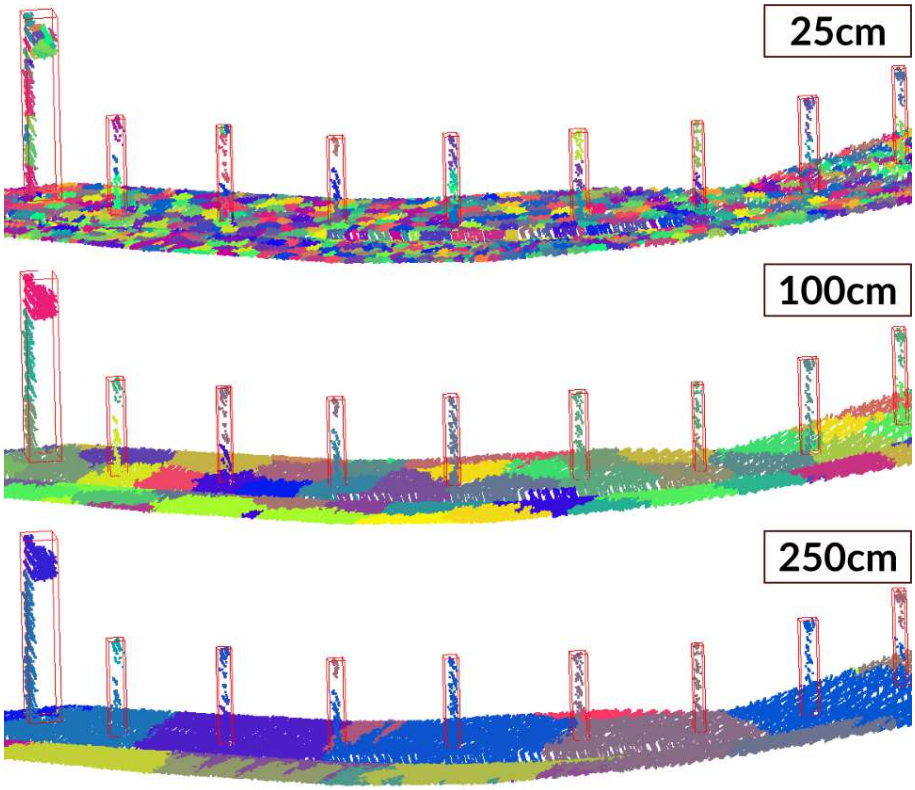

Figure 5.32: Segmentation results using different seed resolutions.

This shall be demonstrated for a small section of the Lille dataset showing the side of the road where some smaller poles are located. The segmentation for three different seed resolutions (25cm, 100cm, 250cm) is shown in Figure 5.32 where the true locations of the poles are highlighted by the red bounding boxes. For the highest resolution (25cm) segments representing poles are separated from the ground while the poles themselves are also split into multiple segments. Already for 1m some poles are merged with the ground whereas for 2.5m none of

the poles is properly segmented.

Therefore the selection of the seed resolution not only depends on the desired number of segments, but also strongly on the objects contained in the point cloud and which level of detail the user wants to achieve. Especially if sizes of the objects vary strongly, as it is the case for most point clouds, the regular seeding approach of SLIC-3D is a major limitation. Especially the design of SLIC-3D in its current form does not allow for a more data driven seeding approach.

This may be the greatest disadvantage compared to the MST-3D approach. Within MST-3D no minimum segment size is enforced as this is an optional step. The segmentation is solely data driven and does not depend on any seed points. Therefore the MST-3D approach has the flexibility to extract larger segments in smoother areas and smaller segments in inhomogeneous areas.

5.2.6 Compactness

The influence of the compactness factor shall be shown for a roof of a building from the Ankeny dataset using the angle between normal vectors. In Figure 5.33 the individual parts of the roof are highlighted with dashed black lines.

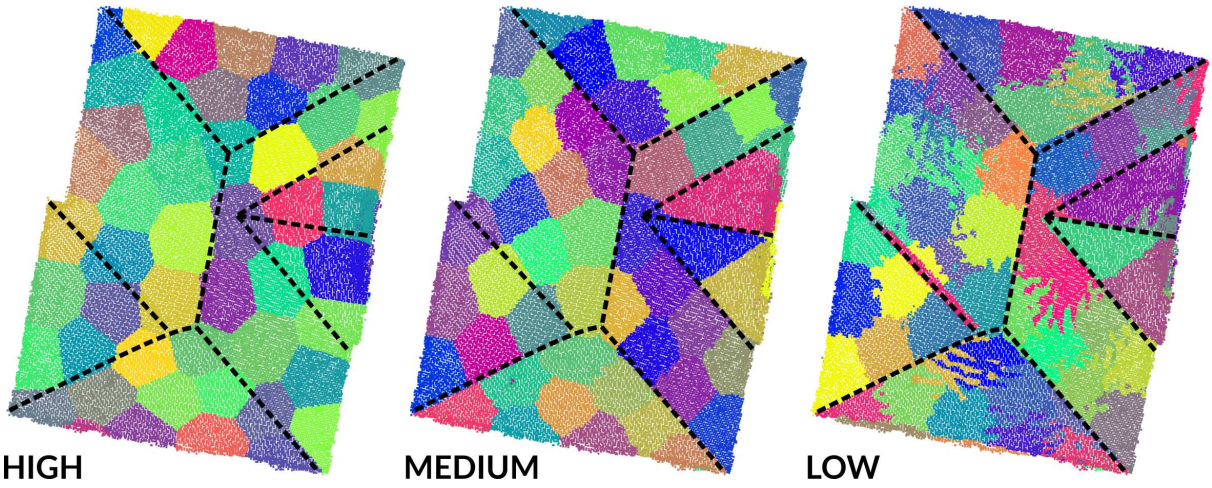

Figure 5.33: Influence of the compactness factor.

While using a high compactness factor the segments in Figure 5.33 do not represent any of the individual parts of the roof, with decreasing compactness one can observe that less segments cross the individual roof parts and better adhere to the boundaries. For the lowest compactness the regular shape of the segments is lost.

