

Automatic forest road extraction from LiDAR data of mountainous areas

P. Even, P. Ngo

Université de Lorraine, CNRS, LORIA, Nancy, F-54000, France
(philippe.even, hoai-diem-phuc.ngo)@loria.fr

Abstract. In this paper, a framework is proposed to extract forest roads from LiDAR (Light Detection and Ranging) data in mountainous areas. For that purpose, an efficient and simple solution based on discrete geometry and mathematical morphology tools is proposed. The framework is composed of two steps: (i) detecting road candidates in DTM (Digital Terrain Model) views using a mathematical morphology filter and a fast blurred segment detector in order to select a set of road seeds; (ii) extracting road sections from the obtained seeds using only the raw LiDAR points to cope with DTM approximations. For the second step, a previous tool for fast extraction of linear structures directly from ground points was adapted to automatically process each seed. It first performs a recognition of the road structure under the seed. In case of success, the structure is tracked and extended as far as possible on each side of the segment before post-processing validation and cleaning. Experiments on real data over a wide mountain area (about 78 km²) have been conducted to validate the proposed method.

Keywords: LiDAR data · Road detection · Point cloud processing · DTM image analysis

1 Introduction

Road location and characterization are important information used for various purposes in forest management, such as wood harvesting, road construction and maintenance, transport, . . . In this context, airborne laser scanning, also called LiDAR for Light Detection And Ranging, is of great help to survey forested areas. It is a 3D acquisition technique based on the emission of a laser beam swept over the measured scene and on the detection of reflected signal from the surface. In forested areas, the received signal is composed of multiple echoes corresponding to the successive hit obstacles, from the forest canopy, down to lower vegetation levels and finally to the ground itself. A surface is interpolated through lower cloud points to classify them as ground and produce a *digital terrain model* (DTM). However, conifers are a strong obstacle. They impede the laser beam reaching the soil, and thus produce holes in the ground point distribution. Achieved point density is quite heterogeneous and this may cause large approximations in the delivered DTM.

In the literature, several methods have been developed for road extraction from LiDAR data [1,2,6,8,10,11]. Most of them rely on the shaded DTM analysis, using standard image processing tools. They show accurate enough for applications in urban and peri-urban areas in which road characteristics are quite regular and well contrasted. However, they are less efficient to detect forest roads in mountainous areas because of the strong geometric irregularities of these objects along their run. Their characteristics vary a lot depending on the terrain and also on the usage, from small narrow tracks up to very wide recent calibrated roads with the traffic line winding in the middle and vegetated sides for stocking timbers. The road surface is more or less rough depending on variable factors, such as erosion conditions or maintenance efforts. Lack of points due to LiDAR signal occlusions by a dense vegetation, especially nearby conifer plantations, drastically increases the detection difficulty. Some authors suggest to use raw data in complement to DTM analysis. Mostly signal intensity was used to recognize the road surface response [1,6], but this data depends strongly on local terrain features and parameters are difficult to set in practice. Raw altimetric information could help to better discriminate roads, but its processing is generally considered as too complicated.

To tackle this issue, we propose a framework based on efficient discrete geometry and mathematical morphology tools to automatically extract forest roads. First, the DTM is used to find relevant seeds, then only raw ground points are processed to extract road sections from each seed. By processing ground points, the new approach is more aware of point distribution in the raw data, and helps to overcome the limits of DTM interpolation. This second step relies on a recent approach for supervised extraction of linear structures from LiDAR raw data [4], that was designed for geomorphologists and archaeologists needs. In this approach, the human expertise is used to recognize roads in the DTM and provide correct initializations of the detection task. Fast responses ensure a good interaction level, that enables on-line corrections to quickly get long road sections with good accuracy. This is not any more possible in pure automatic context, so that adaptations were made to the initial step of this approach to accept less accurate road seeds. These adaptations and the seed selection procedure constitute the novelties put forward in this paper.

The method was tested on a LiDAR data set covering the Fossard mountain, in an area delimited by Remiremont, Docelles and Le Tholy (Vosges, France). Altitude ranges from 360 m up to 820 m. In such a wide region (about 78 km^2) with a dense network of forest roads, a manual extraction of all roads using the supervised approach would be too much time expensive and a tedious task.

