AUTOMATIC TREE-CROWN DETECTION IN CHALLENGING SCENARIOS

Dimitri Bulatov, Isabell Wayand, Hendrik Schilling

Fraunhofer IOSB, Department Scene Analysis, Gutleuthausstr., 1, 76265, Ettlingen, Germany
{dimitri.bulatov, isabell.wayand, hendrik.schilling}@iosb.fraunhofer.de

Commission III, WG III/4

KEY WORDS: 3D-modeling, Classification, Hotspots detection, Surface reconstruction, Tree, Watershed Transformation

ABSTRACT:

In this paper, a new procedure for individual tree detection and modeling is presented. The input of this procedure consists of a normalized digital surface model NDSM, and a possibly error-prone classification result. The procedure is modular so that the functionality, the advantages and the disadvantages for every single module will be explained. The most important technical contributions of the paper are: Employing watershed transformation combined with classification results, applying hotspots detectors for identifying treetops in groups of trees, and correcting NDSM by detecting and geometric reconstruction of small anomalies, such as earth walls. Two minor contributions are made up by a detailed literature research on available methods for individual tree detection and estimation of tree-crowns for clearly identified trees in order to reduce arbitrariness by assigning trees to one of the few types in the output model.

1. INTRODUCTION

There are many applications for 3D models of urban terrain, such as urban planning, navigation, civil security, and others. This is why scientific communities from all over the world pursue more accurate, more precise, and more semantic reconstruction of urban terrain model instances from sensor data. Here, context-based, semantic models mean that the most common instances of urban terrain, such as building, trees, roads, etc., are first detected and then reconstructed using soft constraints that characterize these instances. Examples of such constraints are piecewise planar building roofs or smooth course of roads. The contribution of this work is to establish a similar workflow for trees. As input we have an ALS (airborne laser scanning) data sampled into an elevation map (denoted also as Digital Surface Model, DSM) and an orthophoto. Starting from a possibly imprecise precise classification result for the tree class, we make use of a soft constraint that a local maximum in a smoothed DSM restricted to the tree class basically corresponds to a tree top. Doing so, we will not only be able to perform individual detection of trees and obtain more information on their position, height and diameter, but also to detect and to correct the systematic errors in the classification result whereas possible.

Detecting and exact modeling trees is important because of many application in forestry, life sciences, but also in security-related issues However, to create a separate model for every tree is computationally expensive, especially if interaction is required. In addition, we are striving for a fast procedure in which collection of training data and/or non-local optimization of complicated energy functions are not desirable. As a consequence, we opt for a compromise: after performing single tree detection, we estimate its position, height, and diameter. Then the tree is modeled as a circular or elliptic region. This is sufficient for many applications. Finally, we wish to assign it into one of very few types. For each of these types, a library entry can be created, optionally adjusted according the the season and day-time, and, finally, copy-and-pasted within the model on the relevant positions with corresponding scales. This goes hand in hand with the state-of-the-art activities on the semantic representation of urban terrain (Bulatov et al., 2014) in which the resulting models are usually 1) visually appealing, 2) well-compressed, because firstly building roofs and other planar objects can be represented by a few polygons and secondly, because less important objects, such a grass, can be modeled by a few so-called geo-typical objects (dandelions, blades of grass of difference heights, etc.), 3) easy to modify and 4) inter-operable in the sense that a *tree model* can be given further important properties, such as collision geometry, flammability, and many others.

As for the second question, namely, correcting systematic errors, it is well-known that classification in aerial data is often a challenging task because overlapping of classes is always given, especially in the case of trees that occlude other objects: roads, cars, and even buildings. Besides, the particular interest of this paper is dedicated to types of objects that are not included in the list of classes because their appearance is seldom. An example of such an *anomaly* is an earth hill. If it is grass- or shrubbery covered, it is extremely difficult to differentiate it from a group of trees. Alternatively, if such a hill is covered with trees, computation of 3D positions of these tree trunks becomes a very critical issue with respect to the task we set in the last paragraph. Hence, it will be necessary to recognize the anomalies in the terrain and to correct them.

The paper is organized as follows: In Sec. 2 and 3, respectively, we refer to the related work on tree detection and explain the important preliminaries on obtaining tree areas from sensor data. In Sec. 4, four main tools used for individual tree detection are presented while in Sec. 5 we explain how the anomalies in the terrain can be detected and corrected. In Sec. 6, we describe a simple routine for tree crown classification. Then, Sec. 7 summarizes the proposed algorithm. The results for a challenging dataset are shown in Sec. 8 while for main conclusions and ideas of future work we refer to Sec. 9.

2. OVERVIEW OF PREVIOUS WORK

There is a large amount of methods on tree detection in the literature, see surveys due to (Kaartinen et al., 2012, Eysn et al., 2015). Among contributions analyzed in (Bechtel, 2007), (Pollock, 1996) proposes geometric models (generalized ellipsoid) and fits it by means of a template matching procedure with different parameters (until the height difference falls under 50%). It

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XLI-B3-575-2016

is a typical top-down method. Because of many parameters, it is rather unhandy for big data. A sequence of top-hat operators (Andersen et al., 2001) is applied to extract elevated objects of a predefined size. The hill-climbing method due to (Persson, 2001) starts from the classification result. All pixels with height values over a threshold, which lie in the vegetation mask of a strongly smoothed elevation map, are considered as starting points for a local maximum search by moving into the direction of the steepest gradient.