5.2.7 Feature Distance

While for MST-3D completeness was used to evaluate the influence of the different weights, in the case of SLIC-3D the under-segmentation error with respect to the voxel resolution is used. Due to the design of SLIC-3D and the resulting strong over-segmentation completeness values will be small making them not usable for comparison.

For each feature distance a low, medium and high compactness factor was chosen in order to be able to evaluate the influence of this parameter as well.

5.2.7.1 MA41

Due to the point density of these datasets a seed resolution below 1m does not make any sense as this would result in segments with sizes below a few points. For the Aoi-12 tile a resolution of 1m already results in a mean segment size of 10, respectively 60 for the Aoi-10 tile. For both tiles the combination of Euclidean and orthogonal distance is superior compared to the combination using the angle between normal vectors.

Comparing the achievable under-segmentation error using the highest seed resolution with the segmentation results from MST-3D one can observe that in the case of these datasets the strong over segmentation from SLIC-3D results in the same under-segmentation error as for MST-3D where much higher completeness rates are achieved. This indicates that in the case of these 'lower' resolution datasets, data reduction using the current implementation is only of minor use.

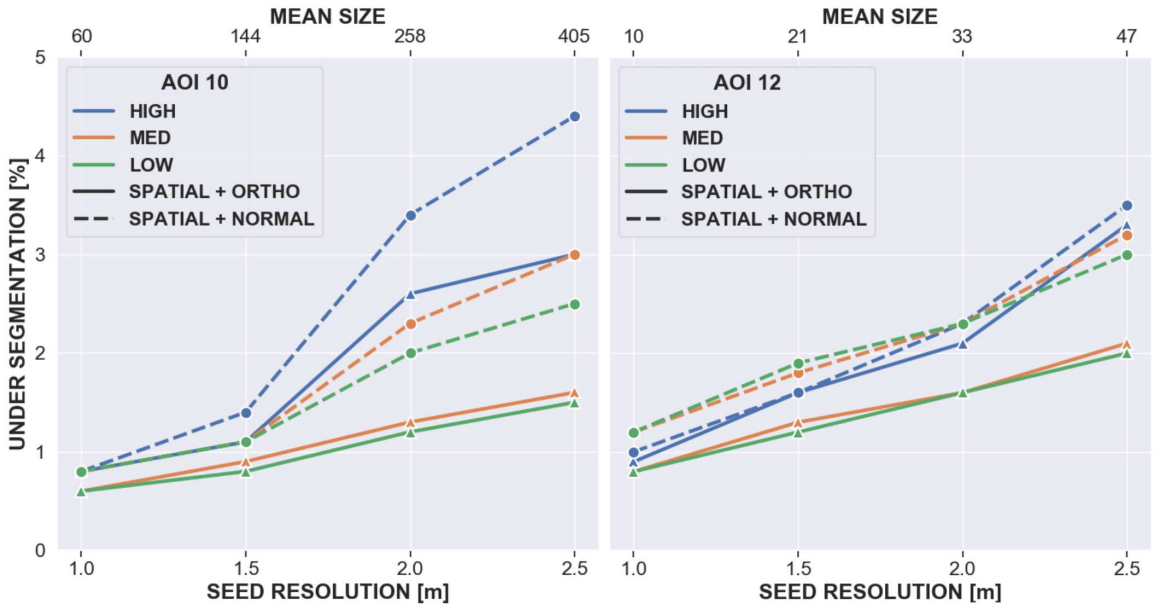

Figure 5.34: Under-segmentation error for both MA41 tiles for different seed resolutions.

5.2.7.2 Lille

In the case of the Lille dataset, for low and medium compactness factors only minor differences in terms of under-segmentation can be seen in Figure 5.35 comparing orthogonal distance and difference in normal vectors.

Using a seed resolution of 50cm results in a mean segment size of 371, being 4 times larger compared to the size using a seed resolution of 25cm. At the same time the under-segmentation error between both only increases marginally.

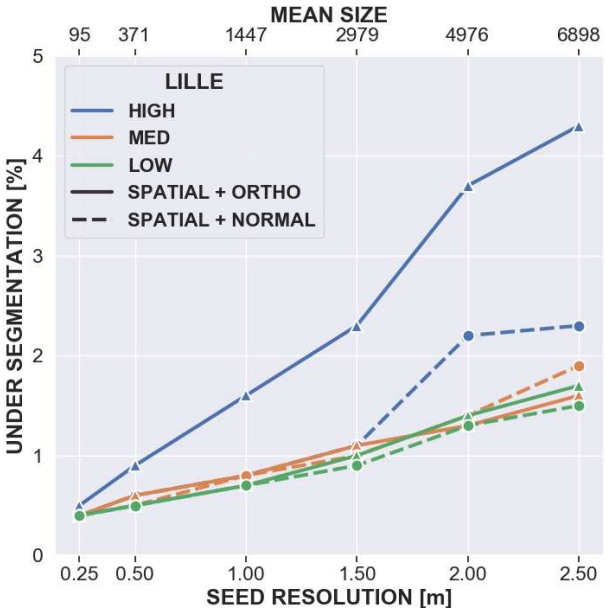

Figure 5.35: Under-segmentation for Lille.

Especially in this case SLIC-3D seems to be a sound method to extract homogeneous compact segments in order to reduce the amount of data without introducing a large under-segmentation error. Using a seed resolution of 50cm an under-segmentation error below 1% is introduced while the number of features reduces from originally 7.5 million points to 200.000.

An average segment size of 400 points already would allow for the calculation of robust attributes (second order statistics, FPFH) which might be helpful in a subsequent classification step compared to a classification solely based directly on the points.

