Zuerst ersch. in : Proceedings of SPIE : Signal Processing, Sensor Fusion, and Target Recognition XXII ; 29 April - 2 May 2013, Baltimore, Maryland, United States / Ivan Kadar (ed.). - Bellingham, Wash. : SPIE, 2013 - Art. 874512. - (Proceedings of SPIE ; 8745). - ISBN 978-0-8194-9536-5
http://dx.doi.org/10.1117/12.2015681

Real-time classification of ground from LIDAR data for helicopter navigation

Ferdinand Eisenkeil*[a], Tobias Schafhitzel[b], Uwe Kühne[b], Oliver Deussen[a]
[a]University of Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany +49 7531880; [b]Cassidian, Claude-Dornier-Straße, 88090 Immenstaad, Germany +49 7545800

ABSTRACT

Helicopter pilots often have to deal with bad weather conditions and degraded views. Such situations may decrease the pilots' situational awareness significantly. The worst-case scenario would be a complete loss of visual reference during an off-field landing due to brownout or white out. In order to increase the pilots' situational awareness, helicopters nowadays are equipped with different sensors that are used to gather information about the terrain ahead of the helicopter. Synthetic vision systems are used to capture and classify sensor data and to visualize them on multi-functional displays or pilot's head up displays. This requires the input data to be a reliably classified into obstacles and ground.

In this paper, we present a regularization-based terrain classifier. Regularization is a popular segmentation method in computer vision and used in active contours. For a real-time application scenario with LIDAR data, we developed an optimization that uses different levels of detail depending on the accuracy of the sensor. After a preprocessing step where points are removed that cannot be ground, the method fits a shape underneath the recorded point cloud. Once this shape is calculated, the points below this shape can be distinguished from elevated objects and are classified as ground. Finally, we demonstrate the quality of our segmentation approach by its application on operational flight recordings. This method builds a part of an entire synthetic vision processing chain, where the classified points are used to support the generation of a real-time synthetic view of the terrain as an assistance tool for the helicopter pilot.

Keywords: Ground detection, LIDAR, situational awareness, active contours, real-time applications, level of detail

1. INTRODUCTION

Helicopter pilots encounter numerous perceptional problems during a flight mission. Landings in environments such as desserts or snowfields are problematic due to dispersed dust or snow when the helicopter approaches the ground. Other hazardous situations are night flights, which allow only for degraded visibility. An additional aid in such situations are modern sensor systems that support navigation by providing additional visual cues in the cockpit or the pilot's head-up display. To generate such cues, sensor data must be classified. State of the art systems for improving the situational awareness of the pilot are Lidar (Light detection and ranging) systems. Such systems emit light pulses to measure the time of flight of reflected light [25]. The scanned range values are recorded in a regular two-dimensional projection plane and thus a visualization possibility is a depth image seen from the helicopter. The range values can be transformed into a global coordinate system when the position and orientation of the helicopter is known exactly.

In addition, navigational values such as flight altitude, speed, and direction are captured and can also be taken into account. A typical system delivers a point cloud with approximately 10-20.000 measured points every 300 milliseconds. The results are shown at the depth image 1a), the corresponding point cloud 1b), and the classification of a deployed operational ground classification in 1c).

*ferdinand.eisenkeil@uni konstanz.de; phone +49 (0) 7531 88 4749; fax +49 (0) 7531 88 4715; www.uni konstanz.de

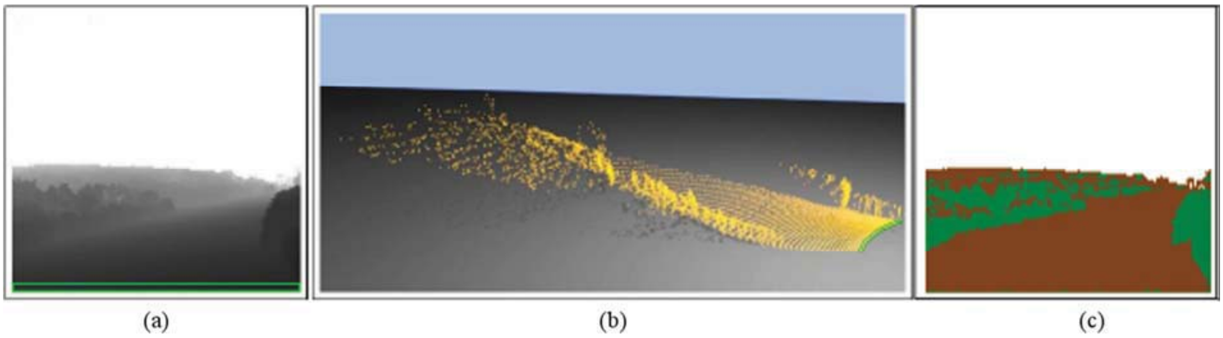

Figure 1. (a) Depth image: Each range value recorded by a Lidar sensor is stored in a pixel and coded as grey value. (b) Range values are transformed to 3D coordinates. The camera perspective differs from (a) to (b), the green box indicates corresponding pixels. (c) Classification result of the operational system. Unclassified pixels are green while ground pixels are brown.

In a next step, the data points are used to generate surface and terrain models. Surface models contain all measured data elements such as ground, obstacles, and vegetation. These models are used to determine potential collision points without differentiating between various types of obstacles.

In contrast, the focus of our "Advanced Synthetic Vision System" (ASVS) is on terrain models. Such models are used to prepare the collected data for obstacle segmentation. Measured points that are classified not to belong to the ground must be analyzed for being potentially dangerous, examples are power lines, man-made objects and trees. The "Helicopter Terrain Awareness and Warning System" [6] describes how obstacles from segmented point clouds are converted into appropriate visualizations for the pilots.

In our system, two separated hardware devices process the obtained sensor data, firstly the sensor in which the data is recorded and classified. Secondly, a mission computer running our ASVS that is not only used for generating a virtual representation of the environment, but also for detecting additional man-made objects such as buildings or power lines.

In the paper at hand, we propose a new method, implemented as part of our ASVS, for the classification of ground points (highlighted in dashed lines with white background in Figure 2). It is based on active contours and allows an adaptive formation of the ground while the data is collected. An efficient implementation of active contours helps us to deal with the real-time requirements during the flight.

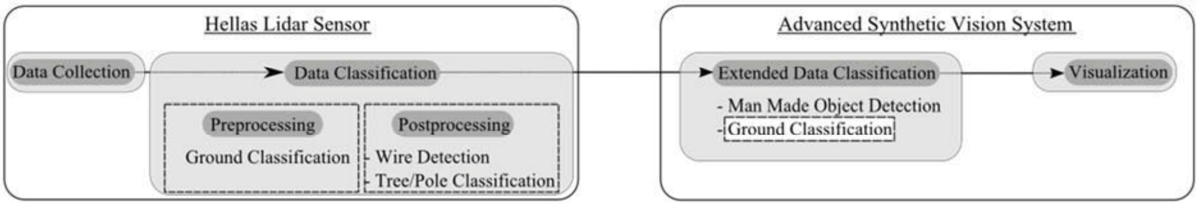

Figure 2. Processing chain from collecting data by a Lidar sensor up to visualization of classification results. The highlighted element in dashed lines with white background shows our approach method.

