Combining Valley Following and Rule Based Segmentation for an Automatic Object Extraction from LiDAR Height Data

Melanie Kolditz, Cristina Petre and Jürgen Roßmann
RWTH Aachen University, Institute for Man-Machine Interaction, Aachen, Germany; {Kolditz, Petre, Roßmann}@mmi.rwth-aachen.de

Abstract. For remote sensing data analysis, the automatic extraction of objects in natural, unstructured environments plays an important role in order to obtain a semantic world model. Due to the diversity of object shapes, such an environment represents a serious challenge for an accurate automatic segmentation and classification. Furthermore, image based data suffers from shading effects and has to be orthorectified. These effects do not occur in normalized digital surface models (nDSM), derived from LiDAR data, which represent the heights of objects, such as single trees and buildings. The algorithm that was developed to extract single trees uses an nDSM with a resolution of 0.4m per pixel. In a first segmentation step a valley following approach is used to find contours of objects. These contours are distinct for coniferous trees, whereas they are more blurred for deciduous trees. To avoid an upper bound on classification accuracy, single tree based species classification prefers over segmentation to under segmentation. Tree tops are detected as the local maxima in the nDSM and contours containing more than one local maximum are further separated using rule based segmentation. This algorithm is validated in a testbed within the Virtual Forest project. The results are furthermore compared to a single tree detection approach, which uses prior knowledge of the tree species to find the tree positions.

Keywords. Valley Following, tree delineation, LiDAR, Rule Based Segmentation, object recognition, height models.

1. Introduction

The analysis of remote sensing data allows the assessment of information about the earth’s surface in a time and cost efficient manner. In order to yield maximum benefit out of gathered data, objects of interest should not only be extracted but also classified. As part of the project “Virtual Forest” [1], a semantic forest model is developed and implemented in a 4D geo information system (4D-GIS). This modelling step results in an identity card for each single tree, which contains information of e.g. the tree position, its species or its unique ID as shown in Figure 1.

To derive such information, regions of interest – in this context individual tree crowns – have to be detected in remote sensing data. Current approaches can be categorized by the type of input data, orthophotos or airborne laser scanner data, used for segmentation. Algorithms using aerial true colour (RGB) or colour infrared (CIR) images are described in [2], [3] and [4]. Their segmentation results vary between 73% and 95% depending on both the resolution of the input images and the forest types. The best results are achieved with CIR images, with a resolution of less than 1cm per pixel in non-mixed forest stands. The three dimensional structure of a forest can be derived from LiDAR data and is used for single tree delineation in [5], [6] and [7]. The same accuracies as achieved from image data can be obtained from laser scanner data. Very good results of up to even 100% are only possible in non-mixed coniferous forest stands.
Melanie Kolditz, et al.: Combining Valley Following and Rule Based Segmentation for an Automatic Object Extraction from LiDAR Height Data

To improve segmentation accuracies in mixed or deciduous stands, both image and LiDAR based algorithms often use prior knowledge of e.g. the corresponding crown size or the tree species [7] to adjust their parameters to each situation, or calculate several segmentation results and compare them to predefined tree models to choose the best fitting parameters [8].

However, image data always has to deal with orthorectification and illumination effects. Figure 2 shows the same forest area on the left side orthorectified and on the right side after true orthorectification where not only ground but also vegetation pixels are corrected. Due to the dark shadows on the ground, artifacts can occur in true orthophotos that might be an error source for a segmentation process. Therefore, LiDAR data has been chosen for the algorithm presented in this paper.

Within the project Virtual Forest, the image segmentation is a preprocessing step for the following tree species classification described in [9]. So the aim is not to detect single trees but to detect objects for the classification algorithm without using any prior knowledge of the object shapes or the tree species. To avoid an upper bound on classification accuracy, one segmented object should not contain more than one tree.

The most common segmentation algorithms for this task are region growing [10], [11], hill climbing [12], [13], watershed [14], [15] and valley following [16], [17]. Both region growing and hill climbing reach their limits in mixed forest stands, as no certain threshold for the region extension can be set for the first, and the algorithm is very sensitive to pixel errors for the latter. Watershed based techniques have been tested in various ways, but require an a-priori knowledge in
the postprocessing part. Valley following approaches are based on the characteristics of human perception and are robust against pixel errors and so are appropriate for the specified task.

2. Methods

2.1. Test area and data

The test area Arnsberg is located in the northwest of Germany (51°23’N 8°03’E) and has a size of approximately 340km². With its mixed forest stands containing both coniferous (48%) and deciduous (52%) tree species, the algorithm can be evaluated for various situations.

LiDAR data was acquired in May and June 2008 by Milan Geoservice GmbH using a Riegl LMS-Q 560 airborne laser scanner in an altitude of 600 to 1000m. This results in a small footprint size of 0.3 to 0.5m and a point density of at least five points per square meter. Based on these point clouds both a digital terrain (DTM) and a digital surface model (DSM) with grid sizes of 0.4m were calculated. By subtracting the DTM from the DSM, a normalized digital surface model (nDSM) containing the height values of the vegetation, is obtained. All three models are depicted in Figure 4.