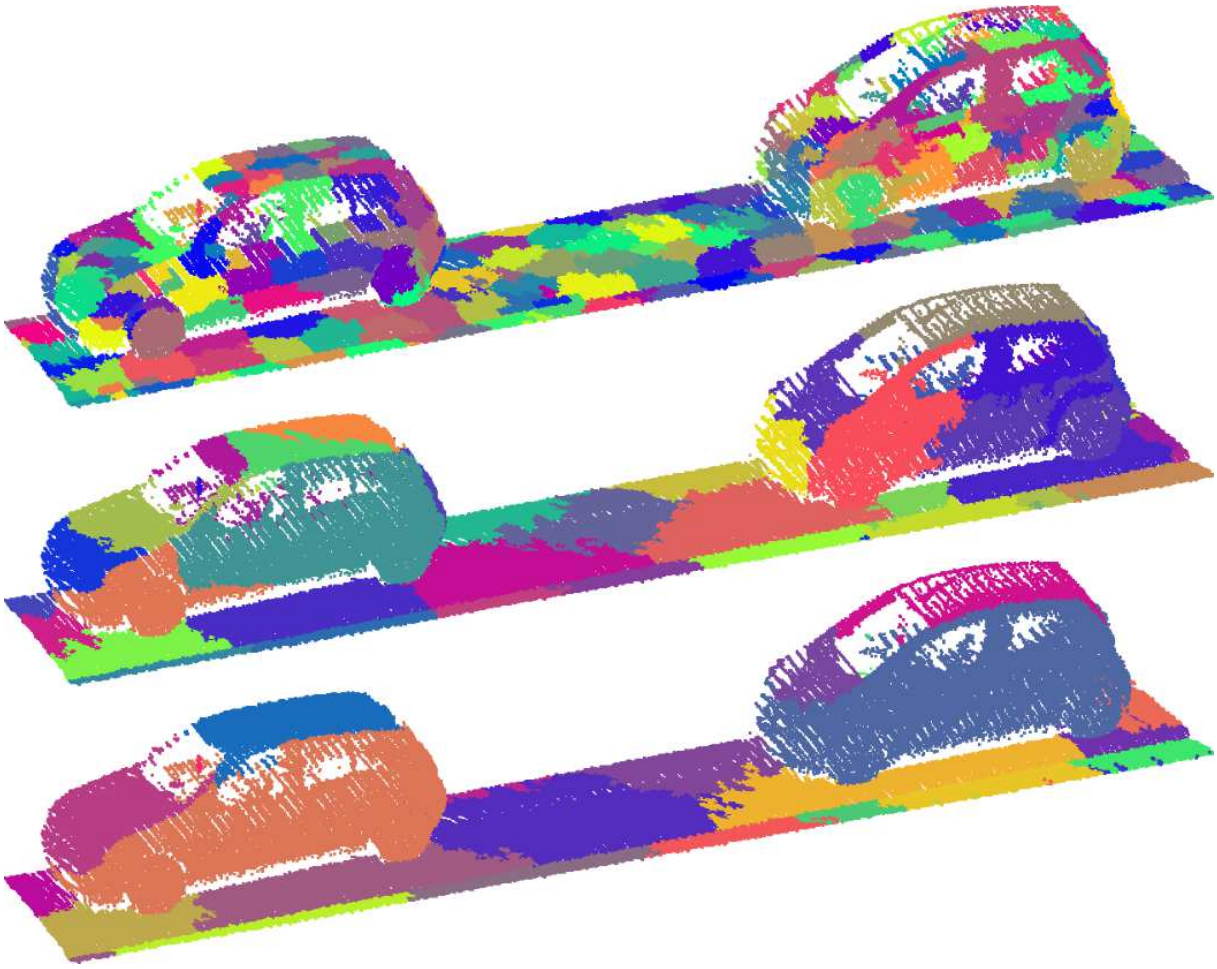

Figure 5.36: Segmentation result for cars parked along the street for the Lille dataset. Different seed resolutions: 50cm (top), 150cm (middle), 250cm (bottom).

In Figure 5.36 the segmentation results for different seed resolutions for a small part of the Lille dataset is shown. For all cases the segment boundaries adhere well to object boundaries and one can nicely observe how the seed resolution influences the size of the resulting segments. While for the highest seed resolution (50cm) cars are split into multiple individual segments, with increasing seed resolution segments better adhere to complete parts of the cars. For a seed resolution of 250cm the left car is nicely segmented into its side, top and bonnet. At the same time, as we have seen in Figure 5.32 increasing the seed resolution leads to problems with smaller objects.

5.2.7.3 Ankeny

Each possible combination of geometrical and colour attribute is analysed in order to find the best one with respect to the introduced under-segmentation error.

Based on Figure 5.37 the combination using the LAB colour space and the orthogonal distance shows the lowest under-segmentation error across all seed resolutions. While the differences for the combinations is only minor for the highest seed resolution of 1m, the difference grows with increasing seed resolution.

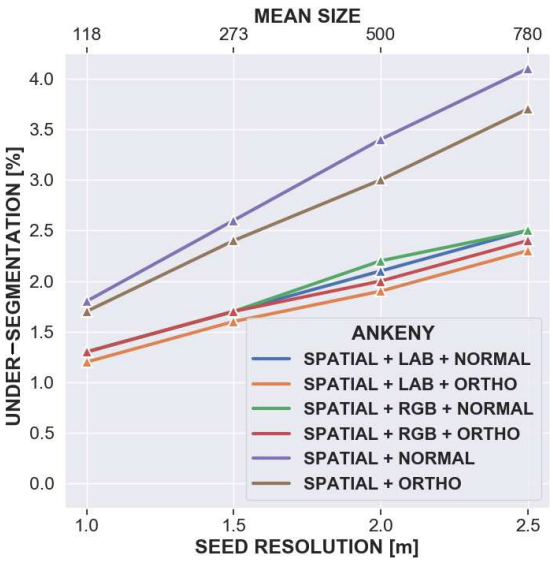

Figure 5.37: Under-segmentation for the Ankeny dataset using SLIC-3D.

For comparison the results using only geometrical weights (violet and brown) are shown as well. Across all resolutions the combination of colour and geometrical weights is clearly superior which is not surprising due to the same reasons outlined for the MST-3D approach regarding the combination of attributes.