The contribution of (Pouliot et al., 2002) is a transect approach from the local maximum of the elevation map to the supposed border of tree. Region-growing approach is proposed by (Dalponte et al., 2014) and (Secord and Zakhor, 2007), where features are calculated from the laser and image data and the weights are determined by a learning procedure. The work of (Straub, 2003) mentions watershed transformation of the DSM. Edges detected in the images are used to specify borders of hypotheses for the individual tree regions. To assess a hypothesis, he considers geometric (area, eccentricity, and second derivative) and radiometric properties (near-infrared, texture measures, etc.). The conclusion is drawn that combining borders of the watershed regions with edges is only necessary to separate trees and groups of trees from the background, not from each other. In addition, it is a cumbersome process of edge filtering since some edges are usually within tree crown. Hence, this step is not necessary if we have a reliable classification result; however, it does bring essential improvements in the areas where the classification result is not correct for one of the reason pointed out in the introduction section. Also, we see that because of its simplicity and efficiency, many other authors use watershed transformation (Wang et al., 2004, Koch et al., 2006, Reitberger, 2010).

3. PRELIMINARIES ON CLASSIFICATION AND EXTRACTION OF TREE REGIONS

The input of our algorithm consists of a DSM and a digital orthophoto. In what follows, a short description of our method on extraction of tree regions is given. This is important to understand the origin of systematic errors in the classification result.

First, the Digital Terrain Model (DTM) is computed by means of the procedure described in Sec. 2.1 of (Bulatov et al., 2014). By computing the difference between DTM and DSM, we obtain the normalized DSM (NDSM) with clearly distinguished smaller elevated objects: Buildings, trees, fences, street lamps, earth walls, and others. We perform the classification of the terrain similarly to (Lafarge and Mallet, 2012). Four measures (relative elevation measure f_h, planarity measure f_p, normalized difference vegetation index (NDVI) measure f_n, and entropy measure f_e of the orthophoto) are computed for each pixel $\mathbf{x}$ to distinguish between four main classes $l \in \{\text{building, tree, grass and ground}\}$. These four measures are collected into energy data terms for the four classes, as described below. A smoothness term penalizing differences of labels between neighboring pixels is added to the energy function. This function is minimized using the semi-global method of (Hirschmüller, 2008). The resulting label map is post-processed by the median filter and particularly for the class *tree*, by morphological operations (3×3 or 5×5 closing and opening).

For example, we show here how the data term is built for trees: They are characterized by a non-negligible elevation, by a rather low planarity, a high measure of NDVI, and a high local texture variation, that is, the entropy. Thus, the data term $E_{l(\mathbf{x})=\text{tree}}$ for a pixel $\mathbf{x}$ to belong to a tree is given by:

$$E_{l(\mathbf{x})=\text{tree}} = (1-f_h)f_p(1-f_n)(1-f_e), \qquad (1)$$

where $f_\circ = f_\circ(\mathbf{x})$. However, for f we use a sigmoid-like function asymptotically approaching 0 and 1:

$$f_\circ(x) = \frac{w(\mathbf{x})}{w(\mathbf{x})+1}, w(\mathbf{x}) = \exp\left(\frac{\circ(\mathbf{x})-\sigma}{\kappa}\right) \qquad (2)$$

with empirical parameters σ and κ as well as $\circ = \{h, p, n, e\}$, instead of a truncated linear function chosen by (Lafarge and Mallet, 2012). The reason is that otherwise an elevated region where the term $(1-f_h)$ vanishes would become completely a tree or a building independently on other measures.

4. TOOLS FOR INDIVIDUAL TREE DETECTION

In this section, we describe the four important tools for detection of isolated trees. One can ask why we apply on our data not less than four modules. The reason is that we wish to process large, mostly "smooth and trouble-free" parts of the data by fast, straight-forward modules. Then we basically put these parts of data aside. At the same time, there are anomalies in the data which must be detected and corrected by advanced methods. We show in Fig. 1, top the flow-chart of the proposed algorithm and give more details of it in Sec. 7.

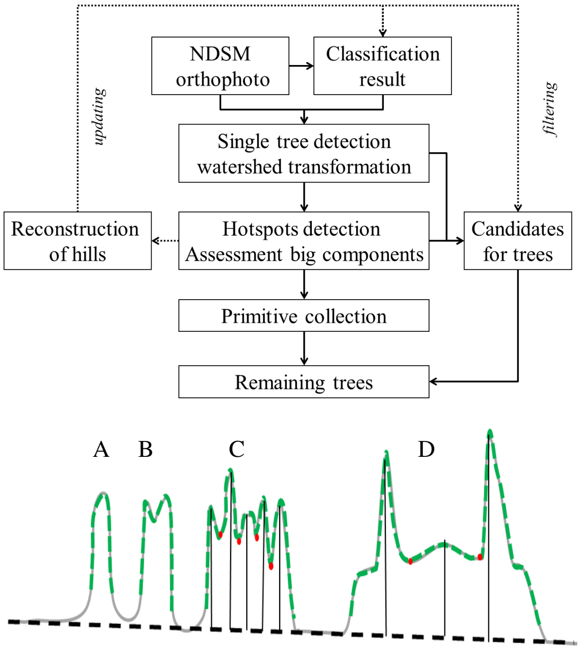

Figure 1: Top: overview of the algorithm with dotted lines depicting correction of the anomalies. Bottom: Proof of principle for individual tree detection with watershed. The height profile for DSM and DTM are shown by solid grey curve and dashed black curve, respectively. The result of classification is depicted by a thick solid green line while centers and borders of different watershed components are shown by vertical black lines and red points, respectively.