For reasons of data storage, the whole test area is divided into 500×500m tiles; with the aforementioned resolution of 0.4m per pixel, each grayscale image has a size of 1250×1250 pixels.

To evaluate the proposed algorithm, 100 trees have been segmented manually on the nDSM images as can be seen in Figure 5. Furthermore, there are single tree positions available, that are
derived from the volumetric algorithm described in [7]. This algorithm calculates the tree positions for each forest stand separately and uses prior knowledge of the object shape.

![Manually segmented tree regions.](image)

**Figure 5:** Manually segmented tree regions.

### 2.2. Tree region segmentation

The segmentation algorithm consists of several steps from the nDSM image to the segmented image as depicted in Figure 6.

![Flow chart of the segmentation algorithm.](image)

**Figure 6:** Flow chart of the segmentation algorithm.

#### 2.2.1. Preprocessing

For a tree species classification, only large trees that are visible in remote sensing data are relevant objects, whereas smaller vegetation is not relevant. Therefore, the first preprocessing step consists of limiting the height values. All pixels that are lower than a threshold of 4m are set to zero. Furthermore, data is limited to a maximum height of 50m as no higher trees are expected. In order to reduce noise and pixel errors in the input images, a Gaussian filter

\[
G(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2+y^2}{2\sigma^2}}
\]

is applied to the nDSM images. The parameters for the kernel size \(x \times y\) and the scale \(\sigma\) have to be chosen for the valley following and the tree top detection step separately.

For the valley following, the filter parameters have to be set to values that reduce pixel errors on the one hand and on the other hand do not blur the borders of the individual tree crowns. A 7×7 kernel with \(\sigma=1.4\) was chosen for this task. For tree top detection, the image has to be filtered so that each tree has only one maximum to be detected. A light 5×5 kernel with \(\sigma=1.25\) proved to be best suitable for this detection, as larger kernels tend to smooth out valid tree tops in coniferous regions.
2.2.2. Valley Following

The concept of the Valley Following algorithm is based on the idea that trees appear as domes in nDSM images. Pixels with lower gray values form V-shaped valleys, which delineate tree crowns. The first step is a classification of the input image into valley pixels (VP) and non-valley pixels (nVP). Due to the preprocessing step in setting all pixels smaller than 4m to zero, all pixels with a gray value of zero can be marked as VPs. The image is scanned in all possible search directions regarding the 8-neighbourhood of a pixel as depicted in the left of Figure 7. The investigated pixel is marked as a possible VP, if it is flanked on at least two sides by pixels with higher gray values shown in Figure 7 middle and right.

These possible VPs might still contain pixels from inside a tree crown, because of branches sticking out or remaining pixel errors. So the algorithm scans, beginning from the top-left corner, the direct 8-neighbourhood of each valid VP. Each possible VP in this neighbourhood is marked as valid. This step is iterated until either no more possible VPs can be found, or a maximum iteration amount is reached. This amount is set to 16, as further iterations did not improve segmentation results.

The valleys are now 8-pixel connected as shown in Figure 8 left. In order to clearly separate the tree regions, the valleys have to be 4-pixel connected as depicted in Figure 8 right. This connection can be achieved by erosion with the element shown in the middle.

The effects of the erosion can be seen in Figure 9 and are more obvious in deciduous stands.
2.2.3. Tree top detection

The proposed algorithm delineates a region around each tree top, so the tree top detection step is important for a correct segmentation. As for the following tree species classification of over segmentation is preferable to an under segmentation, this step should rather detect two peaks for one tree than one peak for more trees. To achieve this result, local maxima are detected in the smoothed image. One pixel is marked as a local maximum – also called seed – if it has the highest gray value in its surrounding 8-neighbourhood as depicted in Figure 10.

![Figure 10: Local maximum (LM) in an 8-neighbourhood.](image)

2.2.4. Region classification

Combining the results of both the valley following segmentation and the detected seeds of the tree top detection, the objects can be classified into invalid, well detected and conglomerate regions based on a set of rules.

Regions containing no seed are in most cases due to the erosion step and are artifacts that belong to neighbouring regions. For regions containing one seed both valley following and tree top detection yield the same result. All other objects contain more than one tree top and have to be further investigated. For the classification the following rules are applied:

1. A region containing exactly one seed is tagged as well-detected.
2. A region containing no seeds is tagged as invalid and deleted.
3. A region containing more than one seed has to fulfill at least one of the following conditions to be tagged as well-detected:
   a. $0.5\bar{A} < A < 1.5\bar{A}$ AND $\bar{C} < C$
      with $\bar{A}$ – area calculated as the sum of pixels
         $\bar{A}$ – mean area of well detected regions from rule 1
         $C = \frac{4\pi A}{P^2}$ – compactness (perimeter $P$ – sum of outline pixels)
         $\bar{C}$ – mean compactness of well detected regions from rule 1
   b. $0.5\bar{A} < A < 1.5\bar{A}$ AND $d_s < 1.2m$
      with $d_s$ – distance of two seeds
4. All other regions are tagged as conglomerate region.

