3D mathematical morphology on voxelized point clouds for outlier detection

Simon Griffioen
S.J.Griffioen@student.tudelft.nl

Mentors:
Ravi Peters
Hugo Ledoux
Maarten Pronk (Deltares)

January 14, 2017

Contents

1 Introduction

Laser scanners help acquiring the 3D geometry of real-world objects in an efficient and simple way. To obtain 3D information of the Earth's surface, airborne Light Detection And Ranging (LiDAR) technology is used to quickly capture high-precision measurements of the terrain. The generated raw point cloud reflects the characteristics of a Digital Surface Model (DSM) of the measured region. Unfortunately, laser scanning techniques are prone to producing outliers and noise (i.e. wrong measurements), due to limitations of the sensor, illumination situations and undesirable artifacts in the scene (Jenk et al., 2006; Han et al., 2017). This causes problems for further processing steps, e.g. extracting a Digital Elevation Model (DEM) or object recognition. Therefore, a pre-process of the point cloud is required to detect and remove spurious measurements.

In 1997 a nationwide DEM of the Netherlands was created with up to one height point per 16 m^2, using airbore LiDAR (ALS) (Lemmens, 2011). In the meantime LiDAR has evolved and is recognized as a technology with many advantages and application purposes. LiDAR features high accuracy and precision with a high level of detail. Point clouds can easily contain up to millions of points per square kilometer (Lemmens, 2011). High levels of automation and quick data capturing make it very convenient for detailed 3D-reconstruction of the real world. However, large data sets with millions of height measurements make manual processing almost impossible. Automatic approaches are necessary due to larger amounts of data, but efficient and effective methods are needed (Papadimitriou et al., 2002), i.e. large computations on point cloud data need to be efficient.

Multiples reflections on surfaces such as metal or window glasses cause fault distance measurements and result in outlying data points. Besides affecting the visual appearance of the dataset, these points cause geometric differences and have a negative effect on further processing steps. An important pre-processing step is to detect and remove such outliers. While outlier detection in datasets has been extensively researched, in 3D point cloud data it is still an ongoing problem. Several methods for detecting outliers in LiDAR data are proposed and studied, and are described in Section 2. Typical methods are designed and developed in commercial software, e.g. TerraScan and LAStools. Free alternatives exist as well in the form of open source projects like Point Cloud Library (PCL). Often a density-based approach is offered, which depends on the local density of its neighborhood. The neighborhood is defined by a sphere (TerraScan, PCL) or a 3x3x3 grid (LAStools). Usually this performs well for detecting isolated points (i.e. outliers), but are incapable of removing outliers in a format of cluster (Shen et al., 2011). Furthermore, Arge et al. (2010) developed a cleaning algorithm which is incorporated in a commercial product. These developments show large interest in automatic outlier detection from the industry.

This thesis will be in collaboration with Deltares, a technological institute for water, soil and infrastructure. Many companies, in e.g. forestry, water management, civil engineering, benefit from accurate 3D models of the real world. Terrain models provide valuable information, and a DEM is especially important for water management. Airborne LiDAR (ALS) is often used for creating such DEMs. Deltares flew airplanes to collect LiDAR data from natural environments to understand the landscape. Large data acquisitions like this

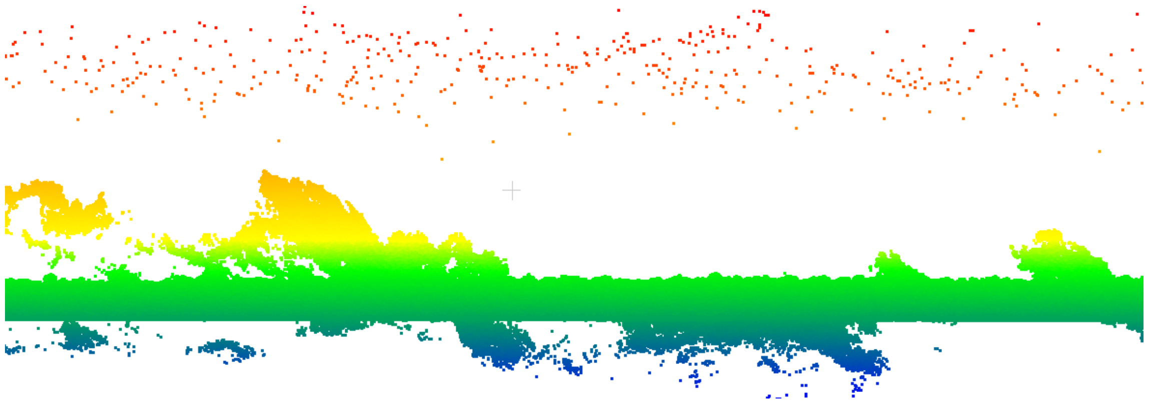


Figure 1: Point cloud with high outliers (red); low outliers (blue)

are costly and can only be done once. However, due to limitations in the acquisition method, outliers appear in the resulting data sets. These outliers can cause inaccuracy in the creation of a DEM. For instance, outliers below the terrain, due to multi path (see Section 1.1), will be seen as ground points and changes the height of its local neighborhood.