2. RELATED WORK

To classify Lidar point clouds into ground points and non-ground points multiple methods exist. In related papers, triangulation methods combine Lidar data with bird's eye views or ground based Lidar data [7]. A number of filter operators are applied to pre-process the data (cf. [20] for an overview). In this survey, it is distinguished between classification methods based on segmentation, morphological filters, interpolation, Triangular Irregular Networks (TINs), and active contours.

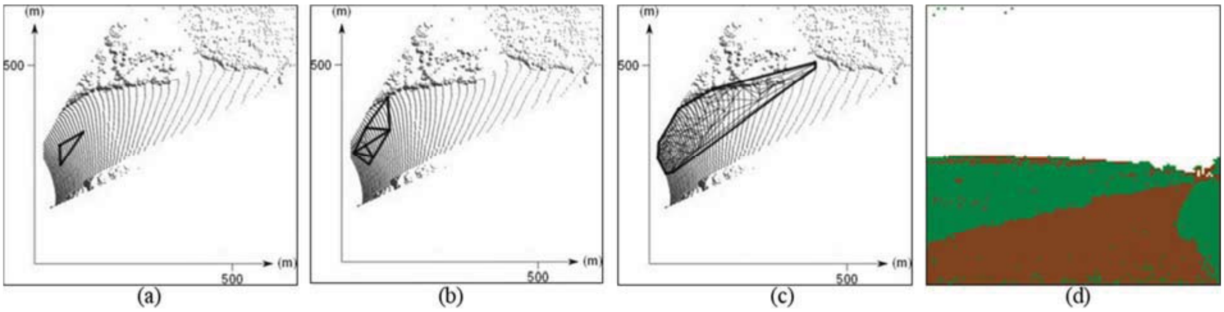

Figure 3. Evolving triangulation: (a) the three points with the lowest elevation form the start triangle. (b) State after a few insertions of points with the next higher elevation. The start triangle is subdivided and extended. (c) A mesh modeling a part of the terrain under the point cloud becomes visible. (d) The result on sensor data by the triangulation method is comparable to the operational classification method with more unclosed ground area in Figure 1(c).

2.1 Segmentation methods

The first method describes how to find homogenous regions in unordered airborne scanned point clouds without the need of meshing. Lidar points are clustered and analyzed in their local neighborhood. Different classes have a typical statistical behavior for their elevation and slope [10]. Rabbani et al. use a smoothness-constrained segmentation to fit planes onto clustered parts of the point cloud. This method results in shapes that are oriented arbitrary in a 3D feature space [22]. Another method to segment areas in airborne scanned Lidar data is based on the elevation of each data point. This implies that the ground must be relatively flat [16]. Also by using a region growing segmentation data, filtering is possible. Neighboring points belong to the same cluster if these fulfill criterions like small pair wise distance and similarity of normal vectors [24].

2.2 Morphological filters

To find a digital terrain model, morphological filters are a common method. As proposed by Arefi and Hahn, this method describes a dual rank filter based on dilation and erosion. The method is applicable to airborne laser data arranged in a grayscale depth image [17]. Arefi's and Hahn's method is also based on dilation with different window sizes to increase the ability to detect objects in different sizes [1]. These methods can be extended by using progressive window sizes changes for the dilatation and erosion filters [27].

2.3 Interpolation methods

Based on a linear least-square interpolation, Kraus [13][14] presented an iterative approach with adaptive weight functions. Schickler and Thorpe tested this filter in a mapping project [23] in applications with both forests and break lines [14], and build-up areas [4], and additionally improved the filter for applications in forest areas [14]. These methods are limited in data with rough terrain and slopes on the surface [19]. Another method is based on a facet model [28].

2.4 Deployed ground segmentation during flight

Additionally there is a method to deploy ground segmentation for dividing ground points from elevated data for helicopter flights. This method is based on a Laplacian pyramid [31] and its classification result is shown in Figure 1(c). The ground points found by this method are not used to generate a realistic terrain model. Its main objective is to find the obstacles that are above the ground, such as power lines or trees. For this purpose, the terrain includes lower vegetation or small rocks in addition to the actual ground. This method is used as a comparison for our algorithm.

2.5 Iterative Triangulation Methods

A common method for classification without using databases with already known terrain information is an iterative triangulation procedure [2][3][12][18]. As a reference to our proposed method, we implemented a variant of Peter Axelson's system [3] where Triangular Irregular Networks are used for classification. The measured points are sorted according to their height. A first triangle is constructed by the three lowest points; by sorting in new points a triangulation is created using Delaunay triangulation. For each new point the nearest neighbor vertex of the triangulation is found and the angle α between triangle and its normal is calculated. If α is greater than a threshold , the point is classified as ground and inserted into the triangulation. A point not classified as ground, is rejected. To generate the TIN, we used the Delaunay triangulation implementation in the OpenCV library [5]. Figure 3 shows how the TIN evolves for a set of Lidar points. To generate classification results for comparison, we implemented this triangulation method in addition to the deployed ground segmentation algorithm for Lidar data.

2.6 Active Contour Methods

Elmqvist et al. [8][9] propose an approach for generating terrain models based on active contours. Such contours are used to approximate a shape of an object of which only noisy points are known. The method contains a convolution as an iterative process in each step. Consequently, the method is not very efficient to generate a terrain-based classificator. Since time is critical, we extended this method in order to be more efficient. This will be described in the next session.

3. ACTIVE SURFACES FOR GROUND GENERATION

An active couture model is a concept for fitting the shape of objects in 2D images and 3D scenes to parametric curves or surfaces. An active contour or surface is influenced by internal forces, e.g. smoothness, and external forces computed from the underlying data. If the external force is formed by the gradient of an image, the contour is drawn in the direction of lines and edges. When the contour reaches a position where its energy is minimal, it stabilizes and the result is taken. This minimization problem is solved by regularization for internal and external forces [15][26]. In order to evolve our method for real-time capability, we use a simplified version of active contours that only regularizes internal forces. We decided to use Elmqvist' method as basis for our algorithm, because most of the related work is used in Lidar data, obtained in birds eye perspective. In addition, none of the compared methods are designed to be used in real-time application where data changes frequently.

Given a data set provided by the sensor, we use the minimization to fit the ground surface from below to the data points. Figure 4 shows how such a contour evolves iteratively in 2D on a point distribution as it occurs in Lidar point clouds and how the classification due to this fitting should be.

