

# Comparison of Algorithms for Construction Detection using Airborne Laser Scanning and nDSM Classification

Sergejs Kodors  
Rezekne Academy of Technologies  
Rezekne, Latvia  
sergejs.kodors@rta.lv

**Abstract** — Traditional approach to classify the point cloud of airborne laser scanning is based on the processing of a normalized digital surface model (nDSM), when ground facilities are detected and classified. The main feature to detect a ground facility is height difference between adjacent points. The simplest method to extract a ground facility is region-growing algorithm, which applies threshold to identify the connection between two points. Region growing algorithm is working with the constant value of height difference. Therefore, it is not applicable due to diverse conditions of earth surface, when height difference must be defined for each region separately. As result, researchers propose hierarchical, statistical and cluster methods to solve this problem. The study goal is to compare four algorithms to generate nDSM: region growing, progressive morphological filter, adaptive TIN surfaces and graph-cut. The experiment is divided into two stages: 1) to calculate the number of detected and lost buildings in nDSM; 2) to measure the classification accuracy of extracted shapes. The experiment results have showed that progressive morphological filter and graph-cut provides the minimal loss of buildings (only 1%). The most effective algorithm for ground facility detection is the graph-cut (total accuracy 0.95, Cohen's Kappa 0.89,  $F_1$  score 0.93).

**Keywords** — buildings, LiDAR, nDSM, remote sensing

## I. INTRODUCTION

The application of a normalized digital surface model (nDSM) is common approach to detect ground facilities in 3D point cloud obtained by airborne laser scanning (ALS). The generation of nDSM is related with the construction of a digital elevation model (DEM), when the LiDAR points of ground facilities are extracted from all dataset (see Eq.1):

$$nDSM = DSM - DEM, \quad (1)$$

where  $DSM$  is a digital surface model or all points of dataset,  $DEM$  – the points, which belong to a ground, and  $nDSM$  – the points of surface facilities.

The construction of DEM or nDSM is actually the filtering of DSM, when the extra points are removed out from it (see Eq.2-3):

$$DEM = DSM - f(DSM). \quad (2)$$

$$nDSM = DSM - f(DSM). \quad (3)$$

Usually the filtering function  $f(DSM)$  is based on the height difference between adjacent points. For example, the region growing algorithm segments whole image into regions, where the pixels have difference of attributes smaller than threshold. If the processed area is sufficiently large, the largest segment should belong to the category “ground”, that can be applied to obtain DEM (see Eq.2) and then – nDSM (see Eq.1).

The goal of this study is to compare four algorithms to generate nDSM: region growing, progressive morphological filter, adaptive TIN surfaces and graph-cut.

Each algorithm applies different principles to detect ground facilities using height difference, from a constant threshold to variable. The study was carried out in two parts. The detected number of ground facilities is evaluated in the first part, the correctness of extracted building shapes – in the second part.

Print ISSN 1691-5402

Online ISSN 2256-070X

<https://dx.doi.org/10.17770/etr2019vol2.4032>

© 2019 Kodors. Published by Rezekne Academy of Technologies.

This is an open access article under the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

## II. LIDAR DATA PREPROCESSING

LiDAR data is filtered and transformed before a nDSM generation. A laser pulse provides many returns in the case of vegetation, considering this fact a point cloud is filtered by the last return.

This study compares grid-based methods, when a point cloud is projected into 2D grid and segmentation algorithms are processing pixels.

In this study, the point cloud is transformed using the lowest points, when the height of the lowest point is assigned to the pixel to obtain 2D DSM (see Fig.1).