1.1 Sources of outliers in ALS

Outliers are basically erroneous measurements caused by the limitations of the laser scanner. Some well knows reasons are described in this section. The first is caused by the footprint of a laser beam, which is an ellipse. It is possible that the beam is divided when it hits the boundary of an object. Thus the reflection value of this point would be a weighted average of the reflection from both surfaces, creating virtual points in between (El-Hakim and Beraldin, 1994). A second reason can be caused by surfaces with very high or low reflectance. Such surfaces, such as black objects, glasses or metals can cause bias in the distance measurement, because the receiver cannot resolve the reflected beam (Beraldin, 2004). The third reason is multi path reflection. When the laser beam hits a surface under a angle, most of the beam can be deflected onto other close surfaces and then reflected back into the receiver. This creates a false longer distance and thus a wrong data point (Sotoodeh, 2006).

The mentioned limitations of LiDAR cause different types of outliers. Many studies distinguish two types, high and low outliers, I will add a third type: cluster outliers (see Fig. 1)

- *Low outliers* - These are points that normally do not belong to the surface. They come from multi-path errors and errors in the laser scanner. Normally, filters work on the assumption that the lowest point belongs to the ground.
- *High outliers* - These are points that also normally do not belong to the surface. They originate from hits off objects like birds, low flying aircraft, or errors in the laser scanner. Most filters handle such outliers, because they float high in the air far from neighboring points.

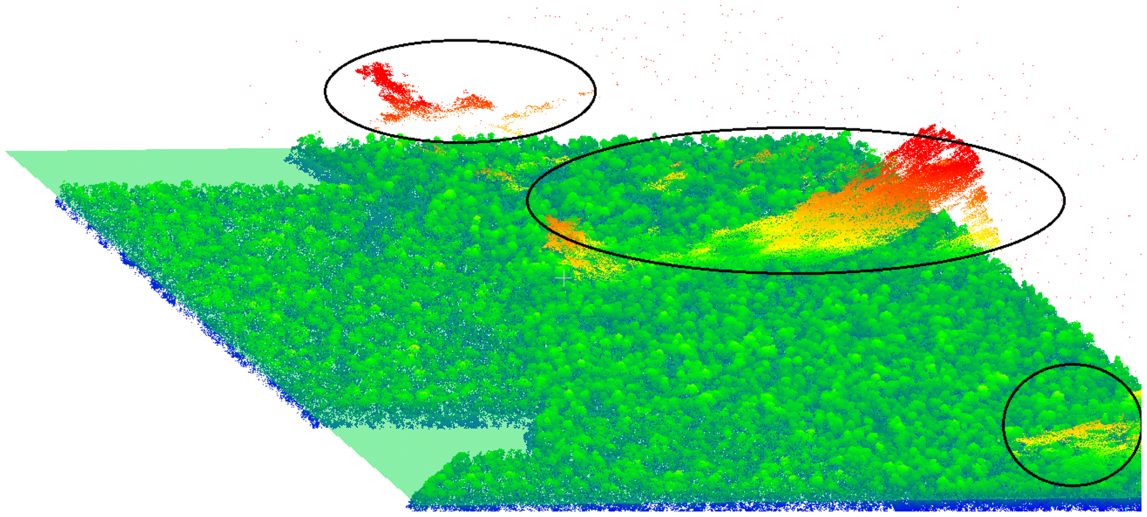


Figure 2: Point cloud of swamp forest in Indonesia with outlying clusters; provided by Deltares

- *Outlier clusters* - These points are similar to high and low outliers, but form a cluster. This makes them harder to detect, because they are not isolated individual points. This asks for different detection methods, which many existing tools do not offer. They can be caused by a flock of birds, but also exist due to limitations of the sensor. An example is seen in Fig. 2, noise is created in an arch shaped cluster.

2 Related work

Most work on outlier detection is done in the field of statistics (Barnett, 1994; Hawkins, 1980). Algorithms in machine learning and data mining have considered outliers, but mostly in a way to deal with them in order to successfully run an algorithm (Angluin and Laird, 1988). Hawkins (1980) defines an outlier as an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism. Many studies propose methods to find these data points, but most of them suffer from two problems. First, they are univariate, i.e. they only consider one variable. Point cloud data is multidimensional (at least x, y, z), making it unsuitable. Secondly, in all cases you have to perform expensive testing to find a distribution that fits the data (Johnson et al., 1998). This makes outlier detection in point cloud data different than what we are used to from statistical methods. Outliers are caused from different sources and are seen as measurements that differ from its local neighborhood, i.e. data points that do not fit the local surface geometry (Sotoodeh, 2006). Different approaches are studied to identify these outlying points, the next section gives an overview.