The rest of the paper is organized as follows: Sec. 2 recalls the discrete geometry notions used in this work. Sec. 3 introduces the overall automatic extraction framework. It is composed of two steps, road seeds selection, then road extraction from each seed, that are described in Sec. 4 and Sec. 5. Held experiments are presented in Sec. 6, and Sec. 7 gives a conclusion and draws some perspectives.

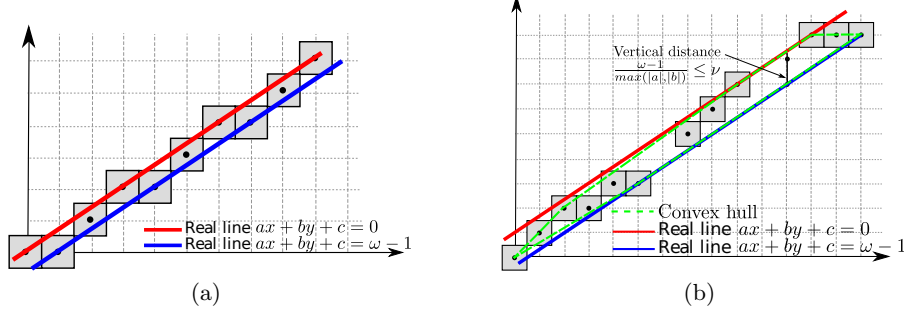


Fig. 1: (a) Naive digital straight line $\mathcal{L}(2, -3, 0, 3)$. (b) Blurred segment of assigned thickness $\varepsilon = 1.5$ and its optimal line $\mathcal{L}(3, -4, 3, 7)$.

2 Background notions

We recall briefly in this section several notions of digital geometry [7] used in this work. We refer the reader to the given references for more details.

A **digital straight line** $\mathcal{L}(a, b, c, \nu)$ is the set of discrete points $p = (x, y) \in \mathbb{Z}^2$ satisfying the inequalities:

$$0 \leq ax + by - c < \omega, \text{ with } (a, b, c, \omega) \in \mathbb{Z}^4 \quad (1)$$

Hereafter, we note $\vec{V}(\mathcal{L}) = (a, b)$ the director vector of \mathcal{L} , $w(\mathcal{L}) = \nu$ its arithmetical width, $h(\mathcal{L}) = c$ its shift to origin, and $p(\mathcal{L}) = \max(|a|, |b|)$ its period. When $\nu = p(\mathcal{L})$, then \mathcal{L} is the narrowest 8-connected line and is called a *naive line* (see Fig. 1(a)). The thickness $\mu = \frac{\omega-1}{p(\mathcal{L})}$ of \mathcal{L} is the minimum of the vertical and horizontal distances between lines $ax + by = c$ and $ax + by = c + \omega$.

A **digital straight segment** is a digital straight line restricted to $[x_{\min}, x_{\max}]$ interval if $|a| < |b|$, to $[y_{\min}, y_{\max}]$ interval otherwise.

A **blurred segment** [3] \mathcal{B} of assigned thickness ε is a set of points in \mathbb{Z}^2 that are all covered by a digital straight line \mathcal{L} of thickness $\mu \leq \varepsilon$ (see Fig. 1(b)). The covering digital line with minimal thickness is called the *optimal line* of \mathcal{B} . Blurred segments can be detected in linear time by a recognition algorithm [3] based on an incremental growth of the convex hull of added points.

A **directional scan** is an ordered partition into scans S_i restricted to the grid domain $\mathcal{G} \subset \mathbb{Z}^2$ of a thick digital straight line \mathcal{D} , called *scan strip*. Each scan S_i is a segment of a naive line \mathcal{N}_i , called *scan line*, orthogonal to \mathcal{D} . The directional scan is defined as:

$$DS = \left\{ S_i = \mathcal{D} \cap \mathcal{N}_i \cap \mathcal{G} \mid \begin{array}{l} \vec{V}(\mathcal{N}_i) \cdot \vec{V}(\mathcal{D}) = 0 \\ h(\mathcal{N}_i) = h(\mathcal{N}_{i-1}) + p(\mathcal{D}) \end{array} \right\} \quad (2)$$

In this definition, $\vec{V}(\mathcal{N}_i) \cdot \vec{V}(\mathcal{D}) = 0$ expresses the orthogonality between the scan lines \mathcal{N}_i and the scan strip \mathcal{D} . The shift $p(\mathcal{D})$ between successive scans \mathcal{N}_{i-1} and \mathcal{N}_i guarantees that all points of \mathcal{D} are traversed only one time. The scans S_i can be iteratively parsed from a start scan S_0 to both ends (see Fig. 2(a)).