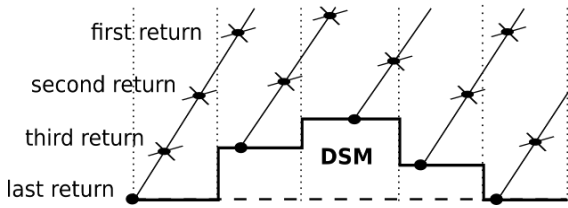


Fig. 1. Schema of DSM generation from LiDAR data

## III. SEGMENTATION ALGORITHMS

This study compares four algorithms to generate nDSM and to detect man-made constructions. Each algorithm is selected considering its principle of height difference definition:

- region growing algorithm applies a constant threshold (see Fig.2a);
- adaptive morphological filter is a hierarchical method, which applies the *min* and *max* height of points in a processing widow (see Fig.2b);
- adaptive TIN surfaces – a hierarchical method, which considers the slope of earth and applies statistics to filter extreme points (see Fig.2c);
- graph-cut analyzes the local similarity of adjacent pixels considering the global information about clusters (see Fig.2d).

### A. Region Growing

It is classical image segmentation algorithm, which uses seeds for region growing. Each iteration of the main loop provides the addition of adjacent pixels to already existing region [1], if pixels satisfy a predefined condition (threshold).

```

Function RegionGrowing( dsm )
    segments ← ∅
    While( NextSeedPoint( dsm ) ≠ ∅ )
        region ← NextSeedPoint( dsm )
        Do
            contour ← GetAdjacentPoints( region )
            pixels ← FitnessFunc( region, contour )
            region ← region ∪ pixels
        While( |pixels| > 0 )
            segments ← segments ∪ region
    End while
    Return segments
End function
    
```

In the case of DSM processing, next seed is the point with the minimal height among all unlabeled pixels. Adjacent points can be selected using pixel connectivity with 4 or 8 neighbors. The selected adjacent points are compared with the closest segment point, if the similarity of points is greater than a predefined threshold, they must be included into the segment. Traditionally, the similarity of two points is defined measuring the height difference between them.

### B. Progressive Morphological Filter

Kilian et al. have proved, that mathematical morphology is applicable to remove buildings and trees from LiDAR data [2], however, the proposed method had difficulties with all non-ground objects of various sizes due to the fixed size of a filtering window. Therefore, Zhang et al. proposed progressive model, which increases the window sizes of morphological filters gradually [3].

The mathematical morphology composes operations based on the set theory. Two fundamental operations, the “dilation” (see Eq.4) and the “erosion” (see Eq.5), are applied to enlarge (dilate) or to reduce (erode) the size of objects in an image [3]. The combination of an erosion and a dilation generates “opening” and “closing” operations that are applied to filter LIDAR data.

$$d_p = \max(w \in W_p). \quad (4)$$

$$e_p = \min(w \in W_p), \quad (5)$$

where  $p$  – a pixel,  $W_p$  – the window around a pixel  $p$ .

The opening operation is achieved by performing the erosion of image followed by the dilation, while the closing operation is accomplished by carrying out the dilation firstly, and then – the erosion [3].