2.1 Overview point cloud outlier detection methods

Point cloud filtering has been an on-going research problem in several fields, including computational statistics, computer graphics and more. As a result many filtering methods have been proposed. Here, only studies associated with large scale point cloud filtering are given attention. Methods for airborne LiDAR (ALS) captured data sets are given highest priority, but Terrestrial Laser Scanning (TLS) and Mobile Laser Scanning (MLS) datasets are studied as well. One major difference between these capturing methods, is that ALS results in a point cloud with an equal density of points, whereas ALS and MLS can have different local densities. This effects density-based approaches.

Papadimitriou et al. (2002) classifies five different categories of filters as distribution-based, depth-based, distance-based, density-based and clustering approaches. Distribution-based methods deploy some standard distribution model (e.g. Normal distribution) and classifies outliers those points that deviate from the model. However, for arbitrary datasets without prior knowledge of the distribution of points (e.g. point cloud datasets), finding the which model fits the data best. Local surface fitting approaches, for instance moving least squares or RANSAC, are also used for outlier detection. These methods can work well when a point cloud is dense and is obtained from a smooth surface, e.g. urban environments. Point clouds with discontinuities or high curvature areas, for example natural environments, are not suited for surface fitting (Sotoodeh, 2007).

The depth-based approach is based on computational geometry and computes different layers of *k*-dimensional convex hulls (Johnson et al., 1998). Objects in the outer layer are identified as outliers. In these approaches, based on some definition of depth, data objects are organized in layers in the data space, with the expectation that objects in the outer layer are more likely to contain outlying points. To avoid the above mentioned problems of distribution fitting and restriction to univariate datasets, depth-based approaches have been developed (Johnson et al., 1998). However, algorithms for this method cannot cope with large three-dimensional datasets (Sotoodeh, 2007).

Clustering algorithms are not optimized for outlier detection, since the main objective is clustering. However, they can be used for this, but outliers are by-products, i.e. don't belong to a cluster (Jain et al., 1999). Teutsch et al. (2011) clusters point cloud data to identify sets of neighboring points that belong to the same region. Most of the resulting regions contained less than 10 points. They considered regions with less than 1000 points outliers, and removed all regions below this threshold. Sithole (2005) segments ALS data to separate terrain, house roofs, bridges and trees. Points that don't belong to a predefined class are labeled as outliers.

Knorr et al. (2000) originally proposed the distance-based approach. An point in a data set P is a distance-based outlier if at least a fraction β of the points in the local neighborhood is further than r from it. This outlier defintion is determined by two parameters, r defines the distance and β defines a threshold for number of data points. Many existing point cloud filters are based on this approach. TerraScan, PCL and GCAL implemented filters which removes all points in its input cloud that don't have at least some number of neighbors within a certain range. It performs well on 3D point cloud data to detect isolated points. However, this can lead to problems when the data set has both dense and sparse regions (Breunig et al., 2000).

The density-based approach is proposed in Breunig et al. (2000). Outliers can be seen as isolated points, which depends on the local density of its neighborhood. The neighborhood is defined by the distance to n nearest neighbors. Value n is a predefined value and determines the size of a local neighborhood, used to calculate the density. This method works without prior knowledge of the distribution of points and handles different local point densities (Breunig et al., 2000). Sotoodeh (2006) adopted the algorithm for outlier detection in laser point clouds. The algorithm does not detect outlier clusters with the point density higher than n points (where n is a predefined threshold). The detection of single outliers and even cluster outliers with lower density than n seems satisfactory. While the algorithm detects a part of outliers, the problem of the detection of the cluster outliers is still a challenge.

The distance- and density-based approach attract more attention for outlier detection, because they are more appropriate for high dimensional, large data sets (Papadimitriou et al., 2002). This is also reflected in the available point cloud outlier detection methods on the market, e.g. TerraScan, LAStools and PCL have routines based on distance and/or density of neighboring points.

2.2 Dealing with large datasets

Not all methods can be categorized in one class. Some studies propose methods combining different approaches. Tian et al. (2012) combines distance-based method and density-based method, and it is effective for isolated outliers and clustered outliers in airborne LiDAR point clouds. Outlier detection is based on a kernel density estimation. This is used to identify low and high points. To deal with altering terrain heights, the dataset is segmented into sections. This is done by dividing the point cloud into many blocks in the 2D horizontal plane. For every separate block the density probability distribution is estimated. Then a density threshold is defined to eliminate outliers. Shen et al. (2011) detects outliers by calculating the mean distance for every point and its k-nearest neighbors. If the average distance is larger than an adaptively preset value, the point is regarded as an outlier. Finding nearest neighbors is a heavy computation for a large data set, such as a point cloud. Therefore, they construct a kd-tree of an airborne LiDAR point cloud data set to efficiently find the k-nearest neighbors. This method is also able to find clusters of outliers by choosing a k larger than the cluster. Large computational cost is often a problem when analyzing 3D data. Many processing steps involve the local neighborhood for every point in the point cloud.