This further shall be illustrated with an example shown in Figure 5.38. A situation from the Ankeny dataset is shown, where objects can either only be separated by colour (grassland and street) or by geometrical weight (individual roof parts). The borders of these regions are highlighted as dashed black lines.

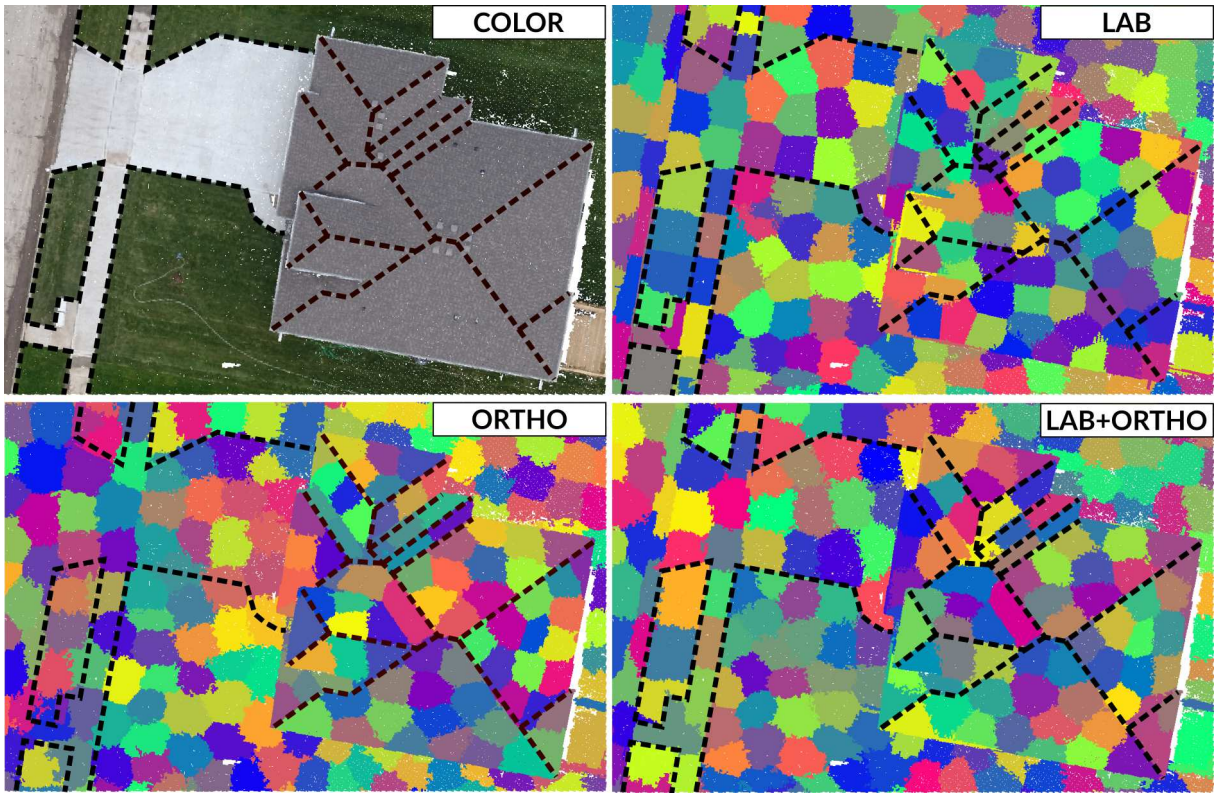

Figure 5.38: Combination of colour and geometrical attributes for the Ankeny dataset

Using only colour (Figure 5.38 - top right) segments adhere well to the borders of grassland / street whereas the individual roof parts are not distinguishable at all. For the orthogonal distance (Figure 5.38 - bottom left) the situation is exactly the other way around. Only the individual roof parts can be recognized whereas segments do not adhere to the different land coverage. Only using a combination of both (Figure 5.38 - bottom right) all the highlighted areas are segmented well.

5.3 Combination of SLIC-3D and MST-3D

So far SLIC-3D and MST-3D have been applied individually and the combination of both methods seems to be the logical next step. Now the question arises if the combination of both methods improves the segmentation. This test shall serve as an example to show how a strong over-segmentation obtained by SLIC-3D might help in subsequent steps.

For this test the Ankeny dataset has been chosen as the tests conducted so far showed that this dataset is the most challenging one. The combined segmentation includes the following steps:

1. In the first step a strong over-segmentation is obtained by using SLIC-3D. The segmentation is conducted individually for the geometrical and colour attribute.
2. Instead of operating directly on the point cloud, the RAG based on the segments obtained from SLIC-3D is segmented using the MST-3D approach. For each segment the normal vector again is selected based on the lowest s_0. In case of colours the arithmetic mean is used.
3. The final segments are obtained by intersecting the individual results.

In the following the 'individual' approach represents the MST-3D approach directly segmenting the point cloud whereas with 'slic-individual' the combined approach of SLIC-3D and MST-3D is meant.