For some applications, for example, security tasks, it may be sufficient to identify isolated trees as closed, almost circular components belonging to the tree class and having a number of pixels in a predefined range. Therefore labeling of components and their filtering by eccentricity (0.8 is an empirical value) and area (between 5 and 150 m^2) is a fast and mostly reliable approach. For compactness, the procedure of *single tree identification* will be denoted by $\mathcal{S}$. In the illustration of the proof of principle, Fig. 1, bottom, we see that applying $\mathcal{S}$ in the scenario A leads to a correct detection of tree (which is also the most frequent case). In the scenario B, it is not immediately clear from the DSM alone whether we have two trees pretty near to each other or a single tree with two crowns.

Our default method is the *watershed* transformation ($\mathcal{W}$) applied on the negative NDSM, but restricted to the class *tree* of our classification result. A watershed component has its root in a local minimum of an image and it ends at the shedding line to the neighbored component. As mentioned before, the transformation is very fast and it is suitable to separate trees from each other while the separation from the background has already taken place by means of the classification. However, there are two disadvantages of this strategy which either leads to single tree decompositions or to regions that are too big for a single tree. First, $\mathcal{W}$ is sensitive to noise and hence, superfluous components are formed around each local minimum of the input image. This is why we need to perform a median filtering and/or a discretization. As a consequence, the difference of heights between two crowns may get over-smoothed and we get components of a larger size than a usual tree. The other problem stems from classification errors: In the dataset analyzed in Sec. 8, several earth hills not covered with grass usually belong to the tree class and, by applying watershed transformation, we obtain one or more rather big components. This is visualized in Fig. 1: situations B and C show a successful while situation D shows an unsuccessful delineation.

In order to bound a component and thus to find an extension of the tree in x and y direction, (Straub, 2003) proposed combination of watershed components and edges in the images. However, the algorithm for Image Segmentation by Optimization of Levels (ISOL) due to (Anderer et al., 1989) has proved to be a suitable alternative. ISOL searches for salient regions, that is, those for which the pixel list does not change significantly after a slightly increasing or decreasing the threshold for gray values. Thus, it is the precursor of the well-known MSER (Maximum Stable Extremal Region) algorithm (Matas et al., 2002), which has produced worse regions. The procedure for detection of *hotspots* based on ISOL and denoted here as $\mathcal{H}$ allows distinguishing trees atop of the hills in scenario D of Fig. 1. In fact, we would obtain three nested components: Two for trees and one for the hill, which can be immediately filtered out because of its magnitude. However, because of the nested components, ISOL without postprocessing should be rather taken for identifying groups of trees (to be able to model hills) rather than for detection of individual trees. For this latter task, we need a post-processing step in which the components are analyzed by the variance of elevations, by eccentricity and by registering their overlap with other components (which is time-consuming). Besides computing time, the necessity of forming a discrete image must be mentioned as a disadvantage of ISOL method.

After performing extraction of single trees, there can still be components which are too big and/or too stretched for a single tree. Here, groups of trees are extracted by the procedure $\mathcal{P}$ (*primitive collection*) proposed in (Bulatov et al., 2014): The highest tree point yields the position of the first tree. Then, all pixels within a circular region around this point are excluded from the component; the radius of the circular region corresponds to the typical tree radius of 6m. This procedure is repeated until there are no remaining pixels in the component.

5. CORRECTION OF NDSM BY DETECTING AND RECONSTRUCTING HILLS

In this section, we want to find out whether the large components obtained after applying $\mathcal{W}$ on the NDSM intersected with the tree class may originate from filtering/discretization artifacts or from anomalies in the DTM/classification. In the latter case, we will perform the reconstruction of the hill surface.

The choice for watershed is made for reasons of speed: We are aware that if the density of trees atop of a hill is very high, it would collapse into very many small regions (hence, ISOL applied on the *whole* NDSM) would be more suitable); however, in our dataset earth walls mostly serve exercise purposes and do not contain too many trees. The situation is thus similar to that exemplified in scenario D of Fig. 1, where a component to be analyzed is formed as a union of three big components with local maxima on the trees and hill top. Cleaning such components from trees – which now are outliers and should be excluded from surface reconstruction – by ISOL and morphological operations (see the red points in Fig. 2, top right), comprise the first step of component analysis.

Next, for all components, we considered four properties. First, the area of the component should not be too small. Second, we computed the average of the NDSM values over all pixels within the component in order not to confuse hills with crop fields etc. Analogously, we assessed the average of second derivatives that were calculated by means of the Laplace filter. This is done because earth walls are more likely to be approximated by planes than dense groups of trees, as one can see in two height profiles in Fig. 2, top left and bottom left. The fourth property we took into consideration was the distribution of relative elevation values. The reason is that the histogram sampled over a hill-like structure is expected to resemble a uniform distribution with some peaks. Hence, after sampling a histogram, we suppressed the bins with too many and too few entries. Eventually, the property assigned to the component is the quotient of standard deviation of the entries of remaining bins divided by the total number of points in the component. This property requires too many parameters and this is why we decided to omit it in favor of the three earlier mentioned measures (*area*, *elevation*, and *planarity*). The thresholds for these measures are empirical for the current implementation, but can be made a result of clustering procedure in the future. The components belonging to hills respectively non-hills are depicted by orange and cyan color in Fig. 2, top right.