Vo et al. (2015) and Plaza et al. (2015) construct a 3D grid of a point cloud, i.e. voxelization overlays the input point cloud. Voxelization aim to solve the time-consumption problem of the local neighborhood techniques by reducing the scene to a voxel model. Features are extracted from points inside a voxel instead of finding k-nearest neighbors for each point. The voxelization of a point cloud starts by determining the minimum bounding box. A specified edge length for each voxel divides the minimum bounding box in a 3D grid of cells. Every voxel with points inside are stored in the voxel data structure, i.e. voxels without points inside are disregarded. This structure has several advantages, the number of voxels are significant less than the number of points, and the voxel structure can be used to find neighboring voxels easily. Vo et al. (2015) overlays the voxels with an octree structure. After the initial partition of voxels, non-empty voxels continue to be divided, until either the minimum voxel size or residual is satisfied. These studies showed that employing a voxel model (1) simplifies the initial data; (2) indexes the data, and (3) defines neighborhood groups, and thereby avoids expensive searches for neighboring points (Vo et al., 2015). Furthermore, by translating (x, y, z)-points data into a 2D regular grid, image processing techniques can be used. In remote sensing, satellite imagery are considered to be raster data and image processing techniques are used, e.g. for object detection. In Section 2.2.1 I will explain why this is important for this thesis.

2.2.1 Raster Processing with Morphological Operators

Mathematical morphology is a well known technique for the analysis and processing of digital images (Soille, 2003). It was originally developed for binary images, but was later extended to grayscale images, and other geometrical structures, such as graphs and meshes. The basic operations in mathematical morphology are *erosion* and *dilation*, which are performed over a neighborhood specified by a structural element (or kernel). Usually, the kernel is a (n x n)-window that is shifted over the image and at each pixel of the image, the kernel is compared with the set of the underlying pixels. Based on erosion and dilation, two other operations, *opening* and *closing*, can be derived. Opening is just another name of erosion followed by dilation, and closing is reverse of opening (i.e. dilation followed by erosion). A nice property of opening is that it's useful for removing noise.

Mathematical morphology is also used for filtering LiDAR data. Chen et al. (2007) proposes a method for filtering airborne LiDAR. The idea of this morphological method is approximating the terrain surface using morphological operations. The basic idea is based on the observation that a large height difference between two nearby cells is unlikely to be caused by a steep slope in the terrain The first step is to create a 2D raster of a point cloud by taking the elevation value of the last return of every pulse in a 1m x 1m cell. Then the morphological opening can approximate the bare earth, i.e. trees and vegetation are removed. Large elevation points can be corrected to the lower elevation value of other cells inside the kernel. The same method also removed high outliers. Kilian et al. (1996) use multiple structure elements of different size in the morphological operation *opening*. This method works on a rasterized dataset, whereas other methods work on the point cloud directly. Vosselman (2000) works with the original, irregularly distributed point data, even though it is computationally more expensive.

Gorte and Pfeifer (2004) designed a method based on 2D mathematical morphology raster processing, and transferred it into the 3D domain. Connectivity and neighborhood relations between voxels in a 3D raster can be established much easier than between the original (x, y, z)-points, using morphological operations. Closing and opening are applied to close gaps and holes, and remove isolated points.

2.3 Available point cloud outlier detection software

Outlier detection and removal software is already available in different formats. Many software designed for LiDAR data analysis have a tool for removing outliers. This section mentions some well known tools.

- *LAStools* is a software package with several different functions for LiDAR data, e.g. classify, convert, tile and remove noise. This specific noise detection function is called lasnoise. This tool flags or removes outliers from LAZ/LAS files. The function searches for isolated points and can be categorized as density-based approach. The tool tries to find points, that have only a few points in their surrounding 3 by 3 by 3 grid of cells. Cell size and maximum number of points within the cells must be determined by the user.

- *Point Cloud Library (PCL)* is open source project for point cloud processing. It includes two outlier removal tools categorized as distance-based approach. The first computes the mean distance to its nearest neighbors. All points whose mean distances are outside an interval defined by the global distances mean and standard deviation, are considered outliers. The second method counts the number of neighbors within a specified radius for every point. Points that don't have enough neighbors in its radius are flagged as outliers.

- *TerraScan* is software for LiDAR Data Processing by TerraSolid. It has a similar tool for outlier detection as PCL. It searches for isolated points, by counting the number of neighbors in a specified radius. It also has a tool for detecting low points below the ground level due to double reflection. TerraScan is often used in practice. For example, AHN3 is removed from outliers using this product.

The available tools are able to detect outliers in an efficient way. However, they all share one major issue: outlying clusters of points are not identified. They also require a lot of parameter adjustment by the user to gain maximum result.

3 Research questions

3.1 Problem statement

Airborne LiDAR is prone to producing outliers in point cloud data. A pre-processing step is required to remove such outliers. Existing software tools mainly use the density-based or distance-based approach to detect and remove outliers (see Subsection 2.3). They succeed for the obvious high-, low- and isolated points, but fail to detect outlying clusters (Subsection 1.1). This requires more complex computations and cannot be solved within density- and distance-based approaches (Subsection 2.1). For efficient data processing, many studies propose a voxel overlay structure of the point cloud (Subsection 2.2). Besides simplifying the data, translating it into a raster creates opportunities to use existing image processing techniques (Subsection 2.2.1). Mathematical morphology is used for filtering point clouds, and several studies show methods for removing noise (Subsection 2.2.1). However, the main focus is ground estimation, and not outliers detection. Moreover, again only isolated points are detected. The full possibilities of raster processing has not been studied yet for the identification of massive airborne LiDAR point clouds. Raster processing can be used in 2D and 3D to analyze a voxelized point cloud, e.g. connectivity, shape detection, and noise removal.