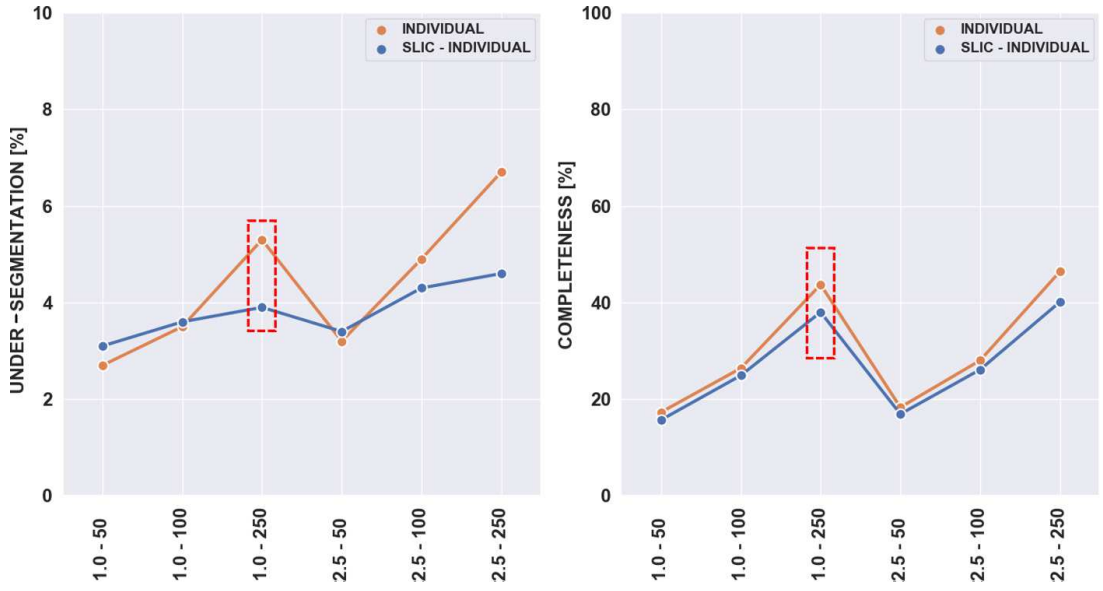

Figure 5.39: Comparison of 'original' and combined segmentation.

In Figure 5.39 completeness and under-segmentation for both approaches are shown. Except for the smallest colour and geometrical attribute the combination of SLIC-3D with MST-3D

results in lower under-segmentation whereas completeness reduces only marginally.

A visual comparison of the results in Figure 5.40 already reveals that for the combined approach much more homogeneous segments are obtained. Especially for ground where MST-3D alone results in numerous small individual segments the combined approach is superior. At the same time the combined approach is able to extract individual parts of the roofs while for the approach without SLIC-3D the segments on the roofs are strongly irregular in shape and size.

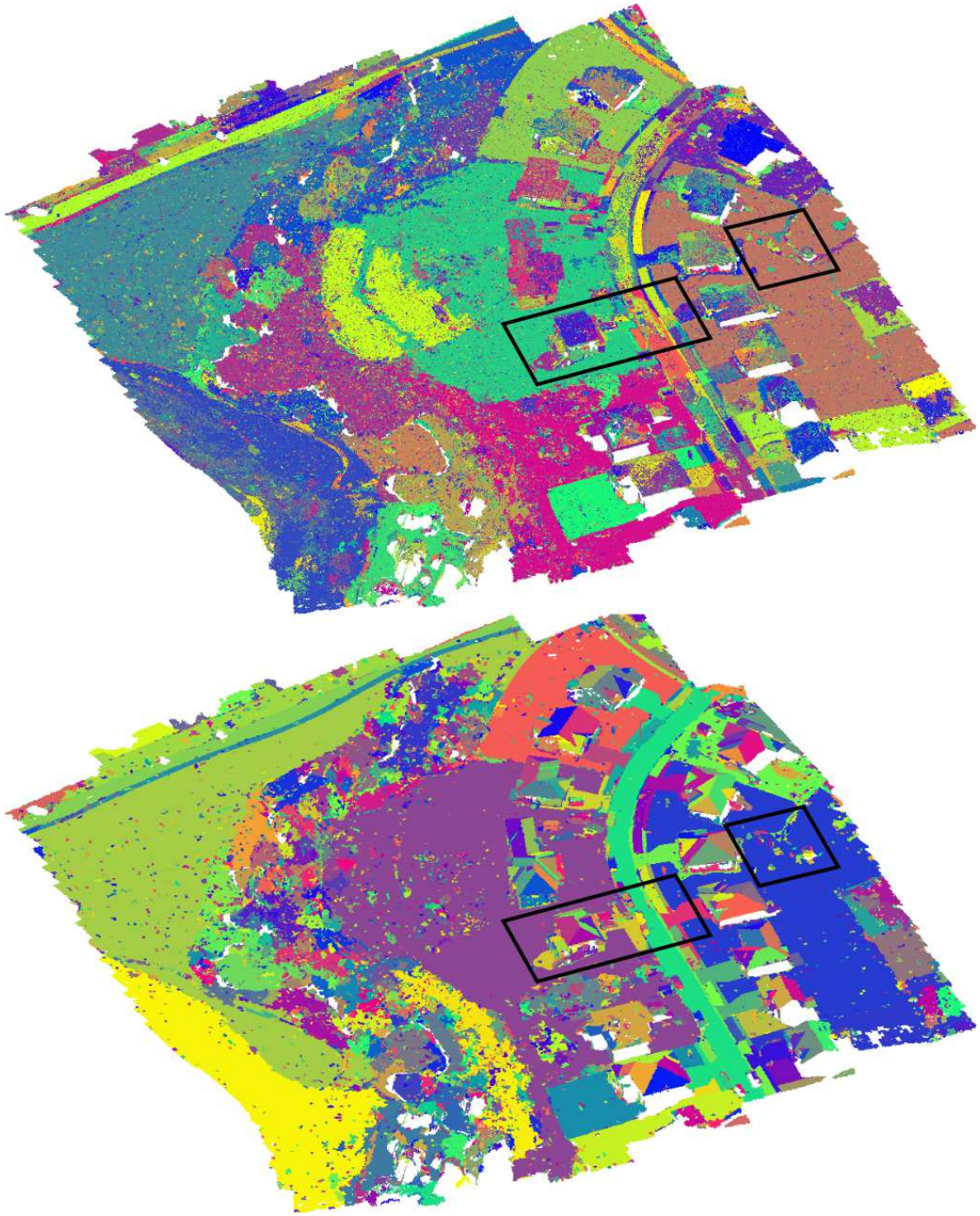

Figure 5.40: Comparison of 'original' and combined segmentation.

Two regions are highlighted in Figure 5.40 which shall be visually analysed in some more detail. The first region contains a building, cars and parts of the street while the second region includes some smaller objects located on the grassland.

Figure 5.41 further confirms the impression gained by looking at the overview image. For the segmentation without using SLIC-3D all objects contain small isolated segments. Both grassland and street are not as well segmented as for the combined approach where especially the borders of the different regions are much sharper. At the same time smaller objects like the cars are also well separated within the combined approach.