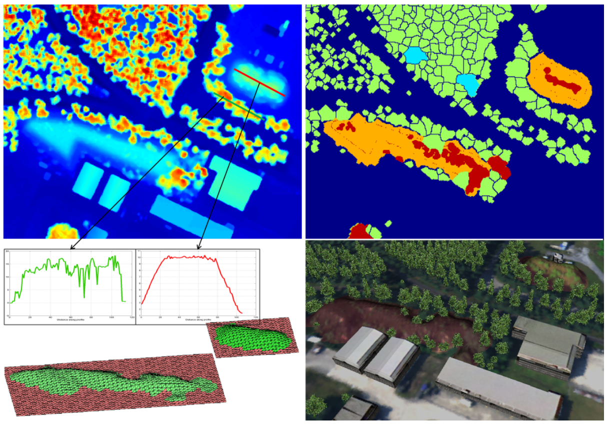

Figure 2: Detection and reconstruction of hills. Top left: Part of the DSM. Top right: Results of the tree detection by $\mathcal{W}$ (green) and $\mathcal{H}$ upon the hills (red) in the corresponding part. The components recognized and non-recognized as hills are marked in orange and cyan color, respectively. Bottom left: Anomalies in the DTM are reconstructed (red and green triangles) and modeled (green triangles). Bottom right: A view of the urban terrain model, with building models computed with the procedure of (Bulatov et al., 2014), hills, and tree regions.

The geometric reconstruction of the surface of the hill starts by collecting points in the bounding box around the hill, though not members of the class *building* and *tree*. A functional considering the vertical distances of the points to the surface and piecewise planarity of this surface is minimized. We decided to use the fast gridfit procedure (D'Errico, 2005) though L_1-spline-based optimization (Bulatov and Lavery, 2010) is more suitable for man-made structures. We apply the constant grid size and receive as output the 3D coordinates of the vertices. By computing the canonical triangulation from the interpolation surface and keeping only those triangles that have at least one point belonging to the hill and their neighbors (that is, at least one common vertex, see green surface in Fig. 2, bottom left), we create a 3D model for the hill. This 3D model is textured by means of the orthophoto, as shown in Fig. 2, bottom right (colors of hills were modified for a better visualization). The new NDSM is extracted for the pixels atop and around of the hill surface. At a later stage, it will be needed for analyzing trees upon the hills.

6. CLASSIFICATION OF TREES

As mentioned before, we wish to assign a tree to one of a few types. We consider here only those trees that have been collected either by $\mathcal{S}$ or by $\mathcal{W}$ module since otherwise point clouds bear risk to contain points from different trees. Also, keeping in mind the computing time, it is not the main goal of the paper to determine the tree type for every tree by means of numerous, time-consuming features since we only wish to make choice of tree types in the output model less random. We assess the conic shape in order to carry out the most intuitive subdivision of trees into two classes: Broad-leafed trees and coniferous trees. According to (Horn, 1971), a tree crown of a deciduous tree can be approximated with an ellipsoid while in the case of a coniferous, it is a cylinder or a cone. A 3D point $\mathbf{x}$ in homogeneous coordinates is a 4×1 vector while a conic is represented by a symmetric 4×4 matrix Q with 10 degrees of freedom. For each point $\mathbf{x}_i, i = 1, ..., n$ belonging to the tree crown parametrized by Q, we formulate an equation in terms of $q_0, q_1, ..., q_9$:

$$\mathbf{x}_i^T Q \mathbf{x}_i = 0, Q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ q_1 & q_4 & q_5 & q_6 \\ q_2 & q_5 & q_7 & q_8 \\ q_3 & q_6 & q_8 & q_9 \end{bmatrix}, \tag{3}$$

and obtain an equation system which is over-determined if $n \geq 10$. For numerical stability, we normalize the points in the way they have zero mean and unity standard deviation, and we set $q_2 = q_5 = 0$ to make one of the conic axes to be parallel to the vertical direction. The resulting equation system is solved with the Direct Linear Transformation (DLT) method (Abdel-Aziz and Karara, 1971). The half axes of the ellipse are given by

$$l_x = \sqrt{-d_1/d_4}, l_y = \sqrt{-d_2/d_4}, l_z = \sqrt{-d_3/d_4}, \tag{4}$$

where d are the eigenvalues of Q in the "natural" order, that is, the eigenvector matrix is close to the identity matrix. Our measure of degeneration of the ellipsoid is

$$\gamma = l_x + l_y - l_z + h, \tag{5}$$

with h the height of the tree. Furthermore, if one of the terms in (4) has a non-zero imaginary part – which means that an ellipsoid is not suitable for the point cloud, we declare the ellipsoid as degenerate as well; for reasons of visualization, a cone is then estimated from (3) by setting $q_2 = q_5 = q_7 = 0$. In the case of a degenerate ellipsoid or a cone, we model the tree by a coniferous, otherwise by a deciduous tree.

7. PROPOSED PROCEDURE

Now we are ready to summarize all the tools of the previous sections into a procedure for individual tree detection, see Algorithm 1 below. We first collect our trees by two fast modules ($\mathcal{S}$ and $\mathcal{W}$), then by a sophisticated tool ($\mathcal{H}$), and finally by $\mathcal{P}$. For this last module, it does not make a difference whether a group of pixels of class *tree* could be identified as a tree or not. We suppress for every module all the trees detected by its predecessor. If there are no coarse errors in the classification, Steps 5-7 of the algorithm are not necessary. In the case we wish to detect anomalies, we can apply $\mathcal{W}$ in Step 1 only if trees grow scarcely enough upon the hills; otherwise the module $\mathcal{H}$ or watershed with markups should be chosen. One could think that once hills were detected and the DTM was modified, it should also be necessary to re-calculate the class *tree* and the components $\mathcal{W}$ and $\mathcal{H}$ at least in the modified areas. But since DTM does not affect the high-frequent oscillation of the elevation map, filtering the previously obtained components in hilly areas by their relative elevations is already sufficient. Thus, there is almost no loss of computation time caused by Step 7.