3.2 Research question

As argued in the introduction and the related work section the main goal of this thesis is to automatically identify outliers (i.e. isolated points, high and low, and outlying clusters) in large scale airborne LiDAR point clouds. The corresponding main research question is therefore:

- *How can morphological operations be used on voxelized airborne LiDAR point cloud, from natural environments, to detect outliers?*

Subquestions for this research are

- *What morphological operators can be used to detect outliers in a 2D raster, and how can this be translated into 3D?*
- *What is the effect of different input parameters (e.g. voxel size, kernel shape) for outlier detection algorithms?; And can these parameters be tuned automatically?*
- *What is the influence of scaling the dataset on processing time?*
- *How to minimize processing time on massive data sets?*

3.3 Research Scope

To set the scope for this research it is important to mention that this study will use, and test algorithms on, airborne LiDAR data from natural environments. Other laser scanning techniques and urban environments will be disregarded. Furthermore, the main focus of this study will be on morphological operators in a 3D raster, because this is where the most contribution can be made. Studying different existing operators with different kernels (i.e. shape/size) to detect different types of outliers is a novel approach.

Furthermore the research aims to develop an efficient method since it should be applicable on massive data sets. It is beyond the scope of this study to benchmark different methods, but the search for an efficient approach will be discussed in a theoretical way. However, results from testing several approaches can be discussed, including the effect of increasing the size of the dataset.

4 Methodology

Outlier detection is done in several steps. First the point cloud will be converted into a 3D voxel structure. After converting the point cloud to a 3D raster, erosion, dilation and opening are applied to filter noise. Opening will remove isolated points, with erosion and

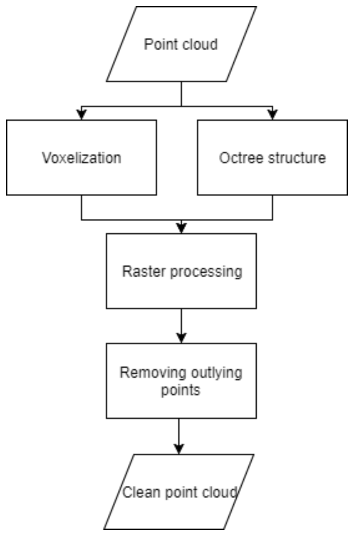


Figure 3: Schematic overview methodology

dilation points can be connected or unconnected. Clusters of noise can be removed by investigating the connectivity. When defining connectivity in 3D, a choice has to be made between 6, 18 and 26-adjacency, depending on the number of voxels that are considered neighbors of a voxel: only those with a common face, also those with a common edge, or even also the ones with a common corner. The following sections will describe the methodology stepwise (Fig. 3).

4.1 Voxelization

Analysis is performed in the 3D raster domain where every cell can be called a *voxel*. In a point cloud, (x, y, z)-coordinates are stored explicitly, in addition topology should be stored explicitly as well. In a 3D raster voxel locations are defined implicitly by the position in the grid. However, a grid also creates voxels on locations where no data points are located. Therefore, it can be decided to not store voxels without points inside. In turn this requires voxels to be stored explicitly. A solution for this problem is the octree data structure, in Subsection 4.3 this will be described.

First the methods will be developed for a regular voxel grid. The voxel value will be limited to *0* for empty voxels and *1* for voxels with laser points inside. The most parameter in this conversion is *spatial resolution*, i.e. the size of one voxel. This also determines the resolution of the 3D raster used for raster processing and how many voxels will be created. The number of voxels in a 3D raster increases with the 3rd power of the resolution (when expressed in voxels/distance) and may easily become impracticable when resolution is chosen too fine. Memory capacity problems may occur in software that needs random access within the 3D space (Gorte and Pfeifer, 2004).

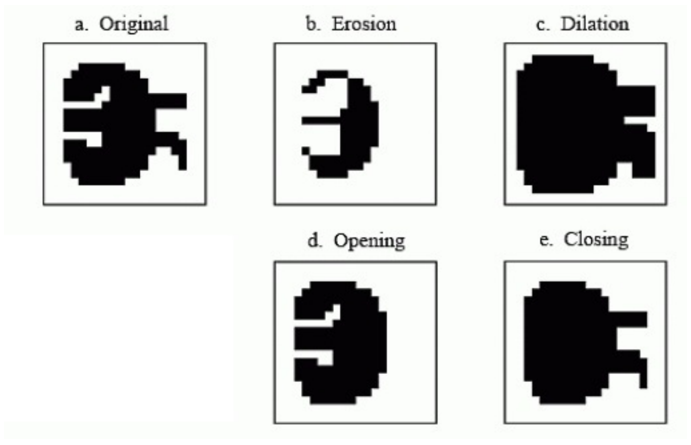


Figure 4: Morphological operations. Four basic operations are used on a example binary image.