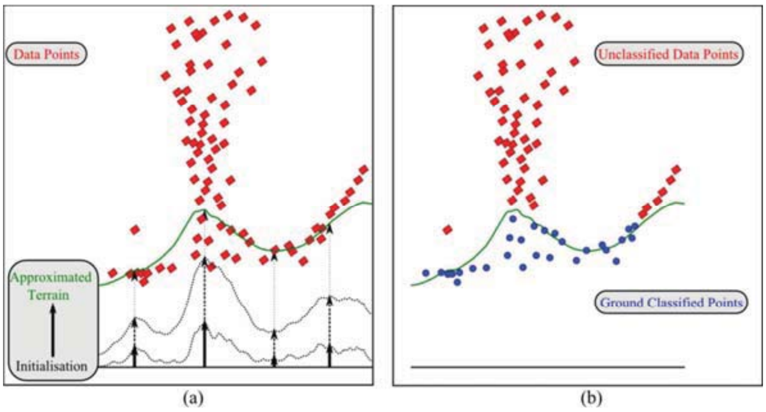

Figure 4. (a) Beginning with a horizontal line, the regularized contour is drawn upward until a stable solution is achieved. (b) Points below and slightly above the terrain contour will be classified as ground (blue dots).

3.1 Fitting the classificator to scanned points

The mathematical description of regularization is derived from the formulations of Terzopoulos' equations [11][21][29]. Suppose that the data set has been obtained by the Lidar sensor, then each of the data points is derived from a function that describes a surface model or rather the environment. A terrain-based classificator, described by a function , shall be recovered from this data. This is an ill-posed problem because the number of possible terrain models is infinite.

In order to select one particular solution, *a priori* knowledge like smoothness of the searched function is needed and is taking into account by using the first derivation as proposed for the surface model "membrane" in [30].

$$- \tag{1}$$

Additionally, the function f should be close to the measured data. Therefore, a distance measure is needed, in our case we choose the squared distance in height, otherwise the function would not be differentiable. Another reason is that the contour should lie in the minimal possible distance below or above to the point.

A solution to these two requirements is found by minimizing the following functional $H: C^1[\mathbb{R}^2, \mathbb{R}] \to \mathbb{R}$:

$$H[f] = \int E\big[f(x,y), f_x\,(x,y), f_y(x,y), x, y\big]dxdy \to min \tag{2}$$

where $H[f] = H_d[f,d] + \lambda H_p[f]$ with *Data Term* $H_d = \frac{1}{2}\sum_i \boldsymbol{\mathcal{K}}(x_i,y_i)[f(x_i,y_i) - d_i]^2$ and *Model Term* $H_p[f] = [D_1 f](x,y)$. The *regularization parameter* λ is a positive number that controls the tradeoff between *Data Term* and *Model Term*. The *Data Term* guarantees the closeness to measured data and the *Model Term* the smoothness. $\boldsymbol{\mathcal{K}}$ is a weighting function that indicates whether a data point was measured on position (x_i, y_i).

$H[f]$ is solved with Euler-Lagrange-Equation by building the partial derivations and results in a linear equation system:

$$\kappa(d - f) + \lambda\Delta f = 0 \tag{3}$$

By interpreting the input grid d and f as 1D vectors and setting up the *Model Term* Δ after solving $H[f]$ and the weighting function $\boldsymbol{\mathcal{K}}$ in the matrices M and K a linear equation system with a stationary solution instead of an iterative function is formulated:

$$\big(K\vec{d} - K\vec{f}\big) + \lambda M\vec{f} = 0 \Rightarrow \vec{f} = (K - \lambda M)\ ^{1}K\vec{d} \tag{4}$$

with M and K in the following forms:

$$M = \begin{pmatrix} -4 & 1 & 0 & \cdots & 1 & \cdots & \\ 1 & -4 & 1 & 0 & \cdots & 1 & \\ 0 & 1 & -4 & 1 & 0 & \cdots & 1 \\ & 0 & 1 & -4 & 1 & 0 & \\ 1 & \cdots & 0 & 1 & -4 & 1 & 0 \\ & 1 & \cdots & 0 & 1 & -4 & 1 \\ & & 1 & \cdots & 0 & 1 & \ddots \end{pmatrix}, K = \begin{pmatrix} \kappa(x_1,y_1) & \cdots & & & 0 \\ 0 & \kappa(x_2,y_2) & \cdots & & 0 \\ 0 & \cdots & \kappa(x_3,y_3) & \cdots & 0 \\ 0 & & \cdots & \ddots & 0 \\ 0 & & & \cdots & \kappa(x_n,y_n) \end{pmatrix} \tag{5}$$

Terzopoulos proposes the second derivation as smoothness prior for surface reconstruction. For our classificator both the first and the second derivation yield the same result (cf. Figure 6). The benefit in using the first derivation is the possibility to construct a matrix M for the *Model term* that simplifies the solution of the linear equation system with the conjugate gradient method. This approach is faster and more accurate as the iterative approximation shown in Chapter 3.

The linear equation system is solved for a regular grid of points. If data points were collected in the vicinity of this grid point, the height is determined (see below) and the weighting function $\boldsymbol{\mathcal{K}}$ is set to 1 for this position, otherwise it is 0. The initialization for the elevation of the grid points is equal to the smallest elevation value $\min(\{z_i\}_{i=0}^{N\ 1})$ in the set d. The grid dimension of the terrain model that is also the domain of a function $f: \mathbb{R}^2 \to \mathbb{R}$ is based on the spatial dimension of the point cloud. If two or more data points of the sensor fall into the same grid cell, the one with the lower elevation is kept for computing the elevation of the cell.

3.2 Classification procedure

Having a ground layer available, we are able to classify the measured points as belonging to the ground or to artificial objects. We determine the orthogonal distance of all points to the terrain-based classificator. Points with a lower elevation than this model are assumed to belong to the ground, also points above the model if their distance is within a given threshold.

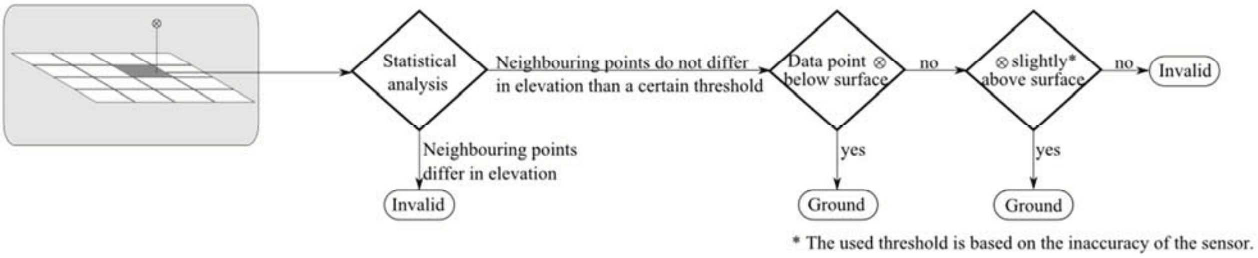
Statistical analysis
Neighbouring points do not differ in elevation than a certain threshold
Data point ⊗ below surface
no
⊗ slightly* above surface
no
Invalid
Neighbouring points differ in elevation
Invalid
yes
Ground
yes
Ground
* The used threshold is based on the inaccuracy of the sensor.