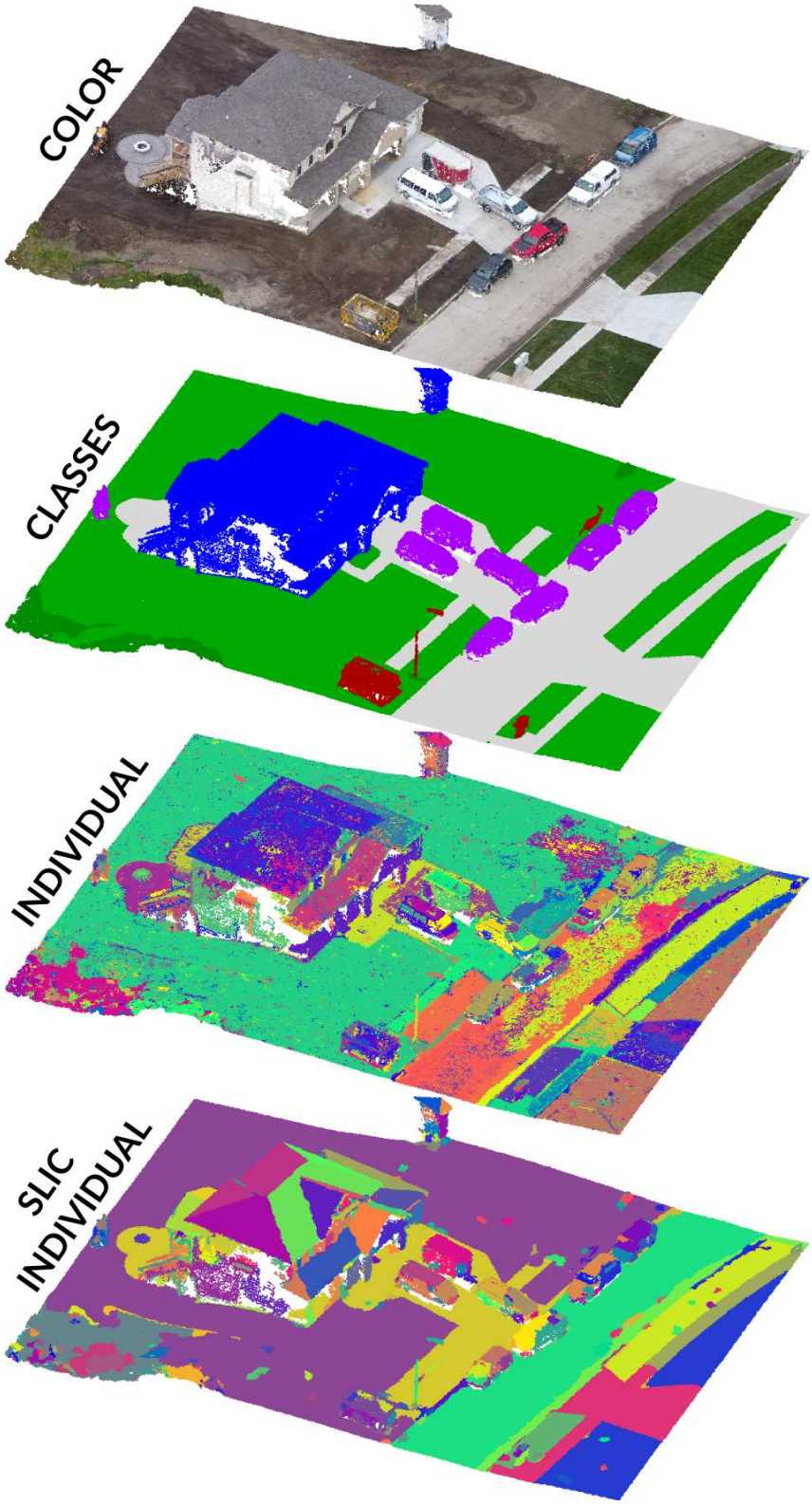

Figure 5.41: Detailed view of a selected region for the Ankeny dataset.

In Figure 5.42 the second area of interest is shown in some more detail. All trees and other

objects are well recognizable but especially ground is much better segmented for the combined approach. While for the approach without SLIC-3D numerous smaller segments and scattered points are visible for the combined approach all of the visible ground is within one segment.