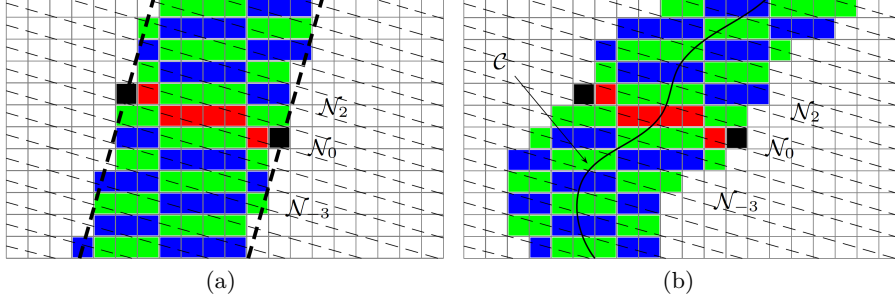


Fig. 2: (a) A directional scan in which the start scan S_0 is in red, odd scans in green, even scans in blue, bounds of scan lines \mathcal{N}_i with dashed lines and bounds of scan strip \mathcal{D} with bold dashed lines. (b) An adaptive directional scan. The scan strip is dynamically fit to the curve position, and the scans are accordingly shifted to continuously cover the curve \mathcal{C} .

An **adaptive directional scan** [5] is a dynamical version of the directional scan with an on-line registration to a moving search direction. Compared to static directional scans where the scan strip remains fixed to the initial line \mathcal{D}_0 , here the scan strip follows a curve \mathcal{C} to track while scan lines remain fixed (see Fig. 2(b)). An adaptive directional scan is defined by:

$$ADS = \left\{ S_i = \mathcal{D}_i \cap \mathcal{N}_i \cap \mathcal{G} \left| \begin{array}{l} \vec{V}(\mathcal{N}_i) \cdot \vec{V}(\mathcal{D}_0) = 0 \\ h(\mathcal{N}_i) = h(\mathcal{N}_{i-1}) + p(\mathcal{D}_0) \\ \mathcal{D}_i = \mathcal{L}(\hat{C}_i, w(\mathcal{D}_0)), i > 0 \end{array} \right. \right\} \quad (3)$$

where \hat{C}_i is a triplet composed of the director vector (a_i, b_i) and the shift to origin c_i of an estimate at position i of the tangent to the curve \mathcal{C} to track. The obtained thick digital line is used to update the scan strip and lines. The last clause expresses the scan bounds update at iteration i .

3 Main extraction framework

The automatic road extraction process is composed of two steps: selection of road seeds, then extraction of road sections. Both steps are detailed in next sections. We give here an overview of the whole process, illustrated in Fig. 3.

The first step consists in selecting a complete set of *road seeds* in the shaded DTM –a 2D gray image. The quality of this seed selection step is very important and affects strongly the final result of full road extraction. These seeds are straight line segments that are assumed to cross the road to detect. To get this selection, a morphological operator is first applied to the slope-shaded DTM (see Fig. 3b) to enhance elongated structures such as roads (see Fig. 3c). The edges of these structures are then extracted using a fast blurred segment detector (see Fig. 3d). Orthogonal segments are distributed along them to provide the road seeds (see Fig. 3e).

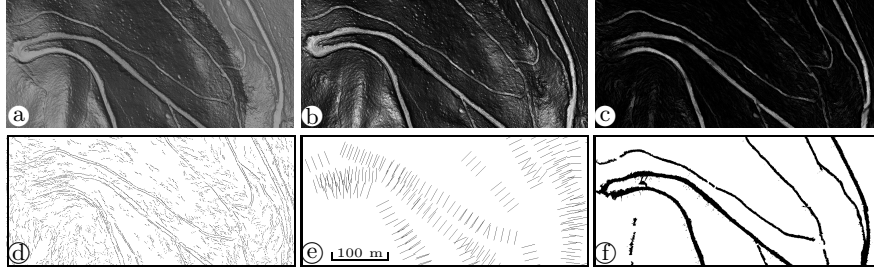


Fig. 3: Road extraction steps: (a) classical hill-shaded DTM, (b) slope-shaded DTM, (c) elongated structures enhancement, (d) edge detection, (e) road seeds selection, (f) road sections extraction.