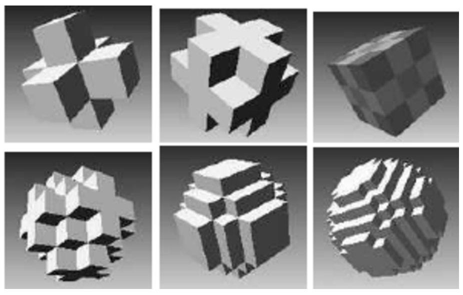


Figure 5: 3D filter kernels

4.2 Morphological Operations

After transferring the point cloud to a 3D raster, neighborhood operators are applied to analyze the data. The neighborhood operators are 3D extensions of mathematical morphological operators, such as erosion, dilation and opening (Fig. 4). Opening operator is used for noise reduction, i.e. isolated points are removed. A bigger kernel size can also remove small clusters of noise. Different kernels must be studied and tested for detecting noise, Fig. 5 shows examples of different kernels and structuring elements.

4.3 Octree

To overcome the problem of storing many empty voxels and memory capacity problems may occur, an octree for space decomposition can be used. In an octree structure all nodes are split into eight identical children, resulting in all voxels at a single level being the same size. Only full octree nodes are subdivided, until a minimum voxel size is reached. Lin and Wong (1996) describes a method for mathematical morphology direct on quadtree images. This can be extended in a 3D octree method. This will be studied after the previous two steps showed promising results.

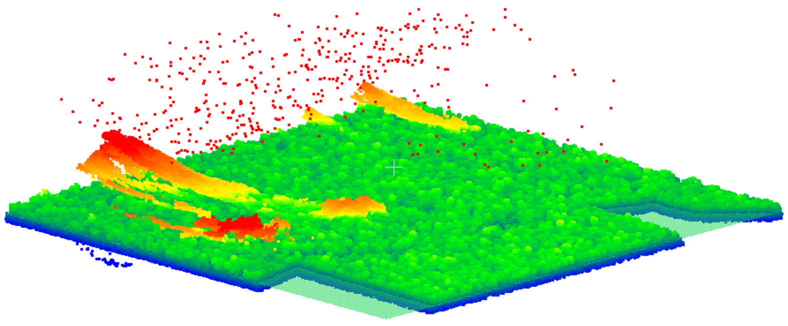


Figure 6: Input point cloud with outliers; provided by Deltares

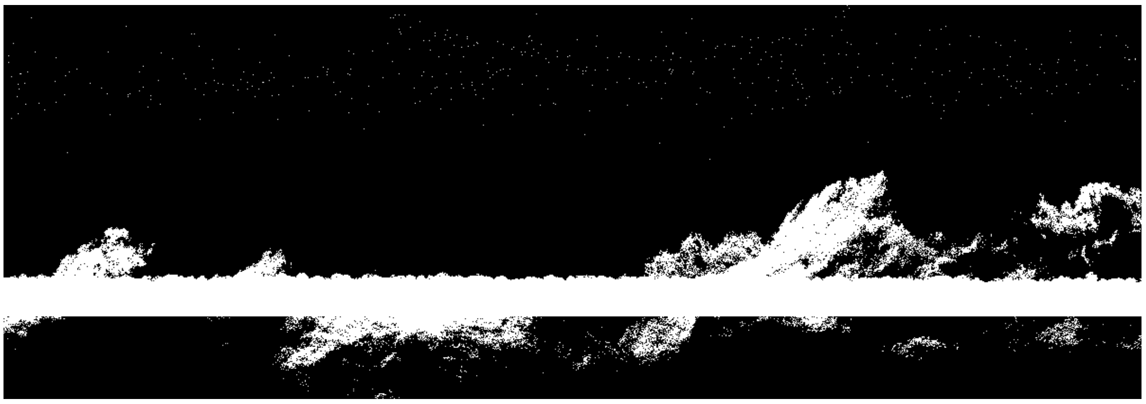


Figure 7: Vertical section of rasterized point cloud.

5 Preliminary Results

This section will show some very preliminary results. Instead of working with a 3D raster, image processing is used on a 2D rasterized point cloud. This already shows some potential in the method and is able to remove isolated outliers. In Fig. 6 the point cloud is shown that is used. It has high and low outliers and cluster outliers that intersect the landscape in a unfortunate way, making it hard to detect and remove.

From the input point cloud a vertical section is taken and converted in a binary raster. Value *1* for cells with laser points inside and *0* for cells without laser points inside. A spatial resolution of 1 meter is used (Fig. 7).

The raster image is then processed with erosion (Fig. 8a), dilation (Fig. 8b) and opening (Fig. 9). The kernel has a resolution of 7x7 pixels. The result shows that all high outliers are removed and many low outliers. From the outlying cluster many points are removed as well. The kernel size influences this a lot, whereas a larger window removes more, but could also remove 'good' points. Also, multiple iterations can be investigated.