Figure 5. Decision chain if a point ⊗ can be classified as ground: The first test is part of the progressive preprocessing. If all the points that match a cell in the classification membrane grid have a large variance in their elevation, the point can impossibly be a ground point. After this step, ⊗ is classified as ground if ⊗ lies below or slightly above the cell. The maximal distance ⊗ can have above the classificator to be a ground pixel is based on the sensor inaccuracy.

For the classification of a data point, we identify the grid cell on the classifier membrane matching points' and coordinates. If the elevation of the grid cell is higher than the elevation of the point, it is classified as ground. Based on the statistical inaccuracy of height values of each pixel, also points slightly above the classifier are tagged as ground. The complete process is given in Figure 5.

3.3 Progressive Runtime Optimization with different Levels of Details

In practice, avionic computers are limited in their performance. Therefore, we propose an optimization to gain real-time behavior of the ground segmentation. The minimization procedure is limited in speed because of solving the linear equation to find the function for every data set every 300 milliseconds. In order to increase the speed, we have to reduce the accuracy of the grid in dependency to their spatial distribution and quality. The goal is to find the best ratio between Level of Detail (LOD) expansion and computation speed and correctness of classification. Points far away from the helicopter are handled using a coarser grid (cf. Figure 6), points close to the helicopter using a smaller grid. The resulting terrain-based classificator is approximated in a progressive behavior.

In the classification of data recorded by a Lidar sensor mounted on a helicopter, we make a special statement: A spatial and temporal coherence between two sensor frames exists so that the terrain-based classificator of one frame provides a basis for the following frame. We perform a statistical analysis on all pixels in the area of a grid cell on the terrain-based classifier. This information is kept for the following frames and all pixels that are in the area of this grid cell not taken into account for solving the linear equation. This reduces the number of calculations to be done in the conjugate gradient method.

This assumption is valid because of the correlation between the speed of the helicopter and the frequency of recording environmental data. The position and the viewing direction of the helicopter does not change significantly in the short time slot of 300 ms. In the best case, the sensor delivers the same information as in the previous frame and no adaption is necessary. This describes a theoretical optimum if the helicopter does not move. In practice, the sensor is biased by noise and errors in the helicopters positioning system.

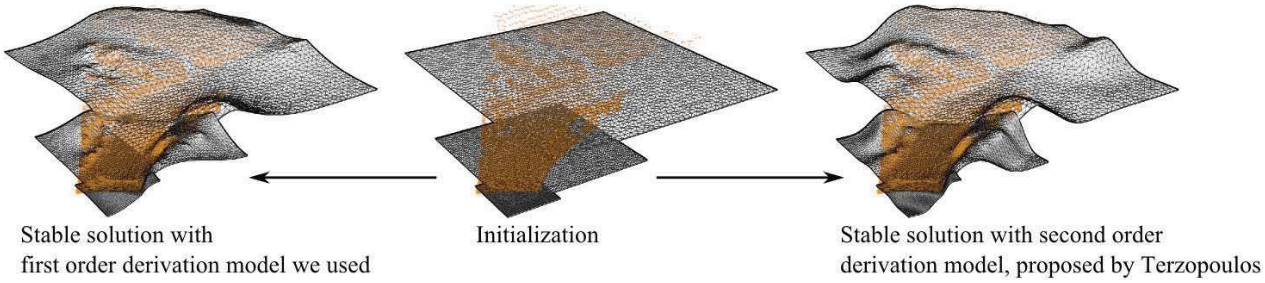
Stable solution with first order derivation model we used
Initialization
Stable solution with second order derivation model, proposed by Terzopoulos

Figure 6. The hierarchical approximated terrain based classificator in different LODs is calculated. This is possible in real time if the first order derivation model is used.

4. RESULTS AND EVALUATION

The triangulation method delivers terrain classification in real-time as well as the for operational issues established method, based on a laplacian pyramid. In both cases, the results satisfy the requirement to filter Lidar data for elevated points for further calculations like power line detection. Especially the operational method is usable for this purpose because irregular surfaces like forest areas are classified as ground. Our approach is slower than the two reference implementations, but its result is more usable for visualization aims and a candidate for a real-time application in avionic computers. The proposed method is an extension of the process Elmqvist et al. introduced in [8]. Our method has a few enhancements and differences to their method:

- The generation of the terrain-based classificator is calculated in a fast way, meaning that the calculation can be done between two sensor frames and therefore fast enough for terrain visualization in real-time.
- Progressive behavior: The spatial coherence between two sensor frames is used to reduce calculation complexity and duration.
- As an additional application to determinate the terrain features, Elmqvist et al. propose change detection, based on environmental alternation over time. In their paper, they use as an example the building of a wall. In a time slot of 300ms it will not happen, that the environment changes in a drastic way. The differences between two sensor frames occur in form of outliers or not yet sampled surfaces.

4.1 Evaluation of classification procedure

The usage of different LODs and discrete grids for fitting a classificator surface under the point cloud effects not only the time complexity but also the classification result. In Figure 7 and 8 the classification results for different methods for arbitrary 3D point clouds is shown. The operational Laplace-based method classifies in the example with the rift ground in the almost flat part and not in the valley of the rift. The triangulation and our method detect the valley as ground. As similar is observed in the arbitrary point cloud in the shape of the tree. In comparison to the triangulation method, our result delivers more ground pixels.

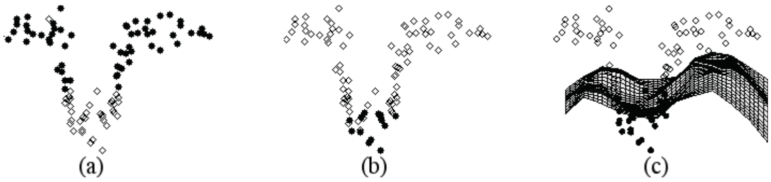

Figure 7. Classification results for three different methods an arbitrary point clouds in the shape of a rift. Black dots are ground pixels, boxes are unknown. (a) Operational Laplace based method. (b) Triangulation method. (c) Our method.

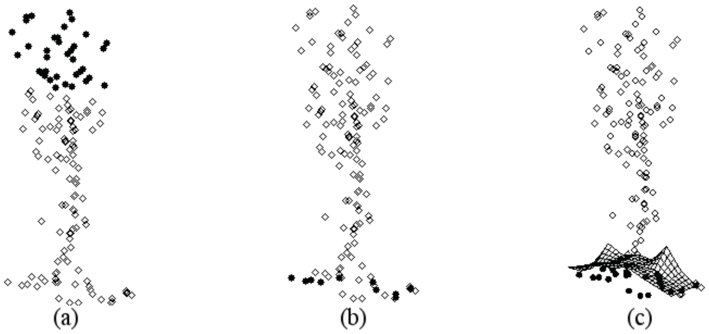

Figure 8. Classification results for three different methods an arbitrary point clouds in the shape of a tree. Black dots are ground pixels, boxes are unknown. (a) Operational Laplace based method. (b) Triangulation method. (c) Our method.