Algorithm 1. Overview of the proposed procedure. See text for details.

```
Step 1.    Perform $\mathcal{S}$ on class tree
Step 2.    Perform $\mathcal{W}$ on NDSM $\cap$ class tree
Step 3.    Collect big components $\rightarrow$ hill hypotheses
for every big component
  Step 4.  Use $\mathcal{H}$ to delete points upon hypothesis
  Step 5.  Perform assessment
  if hill
    Step 6. Geometric reconstruction of hill surface
  end if
end for
Step 7.    Update DTM and NDSM
Step 8.    Perform $\mathcal{P}$ and filtering
Step 9.    Classification of clearly identified trees
```

In the output model, the (x, y) position of the trunk is given by the highest point of the component while the z coordinate is given by the DTM value. The diameter of the circular region is calculated from the number of pixels making up the component and the image resolution. Tree heights are extracted from the NDSM. Finally, the type of a clearly identified tree is assigned as described in Sec. 6, otherwise it is chosen randomly.

8. RESULTS

The dataset considered here represents the test and training site of the Technical Center in *Meppen*, Northern Germany. A DSM sampled with a resolution of 0.5m from an ALS point cloud together with a digital orthophoto is the input of our procedure. AS ALS, Riegl VQ-580 was used with 300 kHz measurement rate and 60° field of view leading to ca. 4 pts/m^2. Optical images were collected with a hyperspectral camera, AISA Eagle II and directly geo-referenced using inertial navigation data and an automatic boresight calibration method. There are two reasons for choosing this dataset: First, it has a large areal of around 2.5×1.5 km^2 with many trees of different types and difficult situations for their detection and classification. The second reason is that in selected areas, it was possible to obtain the ground truth for evaluation of our results. In the following three subsection, we present our results for individual tree detection, correction of DTM by detecting and modeling hills, as well as classification of trees.

8.1 Detection of Trees

Since the modules $\mathcal{S}$ and $\mathcal{P}$ are rather trivial ones, the content of this section shall concentrate on the comparison between the modules $\mathcal{W}$ and $\mathcal{H}$ in groups of trees and in forest regions. We show in Fig. 3 four situations and six tree groups with available ground truth results, which will show the differences between these tools together with the their advantages and disadvantages. The images 1-9 of Fig. 3 make it clear that mostly, the module $\mathcal{W}$ combined with classification results yields a correct result on individual tree detection. The reason is that mostly two assumptions hold: 1) in the slightly smoothed airborne laser data, a clear local maximum of the elevation map corresponds to an individual tree and 2) the region border is given either by the shedding line between two components or by the classification result. In the first two rows, groups of trees G1, G3, G4, were successfully delineated and the configurations correspond to the reality.

However, if the first assumption does not hold, some trees consist of more than one crown and belong to several components. For example, see a tree in the bottom middle of Fig. 3, 7-9. This problem has been noted, to a larger extent, in the high-resolution datasets from multi-view configurations of images. In photogrammetrically generated point clouds of a high resolution (100 points per m^2), see, for instance, (Rothermel et al., 2014), a footprint of a big tree usually has many local maxima. To solve this problem, we will implement the multi-scale method of (Persson, 2001) in our future work. Cases when the second assumption fails concern the trees that are either occluded completely by other trees – which happens for groups G2 and G5 of Fig. 3) – or for which the local maxima have been over-smoothed. For instance, in the top of images 7-9, Fig. 3, three trees near a building wall were accurately cut; hence, $\mathcal{W}$ found just one component with a suspiciously large eccentricity. Hence, this component has been processed with $\mathcal{H}$ and two of three trees were detected, see Fig. 3, image 9. In this special case, also the last tree could be detected within the primitive collection $\mathcal{P}$. In Fig. 3, bottom row, we see how big components are processed with $\mathcal{H}$ and single trees are selected in a repletion of nested regions. The remaining green components are then analyzed whether they are hills. At this point, we arrive at the next section about the detection and reconstruction of hills. It remains to mention that the results presented in this section are superior to those of (Reitberger, 2010) (< 65%), (Persson, 2001) (< 68%) and are slightly below those of the local maximum refinement and delineation algorithm (LMRDA) due to (Pouliot et al., 2002) with its best result of 91% of per-pixel detection; though the number of trees in the sample, the complexity of situation, and the measure for assessment of accuracy certainly vary in all contributions.

8.2 Detection and Reconstruction of Hills

To perform the verification of small hills and earth walls, we first need a reliable ground truth. To obtain it, we applied a pixel-wise supervised classification using hyper-spectral and elevation-based data as well as a support vector machine (Chang and Lin, 2011). The data and the method could seem like an overkill for detection of earth walls, but are necessary for verification. The spectral bands are directly used as hyperspectral features excluding the ones with bad signal-to-noise ratio. Additionally, 20 morphological profiles are used as elevation-based features (Benediktsson et al., 2005). To combine these features in a meaningful way, we normalize them by removing the mean and dividing them by their standard deviation. For several classes like *building*, *tree*, *grass* and the target class *hill*, between 400 and 700 pixels are used to train the classifier. Post classification is used to reduce the false alarm rate. Connected components are identified and