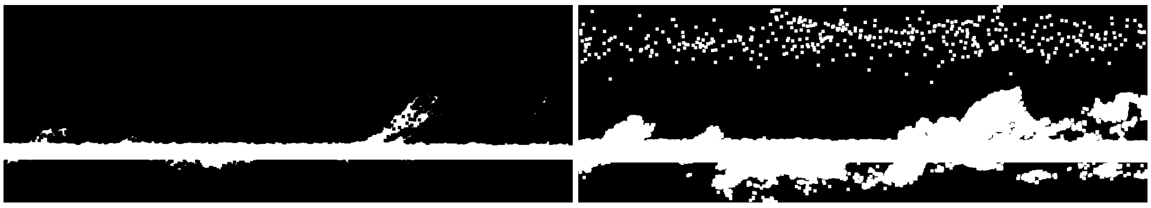

(a) Erosion. (b) Dilation.

Figure 8: Morphological operators erosion and dilation of raster image.

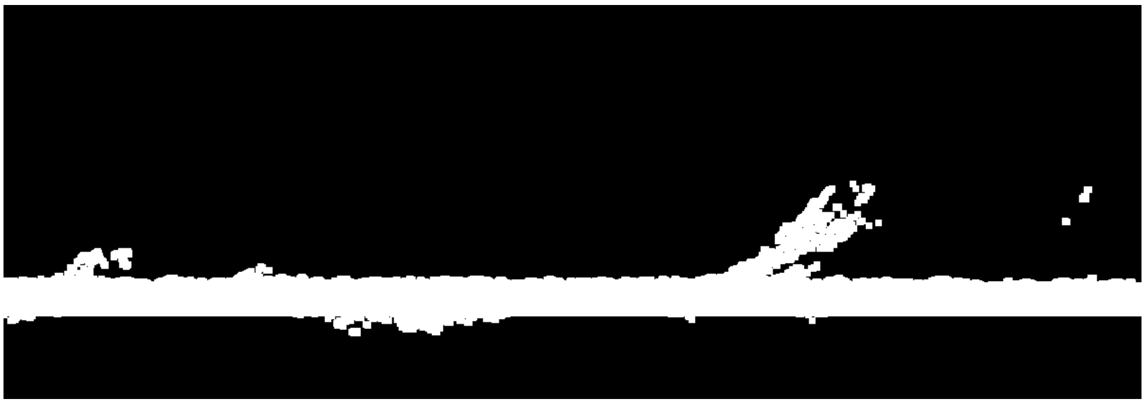
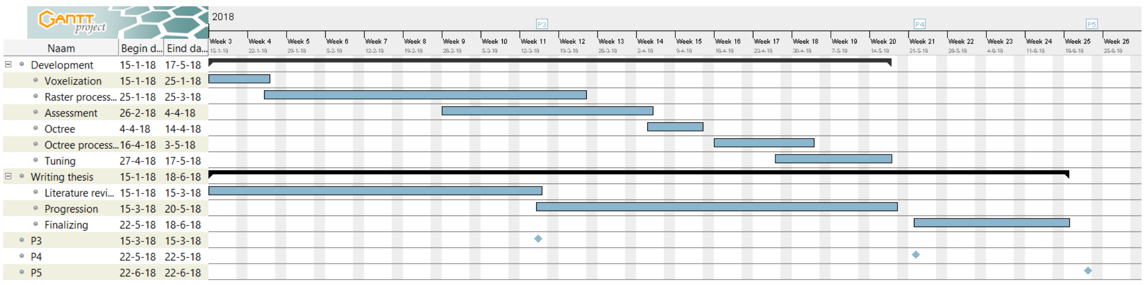


Figure 9: Erosion followed with dilation is the opening operator. High outliers are all removed and much of the low outliers as well. Even large part of the cluster noise is removed.

6 Time planning

In Fig. 10 the planning is shown in a GANTT chart. It is priority to develop a working algorithm using mathematical morphology for removing outliers. After a successful work flow, an octree based implementation can be investigated.



Naam	Begin d...	Eind da...
Development	15-1-18	17-5-18
Voxelization	15-1-18	25-1-18
Raster process...	25-1-18	25-3-18
Assessment	26-2-18	4-4-18
Octree	4-4-18	14-4-18
Octree process...	16-4-18	3-5-18
Tuning	27-4-18	17-5-18
Writing thesis	15-1-18	18-6-18
Literature revi...	15-1-18	15-3-18
Progression	15-3-18	20-5-18
Finalizing	22-5-18	18-6-18
P3	15-3-18	15-3-18
P4	22-5-18	22-5-18
P5	22-6-18	22-6-18

Figure 10: GANTT chart of thesis planning.

7 Tools and datasets used

The most important mentioning in this section is which data set will be used for the study. Deltares has many tiled point cloud data from natural environments containing millions of points. In Fig. 6 one example set is already shown. This data set contains outliers of every type and is very challenging to clean the complete data set. This makes it a good point cloud for testing. For accuracy assessment this dataset needs to be manually classified to compare the results with the ground truth.