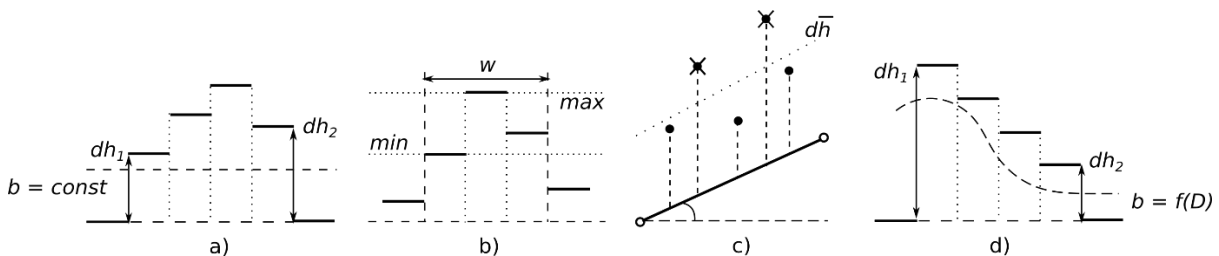


Fig. 2. Schemas of height usage: a) region growing; b) progressive morphological filter; c) adaptive TIN surfaces; d) graph-cut

```

// s - slope of earth elevation,
// dx - cell size (in geospace),
// b - half of window size,
// n - number of iterations,
// dh0 - initial height difference
// (threshold)
// mh - maximal difference of height

Function PrMFiltr(dsm, s, dx, b, n, dh0, mh)
    dh ← dh0
    labels ← 0 // |labels| = |dsm|
    windows ← {f:2bk + 1, b ∈ Z+, k ∈ [0; n - 1]}
    For each w in windows
        dsm' ← OpeningFunc( dsm, w )
        //p - pixel, p' ∈ dsm'
        For each p in dsm
            If (p - p') > dh then
                labels[p] ← k+1
            End if
        End for
        dsm ← dsm'
        dh ← s · w · (b - 1) · dx + dh0
        If dh > mh then
            dh ← mh
        End if
    End for
    //pixels equal to zero are ground points
    Return labels
End function
    
```

### C. Adaptive TIN surfaces

This method is based on constructing of triangular irregular network (TIN), when statistics on the height distance from point to triangle facet is obtained and points are filtered using the median in each iteration [4].

This study applies simplified model without statistics on angle and without mirror points. Only the vertical distance to point is calculated (see Eq.7) using the equation of plane (see Eq.6).

$$Ax + By + Cz = D. \quad (6)$$

$$dh = z - z' = z - \frac{D - Ax - By}{c}, \quad (7)$$

where  $p = (x, y, z)$  are the coordinates of points and  $p' = (x, y, z')$  is the point's projection on triangle facet.

```

Function AdaptiveTINsurfaces( points )
    dx ← 64
    While dx > 0.5
        dsm ← GetDSMbyMinPoints( points, dx )
        tin ← GenerateTIN( dsm )
        distances ← DistPntTriFacet( tin, points )
        median ← GetMedian( distances )
        stdDev ← GetStdDev( distances )
        points ← {p ∈ points | distances < median +
            + 2 · stdDev}
        dx ← dx / 2
    End while
    Return points
End function
    
```

### D. Graph-cut

Min-cut/max-flow algorithms are graph methods to complete image segmentation. It was proved, that graph-cut algorithms can complete the semantic segmentation of remote sensing data, if seed points belong to classes and the distance between adjacent points is defined using metric based on class features [5]. This method was applied to detect buildings using LiDAR data [6]. One of graph-cut methods is Dinic's algorithm.

```

Function DinicsAlgorithm( dsm )
    oSeeds ← FindObjectSeeds( dsm )
    bSeeds ← FindBackgroundSeeds( dsm )
    graph ← CreateGraph( dsm, oSeeds, bSeeds )
    source ← GetSource( graph )
    sink ← GetSink( graph )
    path ← FindShortPath( source, sink, graph )
    While path ≠ ∅
        minv ← GetMinimalCapacity( path )
        graph ← LaunchFlow( graph, path, minv )
        path ← FindShortPath( source, sink, graph )
    End while
    segments ← CutGraph( graph )
    Return segments
End function
    
```

In the case of DSM processing, objects are surface facilities and background – ground. The seeds are the pixels with strong height difference, the higher point is assigned to object, the lower – to background.

## IV. MATERIALS AND METHODS

The experiment is carried out in two stages:

1) the calculation of detected buildings;

2) the classification of all nDSM objects using the random forest algorithm based on the evaluation of geometric features.

The detected building is identified using the intersection of manually created layer and generated nDSM, but the shape of detected building can be distorted enough to be unclassified. Other problem are noise shapes, which are similar to buildings. Therefore, the shapes of nDSM objects are classified using the random forest algorithm, which classifies shapes using their geometric features. The classification method is described in the source [7].

The experiment was completed using the LiDAR data of urban area. The area of region is 1 km<sup>2</sup>. It has relatively flat surface and contains forested areas and 251 buildings.

## V. RESULTS AND DISCUSSIONS

The example of the detected buildings is depicted in Fig.3. The number of detected buildings by each algorithm is provided in Table I.

Tables II-V are confusion matrices to evaluate the shapes of detected objects, where the columns are expected outputs ( $E$ ), the rows – classification results

(R). The number of classified buildings in Tables II-V is different comparing with Table I, because Table I depicts the number of buildings, but Tables II-V depict the number of building parts. Many parts can belong to one building, if algorithm distorted it (see Fig.3c-d). On the other hand, algorithm can group many buildings into one shape (see Fig.3b).