Figure 9 shows the classification result for our method in three different real and simulated cases. Cases (a) and (b) are synthetic sensor frames while (c) is recorded data from a real flight. The first column shows the range image, while the pictures in column 2 indicate the ground truth for (a) and (b). Column 3 shows the operational segmentation result and column 4 the result of the triangulation method. Our result is shown in column 5. The colors depict the class for each pixel. Ground pixels are dark and unknown pixels appear in light grey. Even more scenarios are shown in Appendix A.

In scenario (a) the top side and a part of the cube's side in the foreground is misclassified as ground. Especially for initial sensor frames, misclassifications cannot be excluded. The rest of the image is mostly classified correctly. As illustrated in Table 1, the number of correct classified pixels in scenario (a) is larger than 90% (Correct Ground + Correct Non-ground). For scenario (b) more than 50 % are misclassified ground pixels (cf. Table 1). This is due to the fact that only the point with the lowest elevation in a defined neighborhood is used for the construction of the classificator membrane. Many points lie above the classificator if the surface is not plain. Because our approach generates a smooth terrain classificator, pixels on the left side on the image lying on a ridge, are also not classified as ground. Nevertheless, both, the ground truth and our classification result when given as input for the ASVS, result in almost the same terrain visualization (cf. Figure 10).

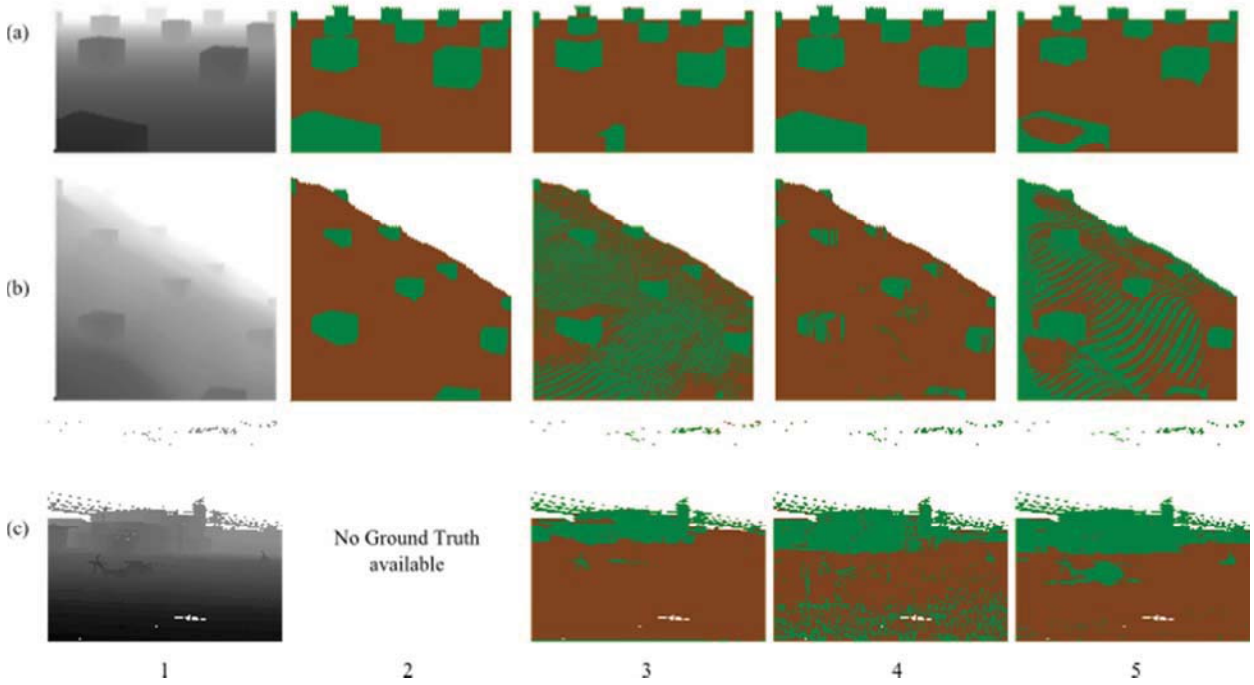

Figure 9. Three scenarios in different situations: (a) and (b) show synthetic sensor data with the range image in column 1 and the ground truth classification in column 2 (brown is ground and green is unknown). Column 3 shows the operational segmentation result and column 4 the result of the triangulation method. The classification result of our system is given in column 5. Line (c) shows real flight data without ground truth.

In comparison to the other methods, our method has view advantages on real data: The operational method misclassifies pixels near the real ground and all flat parts as ground. This behavior is demonstrated on the helicopter pixels in scenario (c, 3) and the top of the cubes in (a, 3). The triangulation method cannot deal with small changes in the slopes. Cubes on the hillside in scenario (b, 4) are partially misclassified. In addition, this method is noise sensitive (cf. Figure 9 (c, 4)).

Our method increases the classification quality of the operational classification in scenario (c). The differences between Figure 9 (c, 3) and Figure 9 (c, 5) shows that the helicopter as well as the lower parts of the buildings' walls in the background are correctly classified as non-ground. As demonstrated on the surface reconstruction behavior for scenery (b, 1) in Figure 9, the noisy misclassification of the ground in the foreground will not influence correct ground visualization in the ASVS. The discretization and the resolution of the terrain-based classifier has a negative impact on the classification result.

Table 1. Comparison between Ground Truth and our classification result. The amounts of pixels that are correct, respectively wrongly classified, are given in percent.

Classification	Synthetic scene with flat ground (Figure 9(a))	Synthetic scene with slope (Figure 9(b))
Correct ground	66,77 %	34,23 %
Correct Non-ground	27,44 %	10,34 %
False Ground	4,74 %	1,03 %
False Non-ground	1,05 %	54,40 %