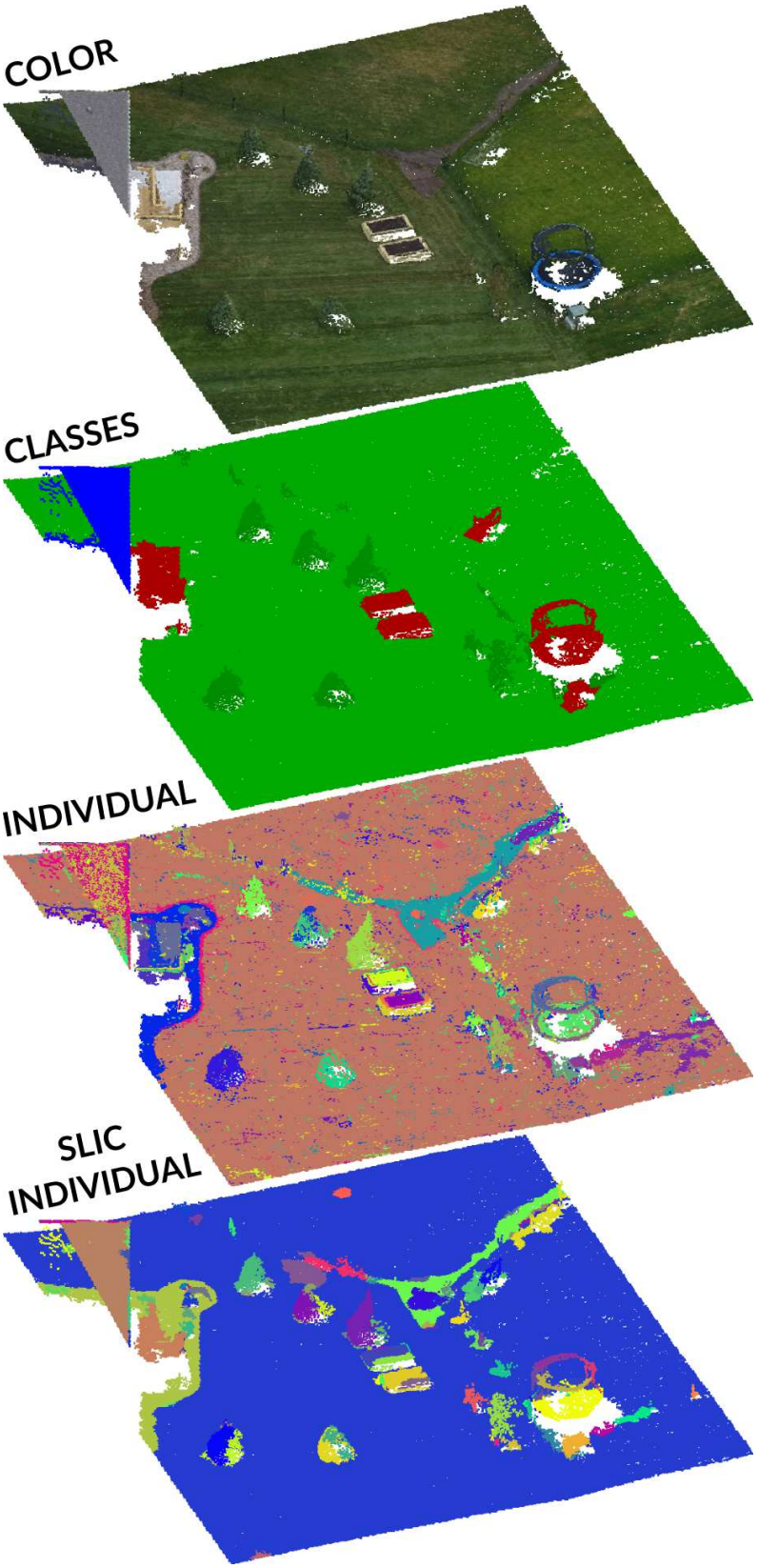

Figure 5.42: Detailed view of a selected region for the Ankeny dataset.

6. Conclusion

Two segmentation approaches, so far mainly used for image segmentation, have successfully been extended to 3D point clouds.

For the evaluation two metrics, under-segmentation and completeness, were used. With the definition of completeness based on individual objects an improved measure was introduced. Especially in case of large segment size variation completeness gives a better impression of the true quality of segmentation.

The results for the MST-3D segmentation method are promising for all tested datasets. Regarding completeness it seems that the current implementation serves as a good starting point towards an object-based segmentation process or semantic classification. Especially the possibility to segment objects having inhomogeneous attributes (for example vegetation) is a clear advantage compared to traditional region growing methods.

One disadvantage of the approach is the definition of the threshold. In the case of classical region growing approaches the user-defined threshold directly represents e.g. the geometrical limit (e.g. difference between two normal vectors) and is easy to understand for the user. In the case of MST-3D the threshold is only indirectly used losing its geometrical interpretability making it more difficult to find suitable values.

A possible extension for the future might be an iterative procedure. In the first step the point cloud is segmented as seen within this thesis but with a small threshold leading to a strong over-segmentation. Afterwards the segmentation is iteratively repeated on the obtained segments where in each step the threshold is increased. Using this approach allows deriving hierarchical features (useful for classification) and the computation of more more object related features (FPFH).

For SLIC-3D the situation is somewhat different. While for the datasets with a higher point density the strong over-segmentation of this approach may serve as a valuable basis for subsequent processes, in the case of the ALS datasets (having lower point density) the approach does not result in useful segments. Especially in the last example for the Ankeny dataset one could observe, that the combination of SLIC-3D with MST-3D improved the segmentation result compared to MST-3D alone.

Especially the regular seeds are a major drawback for point clouds. While images have a regular grid structure, the data distribution of point clouds is irregular. Additionally strongly varying object sizes are problematic for the regular seeding approach as well. Nevertheless the elegance of this approach lies in its simplicity using regular seeds and weighted feature distances. Combining this technique with a more flexible and adaptive seeding approach will further improve the segmentation.

For complex scenes, the combination of both segmentation algorithms can produce improved results as shown in the last example using the Ankeny dataset.

List of Figures

List of Tables

References

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):8. 2274 – 2282.

[Behnel et al., 2011] Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., and Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39.

[Ben-Shabat et al., 2018] Ben-Shabat, Y., Avraham, T., Lindenbaum, M., and Fischer, A. (2018). Graph based over-segmentation methods for 3d point clouds. *Computer Vision and Image Understanding*, 174:12–23.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Bruggisser et al., 2019] Bruggisser, M., Hollaus, M., Wang, D., and Pfeifer, N. (2019). Adaptive framework for the delineation of homogeneous forest areas based on lidar points. *Remote Sensing*, 11:189.

[Che et al., 2019] Che, E., Jung, J., and Olsen, M. J. (2019). Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, 19(4).

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:273–297.

[Csillik, 2017] Csillik, O. (2017). Fast segmentation and classification of very high resolution remote sensing data using slic superpixels. *Remote Sensing*, 9(3).

[Doneus et al., 2008] Doneus, M., Briese, C., Fera, M., and Janner, M. (2008). Archaeological prospection of forested areas using full-waveform airborne laser scanning. *Journal of Archaeological Science*, 35(4):882 – 893.

[Dorninger and Pfeifer, 2008] Dorninger, P. and Pfeifer, N. (2008). A comprehensive automated 3d approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. In *Sensors*.

[F. Felzenszwalb and P. Huttenlocher, 2004] F. Felzenszwalb, P. and P. Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181.

[Filin and Pfeifer, 2005] Filin, S. and Pfeifer, N. (2005). Neighborhood systems for airborne laser data. *Photogrammetric Engineering & Remote Sensing*, 71:743–755.

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

[Gao et al., 2017] Gao, G., Lauri, M., and Frintrop, S. (2017). Saliency-guided adaptive seeding for supervoxel segmentation.