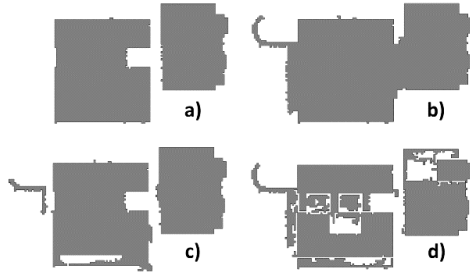


Fig. 3. Fig.3. Detected buildings by different algorithms: a) region growing; b) progressive morphological filter; c) adaptive TIN surfaces; d) graph-cut

TABLE I. NUMBER OF DETECTED OBJECTS

Algorithm	Buildings	Noises
Expected Value	251 (100%)	0
Region Growing	159 (63%)	419
Adaptive TIN Surfaces	245 (98%)	598
Graph-cut	247 (99%)	321
Progressive Morphological Filter	248 (99%)	638

TABLE II. CONFUSION MATRIX OF REGION GROWING

R. / E.	Buildings	Noises
Buildings	<b>0.228</b> (128)	<b>0.127</b> (71)
Noises	<b>0.025</b> (14)	<b>0.620</b> (348)
Total: 561; Accuracy: 0.849; Kappa: 0.646		

TABLE III. CONFUSION MATRIX OF ADAPTIVE TIN SURFACES

R. / E.	Buildings	Noises
Buildings	<b>0.198</b> (162)	<b>0.092</b> (75)
Noises	<b>0.073</b> (60)	<b>0.638</b> (523)
Total:820; Accuracy: 0.835; Kappa: 0.592		

TABLE IV. CONFUSION MATRIX OF PROGRESSIVE MORPHOLOGICAL FILTER

R. / E.	Buildings	Noises
Buildings	<b>0.212</b> (183)	<b>0.124</b> (107)
Noises	<b>0.049</b> (42)	<b>0.615</b> (531)
Total: 863; Accuracy: 0.827; Kappa: 0.590		

TABLE V. CONFUSION MATRIX OF GRAPH-CUT

R. / E.	Buildings	Noises
Buildings	<b>0.389</b> (222)	<b>0.007</b> (4)
Noises	<b>0.047</b> (27)	<b>0.556</b> (317)
Total: 570; Accuracy: 0.946; Kappa: 0.888		

Considering the data of Table I, graph-cut, progressive morphological filter and adaptive TIN surfaces have the best results (98% – 99%), in turn, the region growing has the greatest loss of detected buildings – 37%. And graph-cut provides the smallest number of noises.

The analysis of confusion matrices (see Table II-V) provides next results:

- graph-cut obtained the best classification results (see Table V);
- despite the fact, that region growing algorithm detects the smallest number of buildings, it has sufficiently good classification results (see Table I-II);
- the main problem of region growing algorithm, adaptive morphological filter and adaptive TIN surfaces; are noises classified as buildings (see Tables II-IV).

Considering the large amount of classified noises as buildings, the visual analysis of noises was completed. Fig.4 depicts the noises of similar region, where region growing, progressive morphological filter and adaptive TIN surfaces provide visually similar noises, compact and with larger area unlike graph-cut noises. According to the research [7], the compactness is the main feature in the classification process just like area. Therefore, these methods obtain the smaller classification accuracy as opposed to graph-cut.

Other problem of progressive morphological filter and adaptive TIN surfaces is exactly the usage of window, which approximates results for all pixels in the window (see Fig.5).

Nowadays, the popular solution for semantic segmentation is deep learning. For example, deep learning scholars propose next results for building detection using airborne LiDAR data: overall kappa – 0.89 [8] and  $F_1$  score 0.93 [9] and 0.95 [10]. So, the accuracy of graph-cut method, which provided kappa 0.888 and  $F_1$  score 0.933 (recalculating confusion matrix), is comparable with deep learning solutions.