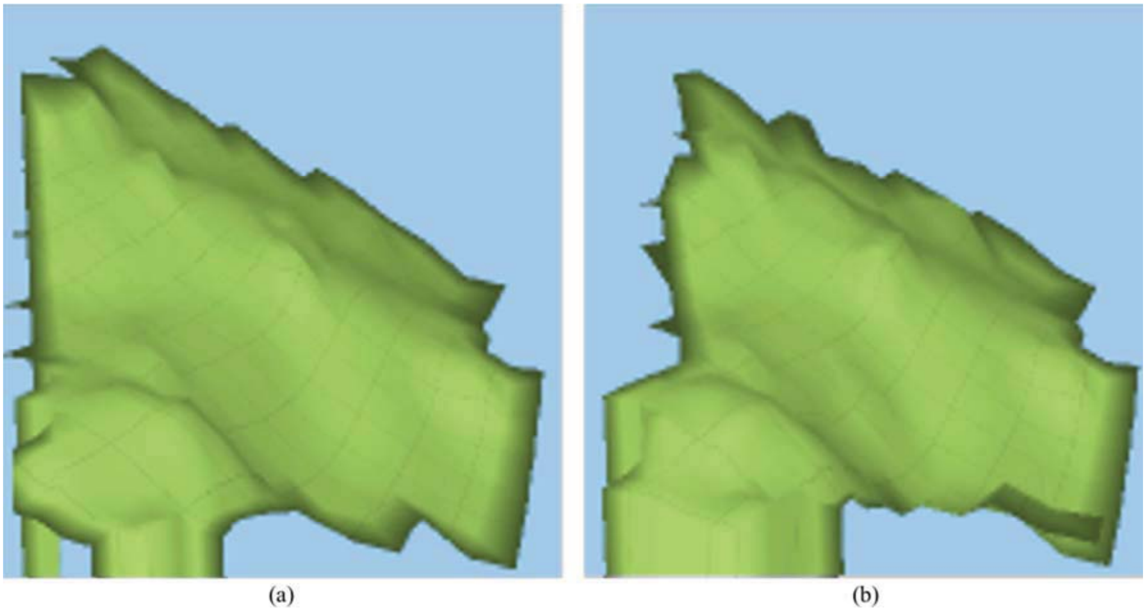

Figure 10. Terrain generated with the synthetic sensor frame shown in Figure 9(b): (a) is generated with Ground Truth and (b) by our method with the classified pixels. Both show almost the same terrain. The only exception is the boarders of the terrain. This behavior is induced by the numerical solution of the linear equation system.

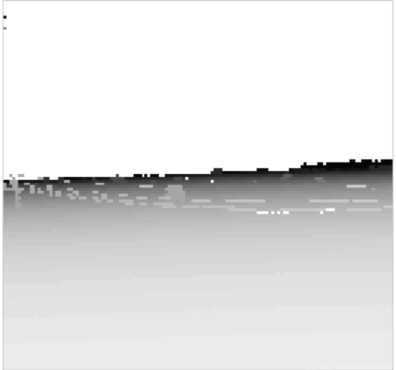

Figure 11. Real sensor data recorded on a flat field. This scenery is one of the scenarios used for testing calculation speed on a mission computer.

4.2 Evaluation of time complexity

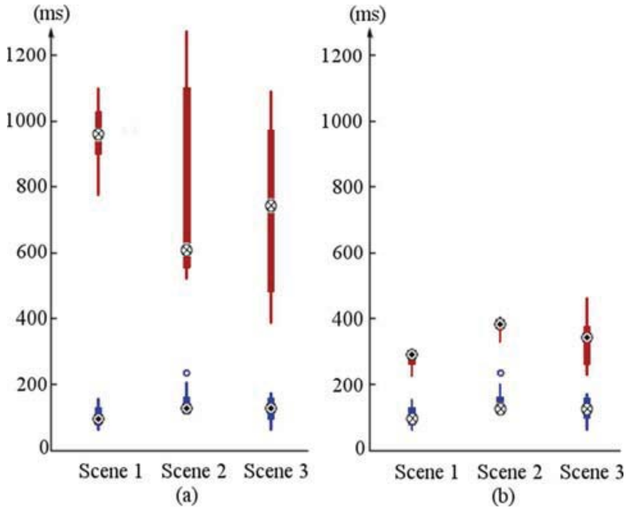

Figure 12. The box plots contain the computation time for 20 frames for each scenario between two sensor frames. (a) Calculation durations over three scenarios with (⦿) and without different LODs (⊗) on a PC. (b) Calculation duration with usage of different LODs on a PC (⊗) and on a mission computer (⦿).

To show the reducing of the computation complexity of Elmqvist et al. method with our approach with different LODs, we measured the calculation duration on a notebook (Quad core i5 CPU@2.4Ghz, 4GB RAM). For Elmqvist's reference implementation, we used a grid with the resolution of the lowest LOD in our method. As input served sensor frames in scenarios of different complexity. For this comparison, we analyzed 20 frames for every flight scene. The computation times as box plots are shown in Figure 12. In scenario 1, the helicopter flies over a valley with bushes and woods on the sides. Scenario 2 is a flat field, and scenario 3 is an airport with hangars and electrical towers with power lines. One frame of scenario 1, 2, and 3 each is given in Figure 1, Figure 11 and 9c). Our approach takes less than 300 ms in all three cases. The computation speed of solving the linear equation in Chapter 3.1 depends on the grid size. It is therefore more efficient to use small dimensions for high LODs. We also tested our implementation on a mission computer with the real-time operating system VxWorks (Version 5.5.2, CPU: 833 MHz, PCI Bus: 33 MHz). The calculation times are slightly higher than on a PC, but with reduction of calculations in higher distances, the classification method is also real-time capable on a mission computer. Based on the inaccuracy of the sensor in high distances over 500 meters, a ground classification might not be necessary.

5. CONCLUSIONS AND FUTURE WORK

In all of the presented methods, the classification generating terrain models can be computed for different purposes. The triangulation method and the Laplace based method deliver models that can be used as basis for obstacle segmentation. Both methods behave similar at classification in clearly elevated but almost homogeneous surfaces like bushes. In such regions ground is detected, which, although sufficiently allowing for preprocessing of obstacle classification, was not primarily conceptualized for visualization issues. In visualizations on basis of ground detected by the Laplace method, the pilot himself has to distinguish between ground and vegetation by comparing the real environment with the 3D visualization. In case of the triangulation, our method contains a major advantage: The cited procedure uses one reference triangle for classification. With the regularized grid, it is possible to classify a point with respect to the neighboring pixels because of the smooth classification membrane. Regularization, used in our approach, improves finding ground in Lidar data, but does not guarantee to find a completely correct terrain-based classificator. Especially in flights above forests, trees form an almost homogeneous surface and consequently lead to false classification.

Our approach delivers a more realistic ground classification for visualization purposes. Our classification result has the quality to facilitate a more realistic terrain reconstruction for common scenarios. Additionally, we demonstrated that our technique is real-time capable on mission computers within the time frame of 300ms. The classified points serve as input for an "Advanced Synthetic Vision System" in order to generate a terrain model. Figure 13 shows how a visualization of the environment would look like for a pilot. The dark areas are constructed by as ground classified data points. The lighter parts of the ground surface are loaded from a digital terrain database. In theory we showed that it is possible to perform all calculations and visualization in real-time.

Mission computers are limited in their performance, because different applications for pilot assistance run simultaneously and have to work as real-time applications. For the ASV application, with ground detection together with other parts as the obstacle detection and visualization, the requirement is to improve the developed calculation optimization to reduce the need for computation power. Consequently, for some of the calculations, e.g. the very time consuming solving of linear equations, a shader implementation on a GPU could be one possibility in future. Also a bilinear interpolation on the grid of the classificator will increase the number of correctly detected ground in areas with strong slopes as shown in Figure 9b). The usage of an efficient implemented bilinear or even higher order interpolations possibly on a GPU would result in better correct classification percentages than given in Table 1 and obtain the possibility to classify Lidar data for real-time visualization.