Processing of the data will be done with *Julia* coding language. This is a novel language trying to achieve the best of both *C+* and *Python*. Dealing with large data sets, *Julia* provides a good alternative for developing filtering methods.

For image processing, OpenCV is used. This is an Open Source Computer Vision library. Julia has a wrapper built for it.

Furthermore other software will be used for dealing with LAZ/LAS file. LAStools for converting, compressing etc., and CloudCompare for visualization.

References

Angluin, D. and Laird, P. (1988). *Machine Learning*, 2(4):343–370.

Arge, L., Green Larsen, K., Mølhave, T., and van Walderveen, F. (2010). Cleaning massive sonar point clouds. In *Proceedings 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 152–161. ACM.

Barnett, T. L. (1994). *Outliers in Statistical Data*. JOHN WILEY & SONS INC.

Beraldin, J.-A. (2004). Integration of laser scanning and close-range photogrammetry – the last decade and beyond. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science Congresss*, pages 972–983.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. ACM Press.

Chen, Q., Gong, P., Baldocchi, D., and Xie, G. (2007). Filtering airborne laser scanning data with morphological methods. *Photogrammetric Engineering & Remote Sensing*, 73(2):175–185.

El-Hakim, S. F. and Beraldin, J. A. (1994). Integration of range and intensity data to improve vision-based three-dimensional measurements. In El-Hakim, S. F., editor, *Videometrics III*. SPIE.

Gorte, B. and Pfeifer, N. (2004). Structuring laser-scanned trees using 3d mathematical morphology. In *International Archives of Photogrammetry and Remote Sensing*, volume 35, pages 929–933.

Han, X.-F., Jin, J. S., Wang, M.-J., and Jiang, W. (2017). Guided 3d point cloud filtering. *Multimedia Tools and Applications*.

Hawkins, D. (1980). *Identification of Outliers*. Springer Netherlands.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

Jenk, P., Wand, M., Bokeloh, M., Schilling, A., and StraSer (2006). Bayesian point cloud reconstruction. *Comput Graph Forum*, 25(3):379–388.

Johnson, T., Kwok, I., and Ng, R. T. (1998). Fast computation of 2-dimensional depth contours. In *4th International Conference on Knowledge Discovery and Data Mining*, pages 224–228.

Kilian, J., Haala, N., and Englich, M. (1996). Capture andevaluation of airborne laser scanner data. In *International Archives of Photogrammetry and Remote Sensing*, pages 383–388.

Knorr, E. M., Ng, R. T., and Tucakov, V. (2000). Distance-based outliers: algorithms and applications. *The VLDB Journal The International Journal on Very Large Data Bases*, 8(3-4):237–253.

Lemmens, M. (2011). *Geo-information*. Springer Netherlands.

Lin, R. and Wong, E. (1996). Morphological operations on images represented by quadtrees. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. IEEE.

Papadimitriou, S., Kitagawa, H., Gibbons, P., and Faloutsos, C. (2002). LOCI: fast outlier detection using the local correlation integral. In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*. IEEE.

Plaza, V., Gomez-Ruiz, J. A., Mandow, A., and Garcia-Cerezo, A. (2015). Multi-layer perceptrons for voxel-based classification of point clouds from natural environments. In *Advances in Computational Intelligence*, pages 250–261. Springer International Publishing.

Shen, J., Liu, J., Zhao, R., and Lin, X. (2011). A kd-tree-based outlier detection method for airborne LiDAR point clouds. In *2011 International Symposium on Image and Data Fusion*. IEEE.

Sithole, G. (2005). *Segmentation and Classification of Airborne Laser Scanner Data*. PhD thesis, TU Delft, Publications on Geodesy, 59. Publication of Netherlands Geodetic Commision.

Soille, P. (2003). *Morphological Image Analysis: Principles and Applications*. Springer Berlin Heidelberg.

Sotoodeh, S. (2006). Outlier detection in laser scanner point clouds. In *The international archives of the photogrammetry, remote sensing and spatial information sciences*, volume 36, pages 297–302. ISPRS.

Sotoodeh, S. (2007). Hierarchical clustered outlier detection in laser scanner point clouds. In *In: Laser07*, page 383.

Teutsch, C., Trostmann, E., and Berndt, D. (2011). A parallel point cloud clustering algorithm for subset segmentation and outlier detection. In Remondino, F. and Shortis, M. R., editors, *Videometrics, Range Imaging, and Applications XI*. SPIE.

Tian, X., Xu, L., Li, X., Jing, L., and Zhao, Y. (2012). A kernel-density-estimation-based outlier detection for airborne LiDAR point clouds. In *2012 IEEE International Conference on Imaging Systems and Techniques Proceedings*. IEEE.

Vo, A.-V., Truong-Hong, L., Laefer, D. F., and Bertolotto, M. (2015). Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104:88–100.

Vosselman, G. (2000). Slope based filtering of laser altimetry data. *International Archives of Photogrammetry and Remote Sensing*, 33(B3/2):935–942.