[Golovinskiy and Funkhouser, 2009] Golovinskiy, A. and Funkhouser, T. (2009). Min-cut based segmentation of point clouds. In *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*.

[Grilli et al., 2017] Grilli, E., Menna, F., and Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3:339–344.

[Hirschmuller, 2008] Hirschmuller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341.

[Hu et al., 2017] Hu, S., Li, Z., Zhang, Z., He, D., and Wimmer, M. (2017). Efficient tree modeling from airborne lidar point clouds. *Computers and Graphics*, 67:1 – 13.

[Jones et al., 2001] Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. [Online; accessed ¡today¿].

[Kraus and Pfeifer, 2011] Kraus, K. and Pfeifer, N. (2011). Advanced dtm generation from lidar data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34:23–30.

[Kruskal, 1956] Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50.

[Liang et al., 2014] Liang, X., Hyyppä, J., Kukko, A., Kaartinen, H., Jaakkola, A., and Yu, X. (2014). The use of a mobile laser scanning system for mapping large forest plots. *IEEE Geoscience and Remote Sensing Letters*, 11(9):1504–1508.

[Lin et al., 2018] Lin, Y., Wang, C., Zhai, D., Li, W., and Li, J. (2018). Toward better boundary preserved supervoxel segmentation for 3d point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

[Mckinney, 2010] Mckinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*.

[Melzer, 2007] Melzer, T. (2007). Non-parametric segmentation of als point clouds using mean shift. *Journal of Applied Geodesy*, 1:159–170.

[Nguyen and Le, 2013] Nguyen, A. and Le, B. (2013). 3d point cloud segmentation: A survey. In *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, pages 225–230.

[Otepka et al., 2013] Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., and Pfeifer, N. (2013). Georeferenced point clouds: A survey of features and point cloud management. *ISPRS Int. J. Geo-Information*, 2:1038–1065.

[Otepka et al., 2012] Otepka, J., Mandlburger, G., and Karel, W. (2012). The opals data manager - efficient data management for processing large airborne laser scanning projects. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:153–159.

[Papon et al., 2013] Papon, J., Abramov, A., Schoeler, M., and Wörgötter, F. (2013). Voxel cloud connectivity segmentation - supervoxels for point clouds. pages 2027–2034.

[Peng et al., 2013] Peng, B., Zhang, L., and Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3):1020 – 1038.

[Pfeifer and Briese, 2007] Pfeifer, N. and Briese, C. (2007). Laser scanning – principles and applications.

[Pfeifer et al., 2014] Pfeifer, N., Mandlburger, G., Otepka, J., and Karel, W. (2014). Opals – a framework for airborne laser scanning data analysis. *Computers, Environment and Urban Systems*, 45:125 – 136.

[Poechtrager, 2016] Poechtrager, M. (2016). Segmentierung großer punktwolken mittels region growing. Master's thesis, Technische Universitaet Wien.

[Pu et al., 2011] Pu, S., Rutzinger, M., Vosselman, G., and Elberink, S. O. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6, Supplement):S28 – S39. Advances in LIDAR Data Processing and Applications.

[Rabbani et al., 2006] Rabbani, T., van den Heuvel, F., and Vosselman, G. (2006). Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36.

[Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152.

[Rusu et al., 2009] Rusu, R., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. pages 3212 – 3217.

[Rusu and Cousins, 2011] Rusu, R. B. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4.

[Rutzinger et al., 2008] Rutzinger, M., Höfle, B., Hollaus, M., and Pfeifer, N. (2008). Object-based point cloud analysis of full-waveform airborne laser scanning data for urban vegetation classification. *Sensors*, 8.

[Schnabel et al., 2007] Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

[Schoenberg et al., 2010] Schoenberg, J. R., Nathan, A., and Campbell, M. E. (2010). Segmentation of dense range information in complex urban scenes. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2033–2038.

[Schonberger and Frahm, 2016] Schonberger, J. L. and Frahm, J. (2016). Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, Los Alamitos, CA, USA. IEEE Computer Society.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905.

[Silberman et al., 2012] Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. pages 746–760.

[Sima and Nuchter, 2013] Sima, M.-C. and Nuchter, A. (2013). An extension of the felzenszwalb-huttenlocher segmentation to 3d point clouds. *Proc SPIE*, pages 02–.

[Strom et al., 2010] Strom, J., Richardson, A., and Olson, E. (2010). Graph-based segmentation for colored 3d laser point clouds. pages 2131 – 2136.

[van der Walt et al., 2014] van der Walt, S., Schönberger, J., Nunez-Iglesias, J., Boulogne, F., Warner, J., Yager, N., Gouillart, E., Yu, T., and scikit-image contributors, t. (2014). scikit-image: Image processing in python. *PeerJ*, 2.

[Vauhkonen et al., 2014] Vauhkonen, J., Maltamo, M., Mcroberts, R., and Næsset, E. (2014). *Introduction to Forestry Applications of Airborne Laser Scanning*, volume 27, pages 1–16.

[Vosselman, 2013] Vosselman, G. (2013). Point cloud segmentation for urban scene classification. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-7/W2:257–262.

[Walt et al., 2011] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.

[Wu and M. Leahy, 1993] Wu, Z. and M. Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:1101–1113.

[Xiao et al., 2019] Xiao, W., Zaforemska, A., Smigaj, M., Wang, Y., and Gaulton, R. (2019). Mean shift segmentation assessment for individual forest tree delineation from airborne lidar data. *Remote Sensing*, 11(11).

[Yang and Dong, 2013] Yang, B. and Dong, Z. (2013). A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 81:19 – 30.

[Yang and Fang, 2014] Yang, B. and Fang, L. (2014). Automated extraction of 3-d railway tracks from mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4750–4761.

[Yao et al., 2009] Yao, W., Hinz, S., and Stilla, U. (2009). Object extraction based on 3d-segmentation of lidar data by combining mean shift with normalized cuts: Two examples from urban areas. In *2009 Joint Urban Remote Sensing Event*, pages 1–6.