In the second step, these seed segments are processed by a fast road extraction tool in raw LiDAR data – a 3D point cloud. For each seed, the road profile is detected under the input segment and then tracked on both sides of this segment to obtain a full road section (see Fig. 3f). This tool is based on a former approach [4] designed for a supervised context, that was adapted here to the automatic detection of forest roads using the selected seeds. In particular, the detector is modified to be more robust to possible road surface inclination and roughness using a blurred segment recognition algorithm, and shifted seed positions are tested if necessary to take into account possible inaccuracies in input data.

4 Road seeds selection from shaded DTM

The DTM is encoded as a normal vector map obtained by derivation of the original height map. Hill-shading is a classical visualization technique based on controllable directional light sources (see Fig. 3a for instance). For detection purpose, we rather use slope-shading which can be seen as a lighting by a zenital source and is simply obtained by the normal vector z-component. As pointed out in [11], slope-shading ensures a good contrast between low gradient road surface and steep adjacent road cuts (see Fig. 3b).

A mathematical morphology filter, namely RORPO [9], is then applied to the slope-shaded map. Based on path operators, it contributes to enhance elongated structures without any blurring effect, that is important for the next step. Although RORPO also provides a local orientation measure, only the output intensity map is used here (see Fig. 3c). In order to speed up this step, we use a parallelized version based on OpenMP Application Program Interface ¹.

Straight edges are then extracted using FBSD blurred segment detector [5]. Based on discrete geometry tools, this detector is fast, accurate and robust to noise. It can be run in pure automatic mode, with quite few parameter settings. In the present case, the only setting is a controllable thickness parameter fixed to 7 pixels in order to cope with slightly curved road edges (see Fig. 3d). Only longer edges (> 80 pixels) are kept for the following.

¹ <https://www.openmp.org>

Straight line segments are arranged across each output edges at regular interval along their run. They constitute seeds for the final road extraction step (see Fig. 3e). A seed length of 40 pixels (corresponding to 20 m in the tested data set) is selected to guarantee the crossing of wide roads with enough safety margin to cope with possible road curves. An interval value of 24 pixels is set as a good compromise between useless multiple detection of road parts and reduced risk to miss some road sections in occluded areas of the point cloud.

5 Road sections extraction from raw data

The input raw LiDAR data is a set of points, that are classified as ground. It is available in LAZ format files, organized in $500 \text{ m} \times 500 \text{ m}$ tiles which cover the whole acquisition area. Ground points are extracted and projected on a high resolution grid \mathcal{G} that covers all the tiles. A resolution of 0.1 m has been chosen in order to match the 0.5 m DTM grid resolution for result display.

The road detection from raw data is based on a previous work on supervised extraction of linear structures [4]. The adapted framework to automatic context is explained in this section. The road model used is first introduced, then the extraction algorithms are described.

5.1 Forest road model

In this work, the road is composed of a sequence of cross sections. A road cross section can be seen as a flat part, called *plateau*, bounded by steep cuts (see Fig. 4). The plateau thickness corresponds to road surface roughness and its length to road width. In order to cope with various forest road types, the plateau thickness is restricted to $\varepsilon = 0.25\text{m}$, and the length is set to a range interval from $L_{min} = 2\text{m}$ up to $L_{max} = 6\text{m}$. Shorter plateaux are discarded, while longer ones are just considered as unreliable and only accepted as far as at least one of the cuts is clearly detected. As a result, the detected road sticks to its sharper cut. Compared to the model used in the former approach [4], a maximal tilt angle set to $\beta_{max} = 6^\circ$ was added to cope with most road surface irregularities. A similar model was used in [6] for forestry management purposes, and showed well adapted to the context of mountainous relief. Even in the few flat areas found, bounds are often marked by surrounding ditches or by some hollowness because of terrain compaction. In valley or urban contexts, flat areas spread on each side of the road, and a denser point cloud is required to detect such subtle details as ditches or footpaths.