These three possibilities are illustrated in Figure 11.
2.2.5. Post processing

A human interpreter would use prior knowledge of the object shapes to split the conglomerate regions. For the algorithm no such knowledge is available, so a probabilistic approach is used to assign the conglomerate pixels to a seed. A pixel belongs most likely to the seed closest to it. This problem can be solved using Voronoi networks, which aim to divide a plane in so called Voronoi cells in respect to given seeds. These Voronoi cells with the corresponding Voronoi edges are depicted in Figure 13.

The whole decision making process is summarized in Figure 12.
To find the Voronoi edges, the Euclidian distance between two seeds is evaluated. An example of split objects can be seen in Figure 14.

In the final region image, no black borders between two objects are desired, so the pixels lost by the erosion step have to be added to the regions using the corresponding dilatation. An example of the final segmentation is given in Figure 15.

2.3. Evaluation

The algorithm is evaluated against both manually segmented regions and automatically derived tree positions.

Only manually segmented trees containing exactly one region are correctly delineated. Trees containing more than one region are over segmented, which is not correct for a single tree delineation. But the algorithm aims to calculate objects for a tree species classification, where an
over segmentation is acceptable. Given this requirement, only regions that contain more than one tree, i.e. under segmented trees can influence classification accuracies. Therefore, two different success rates are calculated for the algorithm; the hard and the soft success rate. The differences are given in Table 1.

**Table 1**: Hard and soft success rates for evaluation against manually segmented trees.

<table>
<thead>
<tr>
<th>Hard success rate</th>
<th>One tree contains one region</th>
<th>One tree contains more than one region</th>
<th>One tree contains part of a region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

| Soft success rate | Correct | Correct | False |

The evaluation with the automatically derived tree positions from the volumetric algorithm in [7] is similar to the evaluation explained before. For the hard success rate only one region per tree position is a correct result. An over segmentation results in regions without a tree position are counted as correctly segmented for the soft success rate. If one region contains more than one tree, the image is under segmented, which is a wrong segmentation for both success rates. The evaluation scheme is summarized in Table 2.

**Table 2**: Hard and soft success rates for evaluation against automatically detected tree positions.

<table>
<thead>
<tr>
<th>Hard success rate</th>
<th>One region contains one tree</th>
<th>One region contains no trees</th>
<th>One region contains more than one tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>

| Soft success rate | Correct | Correct | False |

3. Results

An overlay of manually delineated trees with the segmented regions is given in Figure 16.

![Figure 16: Overlay of manually delineated trees with the segmented tree regions.](image)

Most of the reference trees, given with their white borders, are correctly detected by the proposed algorithm. The blue encircled tree has a larger region than the reference tree, because
a human interpreter cannot see lower gray values in the nDSM image, whereas the algorithm can detect the lower parts of the crown. Therefore, an overlapping of at least 50% is evaluated as a correct recognition. Other trees in this image are over-segmented, but the detected regions are still large enough to be valid objects for a classification algorithm. Only regions overlapping more than one tree – the red encircled tree in Figure 16 are falsely detected. For all 100 manually detected trees, the algorithm achieves a hard success rate of 77% and a soft success rate of even 91%. Due to over-segmentation, even the last 9% can be classified at least for a part. Given the red encircled falsely segmented tree, about ¾ of this tree can be classified correctly.

The results of the evaluation with the tree positions calculated with the volumetric algorithm are shown as an overlay with the tree regions in Figure 17.

Figure 17: Overlay of the tree positions from the volumetric algorithm and the segmented objects.

Most of the calculated tree regions contain exactly one tree position, so that a hard success rate of 82% can be achieved. Also regions containing no tree position are valid classification objects, so that the soft success rate is increased slightly to 84%. On the other hand, 16% of the regions contain more than one tree, which induces an under-segmentation of the image. Whether the mistakes have been made by the volumetric algorithm or the object segmentation algorithm cannot be answered without further evaluation.

4. Conclusions

The proposed object recognition method is able to segment an image without using any prior knowledge into valid classification objects with an accuracy of 91%. The same parameters are applied to every 500×500m image. Other algorithms such as the compared volumetric algorithm use afore segmented forest stands instead of whole images, which allows the use of an adaptive parameter choice. As described before, accuracies up to 100% can only be achieved in non-mixed coniferous forest stands. Therefore, the detected objects are suitable for a tree species classification even in deciduous or mixed stands.
Nevertheless, the algorithm can still be improved. All variables – for the Gaussian filters and the amount of iterations – have been chosen by visual interpretation of the segmentation results. Optimal parameter sets can be found by varying each parameter and analyzing the results using the receiver operator characteristics. Furthermore, the division of conglomerate regions can be improved by not simply using Voronoi networks but searching for valleys near the Voronoi separation line. Also the reference data should be increased by delineating more trees and using ground truth tree positions instead of the results of another tree detection algorithm.

Acknowledgements

This project was co-financed by the European Union and the federal state of North Rhine–Westphalia, European regional development fund (ERDF). Europe – investing in our future.

References