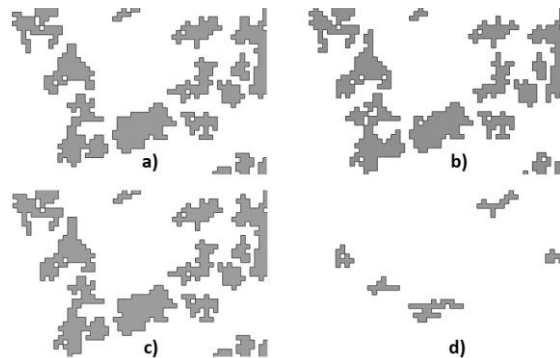


Fig. 4. Noises provided by algorithms: a) region growing; b) progressive morphological filter; c) adaptive TIN surfaces; d) graph-cut

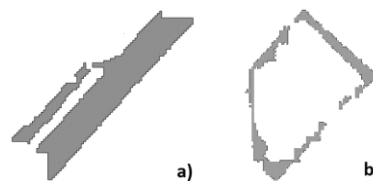


Fig. 5. Examples of window cuts: a) progressive morphological filter; b) adaptive TIN surfaces

## VI. CONCLUSIONS

The completed experiments have showed, that graph-cut method detects the largest amount of buildings (99%) and provides the best classification accuracy (overall accuracy 0.95, kappa 0.89 and  $F_1$  score 0.93).

The main problem of other methods is the high number of noises, which have large area and compact shapes similar to buildings.

The usage of methods with dynamic threshold better detects buildings, however, methods must be cluster based for more correct shape extraction.

## ACKNOWLEDGMENTS

Author expresses his gratitude to the State Land Service of Latvia and to Latvian Geospatial Information Agency for providing samples for the research purposes.

## REFERENCES

- [1] S. Szénási, "Distributed Region Growing Algorithm for Medical Image Segmentation," *International Journal of Circuits, Systems And Signal Processing*, vol. 8, pp. 173-181, 2014.
- [2] J. Kilian, N. Haala, and M. English, "Capture and evaluation of airborne laser scanner data," *International Archives of Photogrammetry and Remote Sensing*, vol. 31, pp. 383-388, 1996.
- [3] K. Zhang, S.-C. Chen, "A Progressive Morphological Filter for Removing Nonground Measurements From Airborne LIDAR Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 4, pp. 872-882, 2003.
- [4] P. Axelsson, "DEM Generation from Laser Scanner Data using Adaptive TIN Models," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, pp. 110-117, 2000.
- [5] S. Kodors, "Land Cover Recognition using Min-Cut/Max-Flow Segmentation and Orthoimages," *Proceedings of the 10th International Scientific and Practical Conference*, vol. 3, pp. 127-133, 2015.
- [6] S. Kodors, A. Ratkevics, A. Rausis, J. Buls, "Building Recognition Using LiDAR and Energy Minimization Approach," *Procedia Computer Science*, vol. 43, pp. 109-117, 2015.
- [7] S. Kodors, "Geometric Feature Selection of Building Shape for Urban Classification," *Proceedings of the 11th International Scientific and Practical Conference*, vol. 2, pp. 78-83, 2017.
- [8] Z. Sun, X. Zhao, M. Wu, C. Wang, "Extracting Urban Impervious Surface from WorldView-2 and Airborne LiDAR Data Using 3D Convolutional Neural Networks," *Journal of the Indian Society of Remote Sensing*, vol. 47, pp. 1-12, 2018. <https://doi.org/10.1007/s12524-018-0917-5>
- [9] Z. Yang, W. Jiang, B. Xu, Q. Zhu, S. Jiang, W. Huang, "A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds," *Remote Sensing*, vol. 9, no. 9, p. 936, 2017. <https://doi.org/10.3390/rs9090936>
- [10] Y. Sun, X. Zhang, X. Zhao, Q. Xin, "Extracting Building Boundaries from High Resolution Optical Images and LiDAR Data by Integrating the Convolutional Neural Network and the Active Contour Model," *Remote Sensing*, vol. 10, no. 9, p. 1459, 2018. <https://doi.org/10.3390/rs10091459>