We assume here that the plateau point set is a blurred segment. Because of the discrete nature of the point cloud, each bound can not be exactly detected but just estimated within an interval lying between the end point of the blurred segment and next point in the scan. Bounds are considered as detected if this interval is less than $\delta_{max} = 0.5\text{m}$. When both bounds are detected, the plateau

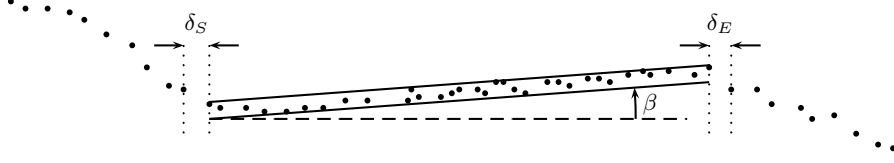


Fig. 4: Road model from an altimetric profile: the road plateau is a blurred segment with tilt angle β and bound accuracy δ_S and δ_E .

is reliable and the road section position can be estimated. But very often, only one bound is detected because of the presence of trees nearby the road. In that case, the road position is interpolated between reliable plateaux.

5.2 Road plateau detection

The new initial plateau detection procedure described in Algo. 1 consists in detecting a connected sub-sequence of nearly aligned points, namely a blurred segment, from a given start position, into a sequence of sorted 2D points along the scan direction.

The blurred segment is initialized with the nearest point to start position in the sequence. It is then extended by progressively inserting next points until outliers are met on each side. The blurred segment thickness is controlled by an initially assigned value ε . In order to optimize the output plateau thickness for further comparisons, a *pinching* procedure is performed after N_P points are inserted and an observation distance D_P is reached. The assigned thickness is set to a dilated value of the blurred segment thickness μ . The dilation ratio δ_P provides a security margin, that allows to cope with possible road roughness.

Moreover this pinching procedure helps to detects the plateau bounds with more accuracy. As long as the initial assigned thickness is set, no constraint is put on the segment orientation. It lets large freedom to incorporate spurious points: in our case, upper points on one side of the road, and lower points on the other side. The obtained blurred segment is thus thick and ill-oriented. To solve this drawback, the pinching procedure is essential.

5.3 Road section extraction

The automatic road extraction is performed on all the selected seed segments $s \in \mathcal{S}$. Algo. 2 describes the road extraction procedure. At first, this segment is used to initialize an adaptive directional scan \mathcal{A} in grid \mathcal{G} . An altimetric profile \mathcal{V} is generated from \mathcal{A} with the points in the start scan cells, sorted along the seed direction. In order to ensure a more regular repartition of the points along the profile, the grid is subdivided in smaller cells of factor N_R , and a set of N_R scans is processed instead of the initial grid scan. We select a resolution factor of $N_R = 5$, that leads to the final grid resolution of 0.1 m for \mathcal{G} .

Then, a series of plateaux is detected from regularly spaced positions around the altimetric profile center. Too narrow or too tilted ones are discarded. Among

Algorithm 1: Detection of a road plateau profile (`detectPlateau`).

Input data : A sorted sequence of points \mathcal{P} , a start position c
Parameters: Initial assigned thickness ε , pinch size N_P , pinch length D_P and pinch ratio δ_P
Output : The detected plateau \mathcal{R}

```

1   $\mathcal{R} \leftarrow \emptyset$ ;
2   $p \leftarrow \text{getNearestPoint}(\mathcal{P}, c)$ ; // Select the nearest point to start position
3   $\mathcal{P} \leftarrow \mathcal{P} - p$ ;
4   $\mathcal{B} \leftarrow \text{initializeBlurredSegment}(p, \varepsilon)$ ; // Initialize  $\mathcal{B}$  on point  $p$ 
5   $\text{pinched} \leftarrow \text{false}$ ;
6   $\text{right} \leftarrow \text{true}$ ;
7   $\text{left} \leftarrow \text{true}$ ;
8  while  $\mathcal{P} \neq \emptyset$  and ( $\text{right}$  or  $\text{left}$ ) do
9       $p \leftarrow \text{getNearestPoint}(\mathcal{P}, c)$ ; // Select next point
10      $\mathcal{P} \leftarrow \mathcal{P} - p$ ;
11     if ( $p < c$  and  $\text{left}$ ) or ( $p > c$  and  $\text{right}$ ) then
12          $\mathcal{B}' \leftarrow \mathcal{B} + p$ ; // Extend  $\mathcal{B}$  on both sides of start point
13         if  $\text{isBlurredSegment}(\mathcal{B}')$  then
14              $\mathcal{B} \leftarrow \mathcal{B}'$ ;
15             /* Blurred segment pinching */
16             if ( $\text{!pinched}$ ) and  $\text{size}(\mathcal{B}) > N_P$  and  $\text{length}(\mathcal{B}) > D_P$  then
17                  $\mathcal{B} \leftarrow \text{copyBlurredSegment}(\mathcal{B}, \min(\varepsilon, \text{thickness}(\mathcal{B}) * \delta_P))$ ;
18                  $\text{pinched} \leftarrow \text{true}$ ;
19         else
20             if  $p < c$  then
21                  $\text{left} \leftarrow \text{false}$ ;
22             else
23                  $\text{right} \leftarrow \text{false}$ ;
24 if  $\text{pinched}$  then
25      $\mathcal{R} \leftarrow \text{getPointsOfBlurredSegment}(\mathcal{B})$ ;