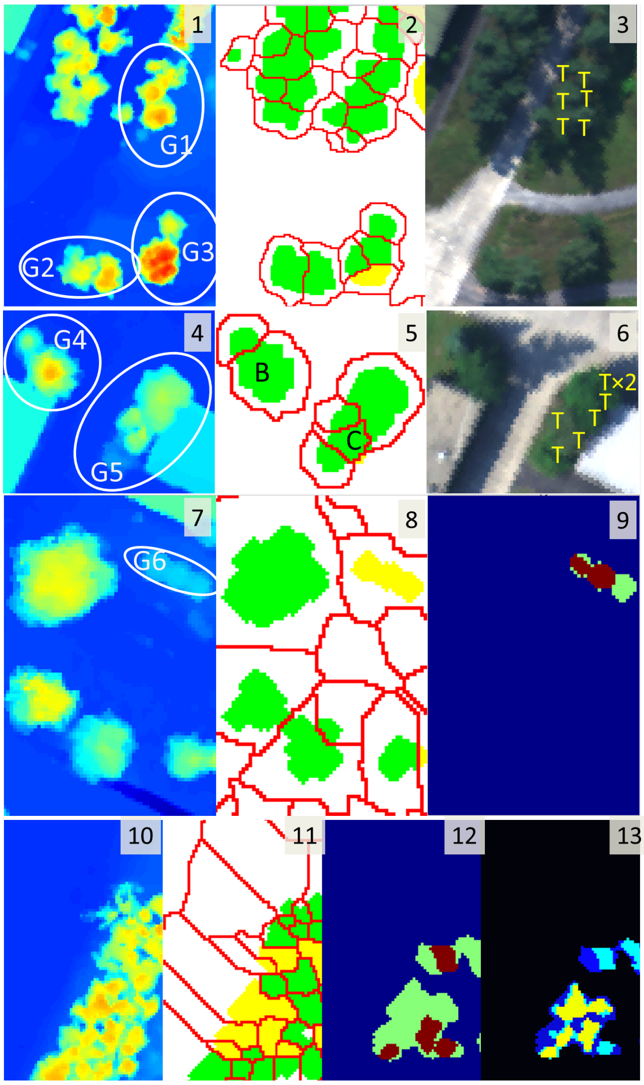

Figure 3: Detection of individual trees for four subsets of the dataset *Meppen*. The corresponding fragments of DSM and watershed components are shown in images in the first two columns while the corresponding fragments of the orthophoto are presented together with the approximate tree positions in images 3, 6. The watershed components are separated from each other by red lines. The green and yellow components demonstrate, respectively, the tree-like and non-tree-like intersections of classification results and watershed components. The large non-tree-like regions are subject to hotspots detection (images 9 and 11), of which tree-like and non-tree-like hotspots are shown in red and green colors, respectively. Image 13 shows the nested components. Trees B and C in image 5 are examples used in Sec. 8.3.

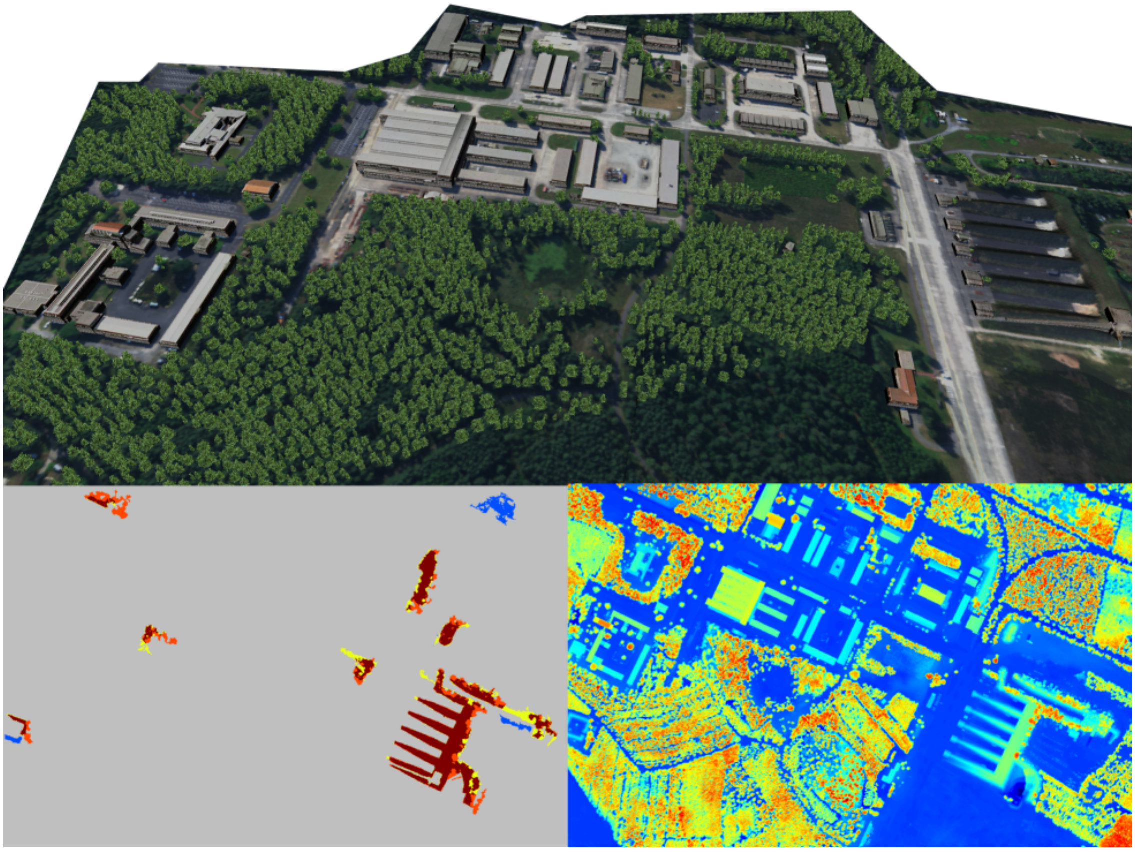

Figure 4: View of a urban terrain reconstruction result for the complete dataset. In the bottom part, on the left: A larger fragment with results for hills detection (see text for details); on the right: The corresponding fragment of the DSM.

isolated pixels misclassified as hills are removed. The overlap of the ground truth and our result is depicted in Fig. 4, bottom left. The blue color indicates the false negatives: hills in the ground truth but not detected by our procedure. There are three false negatives: One because of its area and two because of the planarity measure (see Sec. 5). Also, there were no false positives. There are ten true positives, such that the results for completeness and correctness are 76% and 100%, respectively. For true positives, we also showed the per-pixels detection results. By dark red, yellow, and orange colors, respectively, we specify the areas of accordance, missed part of a hill, and superfluous part of the hill (which is not a problem, since trees upon the hills are filtered out anyway).