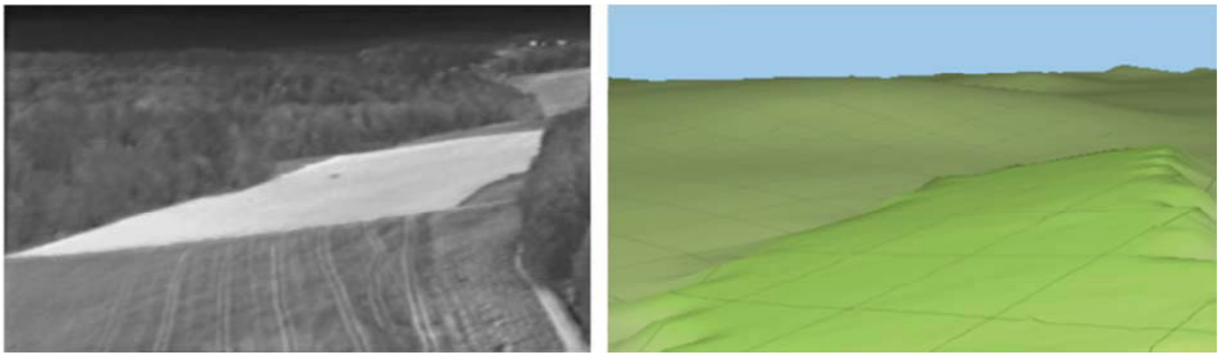

Figure 13. A synthetic view with our "Advanced Synthetic Vision System" is created based on the new classification method. The artificial generated scene correlates to the image taken with an infrared camera at the same time.

6. APPENDIX

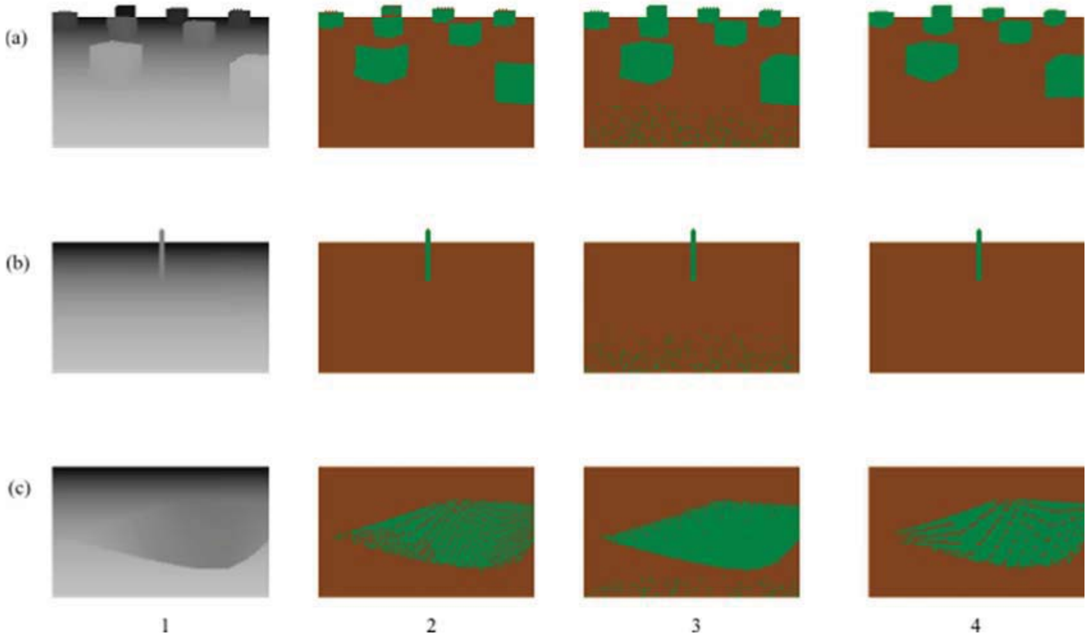

Figure 14. Classification results on perfect synthetic sensor images. Column 1 shows range images. Columns 2 4 show the classification result for operational, triangulation, and our method.

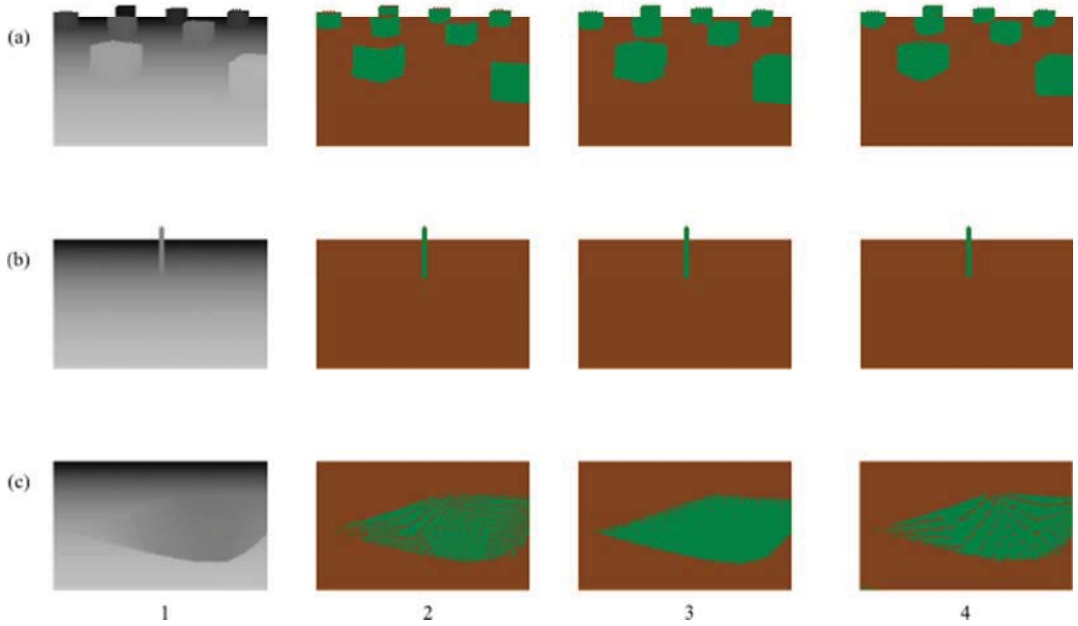

Figure 15. Classification results on noised synthetic sensor images. Noise is generated by adding normal distributed values between 1 and 1 meter on the distance value. Column 1 shows range images. The classification result for operational, triangulation and our method is shown in columns 2 4.

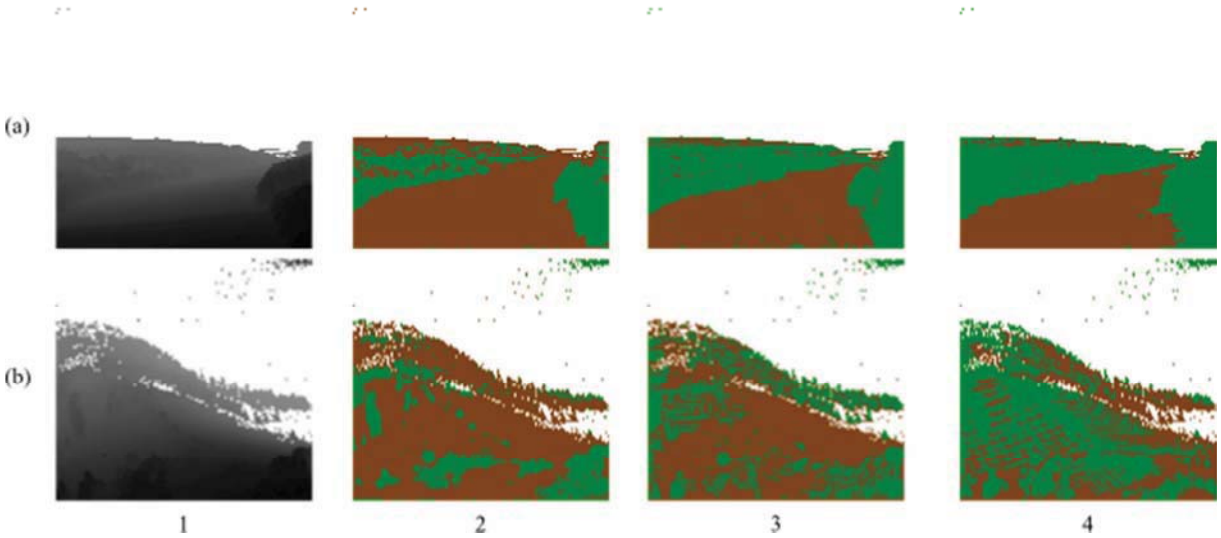

Figure 16. Classification results on real sensor images. Column 1 shows range images. Columns 2 4 show the classification result for operational, triangulation, and our method.

REFERENCES

[1] Arefi, H. and Hahn, M., "A morphological reconstruction algorithm for separating off-terrain points from terrain points in laser scanning data," Proc. ISPRS Workshop Laser Scanning (2005).

[2] Axelsson, P., "Processing of laser scanner data - algorithms and applications," ISPRS Journal of Photogrammetry and Remote Sensing 54, 138 147 (1999).

[3] Axelsson, P., "Dem generation from laser scanner data using adaptive tin models. International Archives of Photogrammetry and Remote Sensing," 110 117 (2000).

[4] Briese, C. and Pfeifer, N., "Airborne laser scanning and derivation of digital terrain models," Proc. of Fifth Conference on Optical 3-D Measurement Techniques, (2001).

[5] Bradski, G., "The OpenCV Library," Dr. Dobb's Journal of Software Tools, (2000).

[6] Cabler, S. J., "Technical Standard Order. Subject: Helicopter Terrain Awareness and Warning System," Tech. rep., Department of Transportation, (2008).

[7] Carlberg, M., Andrews, J., Gao, P., Zakhor, A., "Fast Surface Reconstruction and Segmentation with Ground-Based and Airborne LIDAR Range Data," Tech. Rep. UCB/EECS-2009-5, EECS Department, University of California, Berkeley, (2009).

[8] Elmqvist, M., Jungert, E., Lantz, F., Persson, Å., Söderman, U., "Terrain modelling and analysis using laser scanner data," International Archives of Photogrammetry and Remote Sensing, 22 24 (2001).

[9] Elmqvist, M., "Surface estimation from airborne laser scanner data using active shape models," Photogrammetric Computer Vision-ISPRS Commission III Symposium vol. XXXIV part A, 114 118.

[10] Filin, S., "Surface clustering from airborne laser scanning data," International Archives of Photogrammetry and Remote Sensing Vol. XXXII 3A, 119 124 (2002).

[11] Girosi, F., Jones, M., Poggio, T., "Regularization theory and neural networks architectures," Neural Computation 7, 219 269 (1995).

[12] Haugerud, R., A. and Harding, D., J., "Some algorithms for virtual deforestation (VDF) of lidar topographic survey data," International Archives of Photogrammetry and Remote Sensing 34 3/4, 211 217 (2001).

[13] Kraus, K. and Pfeifer, N., "Advanced dtm generation from lidar data," Proc. of the ISPRS Workshop on Land Surface Mapping and Characterization Using Laser Altimetry (2001).

[14] Kraus, K., Rieger, W., "Processing of laser scanning data for wooded areas," Photogrammetric Week, 221 231 (1999).

[15] Kass, M. Witkin, A., Terzopoulos, D., "Snakes: Active contour models," International Journal of Computer Vision 1, 321 331 (1988).

[16] Lohmann, P. and Jacobsen, K., "Segmented filtering of laser scanner dsms"

[17] Lohmann, P., Koch, A., Schaeffer, M., "Approaches to the filtering of laser scanner data," Int. Arch. Photogramm. Remote Sens. 33, 540 547 (2000).

[18] Lu, W., L., Littler, J., J., Sheffer, A., Fu, H., "Deforestation: Extracting 3d bare-earth surface from airborne lidar data," Canadian Conference on Computer and Robot Vision, 203 210 (2008).

[19] Lee, H., S. and Younan, N., H., "Dtm extraction of lidar returns via adaptive processing," IEEE T. Geoscience and Remote Sensing 41, 2063 2069 (2003).

[20] Meng, X., Currit, N., Zhao, K., "Ground filtering algorithms for airborne lidar data: A review of critical issues," Remote Sensing 2, 833 860 (2010).

[21] Poggio, T., Torre, V., Koch, C., "Computational vision and regularization theory." Nature 317, 314 319 (1985).

[22] Rabbani, T., Van Den Heuvel, F., A., Vosselman, G., "Segmentation of point clouds using smoothness constraint," Int. Arch. Photogramm. Remote Sens. 36, 248 253 (2006).

[23] Schickler, W., Thorpe, A., "Surface estimation based on lidar," Proc. of the ASPRS Annual Conference (2001).

[24] Tóvári, D. and Pfeifer, N., "Segmentation based robust interpolation - a new approach to laser data filtering," In ISPRS WG III/3 III/4 V/3 Workshop "Laser scanning", 12 14 (2005).

[25] Wehr, A. and Lohr, U., "Airborne laser scanning - an introduction and overview," ISPRS Journal of Photogrammetry and Remote Sensing 54, 68 82 (1999).

[26] Xu, C. and Prince, J., L., "Snakes, shapes, and gradient vector flow," IEEE Transactions on Image Processing 7, 359 369 (1998).

[27] Zhang, K., Chen, S., Whitman, D., Shzu, M., "A progressive morphological filter for removing nonground measurements from airborne lidar data," IEEE Trans. Geosci. Remote Sens. 41, 872 882 (2003).

[28] Zheng, S., Shi, W., Liu, J., Zhu, J., "Facet-based airborne light detection and ranging data filtering method," Opt. Eng. 46, (2007).

[29] Terzopoulos, D., "Multilevel Computational Processes for Visual Surface Reconstruction", Computer Vision, Graphics, and Image Processing 24, 52-96 (1983).

[30] Li, S., Z., [Markov random field modeling in computer vision], Springer-Verlag, London, (1995).

[31] Burt, P., J. and Adelson, E., H., "The Laplacian Pyramid as a Compact Image Code," IEEE Trans. on Communications 31, 532-540 (1981).