We can conclude from the figure and from the explanations that detection of hills is a complicated job if they are occluded by trees. However, the numbers for completeness and correctness are encouraging; they can be improved in the future by adjusting thresholds or a reasonable integration of other features.

As for the hill reconstruction, the grid size for different hills has remained constant (around 1m), except that for reasons of speed and numerical precision, the number of grid points should remain below 100×100. The smoothness parameter for the functional in (Bulatov and Lavery, 2010, D'Errico, 2005) was chosen around 0.5. No big change of the results occur for moderate ($\pm 10\%$ to 20%) changes in these parameters. The main difference is caused by the choice of the reconstruction algorithm. The procedure based on work of (Bulatov and Lavery, 2010) is better suited for modeling regions of both slow and rapid smooth change than the procedure based on (D'Errico, 2005), at cost of computing time.

8.3 First Results on Classification of Trees

We selected 24 trees which were detected as individual trees by $\mathcal{S}$ and $\mathcal{W}$ and made following observations, which are also visualized in Table 2 and Fig. 5: With an exception of one deciduous tree (C) which is very cramped between the building wall and some other trees, the coniferous are more likely to be modeled by very narrow ellipses or cones. Several kinds of coniferous trees, however, exhibit a large value of the measure from (5). The problem could mostly be observed for trees detected with $\mathcal{W}$. Hence, one possible explanation could be that the delineation was not clean enough. However, it is obvious that obtaining reliable results for classification of trees by using only one characteristics – tree shape – is hardly possible.

9. CONCLUSIONS AND OUTLOOK

The work presented here is focused on detection of single trees in challenging scenarios. The procedure starts with the normalized DSM and a classification result, and goes the way over eventual correction of DTM, detection and, finally, classification of individual trees. The procedure is relatively fast because its most time-consuming modules (such as detection of hotspots and surface reconstruction) can be omitted or be performed in the areas of the image where it is really necessary. Additionally, other modules, such as fitting a conic into tree crown, are easily parallelizable. With respect to the performance evaluation, watershed transformation, already chosen by many authors, is a suitable tool to detect tree crown tips and separate trees from each other. However, unlike many related works, the *delineation* of trees is carried out a-priori, by using the classification result. As a consequence,

Table 1: Behavior for the modules $\mathcal{W}$ and $\mathcal{H}$ for six example clusters of trees. The second row shows number of trees in the group while the second row shows the complexity level of the situation (trees growing close to each other, under the crones of other trees etc.): Not very challenging (1), challenging (2), and very challenging (3). The lines 3-6 show the absolute and relative numbers of detected individual trees for the methods discussed above. Unfortunately, no ground truth was obtained for Fig. 3, 10-13. Also, delineation of some single trees was not recorded.

Cluster nr.	1	2	3	4	5	6	Σ
# Trees	6	3	4	2	7	3	25
# Complexity	2	3	1	3	3	3	-
# ($\mathcal{W}$)	6	2	3	2	4	1	18
In %	100	67	75	100	57	33	72
# ($\mathcal{W}+\mathcal{H}+\mathcal{P}$)	6	2	4	2	4	3	21
In %	100	67	100	100	57	100	84

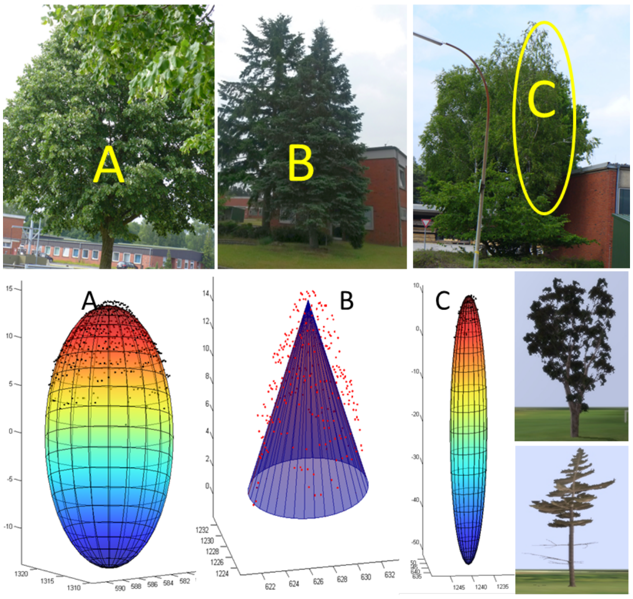

Figure 5: Top row: The first three images show selected trees which are approximated by conics as shown in the three images of the bottom row. Top right: Part of the DSM. Bottom right: Two prototypes for tree models.

Table 2: Preliminary results of tree classification

	dec. trees	conif. trees	Σ
dec. trees found by $\mathcal{S}/\mathcal{W}$	15 6/9	3 0/3	18
con. trees found by $\mathcal{S}/\mathcal{W}$	1 0/1	5 3/2	6
Σ	16	8	24

the situations where this result is not reliable should be recognized and corrected. Recognition (especially with respect to the component analysis based on hotspots detection with ISOL algorithm) and reconstruction of such anomalies make up the main contributions of this work. The hotspots detection is applied in order to identify trees upon hills and in other areas where watershed transform has failed. However, ISOL and MSER have both a number of empiric thresholds. To their further disadvantages with respect to watershed transformation, computation time and necessity to analyze nested components can be added.

We are aware that further theoretical concepts are necessary to analyze and to correct more general cases of misclassification that may theoretically occur (cavities instead of hills, trees growing upon building roofs, and many others) and we will consider it in our future work. Finally, with respect to simulations, even a few tree types with corresponding texture pictures aggravate the interaction if the total number of trees is very large. This is the reason why on the one hand, we want to improve and broaden the classification of trees, which at the moment is based merely on parameters of a conic approximating the point cloud, and on the other hand, we search for a memory-efficient representation of large forest regions in areas less relevant for simulation.

REFERENCES

Abdel-Aziz, Y. and Karara, H., 1971. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Proc. of Symp. on Close-Range Photogrammetry, American Society of Photogrammetry* pp. 1–18.

Anderer, C., Thönnessen, U., Carlsohn, M. F. and Klonz, A., 1989. Ein Bildsegmentierer für die echzeitnahe Verarbeitung. *DAGM-Symposium* pp. 380–384.

Andersen, H.-E., Reutebuch, S. E. and Schreuder, G. F., 2001. Automated individual tree measurement through morphological analysis of a lidar-based canopy surface model. In: *Proc. International Precision Forestry Symposium, Seattle, WA, USA*, pp. 11–21.

Bechtel, B., 2007. Objektextraktion von Bäumen aus Luftbildern: Vergleich und Steuerung von Segmentierungsverfahren zur Vorbereitung eines Expertensystems. PhD thesis, Universität Hamburg, Hamburg, Germany.

Benediktsson, J. A., Palmason, J. A. and Sveinsson, J. R., 2005. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing* 43(3), pp. 480–491.

Bulatov, D. and Lavery, J. E., 2010. Reconstruction and texturing of 3D urban terrain from uncalibrated monocular images using L_1 splines. *Photogrammetric Engineering & Remote Sensing* 76(4), pp. 439–449.

Bulatov, D., Häufel, G., Meidow, J., Pohl, M., Solbrig, P. and Wernerus, P., 2014. Context-based automatic reconstruction and texturing of 3D urban terrain for quick-response tasks. *ISPRS Journal of Photogrammetry and Remote Sensing* 93, pp. 157 – 170.

Chang, C.-C. and Lin, C.-J., 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3), pp. 1–27.

Dalponte, M., Ørka, H. O., Ene, L. T., Gobakken, T. and Næsset, E., 2014. Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data. *Remote Sensing of Environment* 140, pp. 306–317.

D'Errico, J., 2005. Surface fitting using gridfit. *MATLAB central file exchange*.

Eysn, L., Hollaus, M., Lindberg, E., Berger, F., Monnet, J.-M., Dalponte, M., Kobal, M., Pellegrini, M., Lingua, E. and Mongus, D., 2015. A benchmark of LiDAR-based single tree detection methods using heterogeneous forest data from the Alpine space. *Forests* 6(5), pp. 1721–1747.

Hirschmüller, H., 2008. Stereo processing by semi-global matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.

Horn, H. S., 1971. *The adaptive geometry of trees*. Vol. 3, Princeton University Press.

Kaartinen, H., Hyyppä, J., Yu, X., Vastaranta, M., Hyyppä, H., Kukko, A., Holopainen, M., Heipke, C., Hirschmugl, M., Morsdorf, F. et al., 2012. An international comparison of individual tree detection and extraction using airborne laser scanning. *Remote Sensing* 4(4), pp. 950–974.

Koch, B., Heyder, U. and Weinacker, H., 2006. Detection of individual tree crowns in airborne lidar data. *Photogrammetric Engineering & Remote Sensing* 72(4), pp. 357–363.

Lafarge, F. and Mallet, C., 2012. Creating large-scale city models from 3D-point clouds: a robust approach with hybrid representation. *International Journal of Computer Vision* 99(1), pp. 69–85.

Matas, J., Chum, O., Urban, M. and Pajdla, T., 2002. Robust wide baseline stereo from maximally stable extremal regions. *Proc. of British Machine Vision Conference* pp. 761–767.

Persson, A., 2001. Extraction of individual trees using laser radar data. *Department of Signals and Systems, Chalmers University of Technology, Göteborg, Sweden*.

Pollock, R. J., 1996. The automatic recognition of individual trees in aerial images of forests based on a synthetic tree crown image model. PhD thesis, Concordia University, Montreal, Canada.

Pouliot, D., King, D., Bell, F. and Pitt, D., 2002. Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration. *Remote Sensing of Environment* 82(2), pp. 322–334.

Reitberger, J., 2010. 3D-Segmentierung von Einzelbäumen und Baumartenklassifikation aus Daten flugzeuggetragener Full Waveform Laserscanner. PhD thesis, TU Munich, Munich, Germany.

Rothermel, M., Haala, N., Wenzel, K. and Bulatov, D., 2014. Fast and robust generation of semantic urban terrain models from UAV video streams. In: *Proc. of 22nd International Conference on Pattern Recognition, ICPR*, pp. 592–597.

Secord, J. and Zakhor, A., 2007. Tree detection in urban regions using aerial lidar and image data. *IEEE Geoscience and Remote Sensing Letters* 4(2), pp. 196–200.

Straub, B.-M., 2003. Automatic extraction of trees from aerial images and surface models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 34(3/W8), pp. 157–166.

Wang, L., Gong, P. and Biging, G. S., 2004. Individual tree-crown delineation and treetop detection in high-spatial-resolution aerial imagery. *Photogrammetric Engineering & Remote Sensing* 70(3), pp. 351–357.