```

remaining candidates, the thinner plateau is kept. Notice that this part of the algorithm differs from the former method described in [4]. In supervised context, the user is assumed to select a visually suitable start seed. At least, he can adjust his selection. Strict conditions are required on the found plateau (no tilt, accurate bounds) in order to get longer road sections. This can not be ensured in pure automatic mode, so that the default plateau detection must be run at several test positions instead. Expected road sections are thus shorter in automatic mode than in supervised mode.

The final part of the algorithm is identical to the former approach [4]. In case of success of the first plateau detection, the structure is tracked and extended as far as possible on each side of the seed. At each step, a template based on the last detected plateau is built. It contains the altitude of the plateau and

the position of its bounds. In case of failure of a detection, these attributes are incremented with an estimate of the road drift in position and altitude.

Algorithm 2: Detection of a forest road from a seed line segment.

Input data : A seed line segment s and a grid of cloud points \mathcal{G}
Parameters: Number of initial trials N_I , minimal profile size N_P , maximum number of successive fails N_F and grid subdivision factor N_R
Output : The detected road section \mathcal{R}

```

1  $\mathcal{R} \leftarrow \emptyset$ ;
  /* First road detection at the segment  $s$  of  $\mathcal{S}$  */
2  $\mathcal{A} \leftarrow \text{getAdaptiveDirectionalScan}(s)$ ; // See Sec. 2
3  $\mathcal{C} \leftarrow \text{getFirstScans}(\mathcal{A}, N_R)$ ; // Get the first  $N_R$  scans of  $\mathcal{A}$  around  $s$ 
4  $\mathcal{V} \leftarrow \text{getAltimetricProfile}(\mathcal{C}, \mathcal{G})$ ; // Get point cloud of  $\mathcal{G}$  under  $\mathcal{C}$ 
5  $\mathcal{P} \leftarrow \emptyset$ ;
6  $c \leftarrow \text{getProfileCenter}(\mathcal{V})$ ; // Get the center of profile  $\mathcal{V}$ 
7 for  $\delta \leftarrow -N_I/2$  to  $N_I/2$  do
8    $\mathcal{P}' \leftarrow \text{detectPlateau}(\mathcal{V}, c + \delta)$ ; // See Algo. 1 and Sec. 5.2
9   if isConsistent ( $\mathcal{P}$ ) and thickness ( $\mathcal{P}'$ ) < thickness ( $\mathcal{P}$ ) then
10     $\mathcal{P} \leftarrow \mathcal{P}'$ ;
11     $\delta \leftarrow \delta + 1$ ;
12 if  $\mathcal{P} \neq \emptyset$  then
13    $\mathcal{R} \leftarrow \mathcal{R} + \mathcal{P}$ ; // Add the first detection to result
14    $\mathcal{T}_0 \leftarrow \text{getRoadParameters}(\mathcal{P})$ ; // Get parameters for the tracking step
  /* Road tracking on both sides of the segment  $s$  */
15   for side  $\leftarrow 1$  to 2 do
16      $\mathcal{T} \leftarrow \mathcal{T}_0$ ;
17      $N \leftarrow 0$ ;
18     do
19        $\mathcal{A} \leftarrow \text{getCenteredADS}(\mathcal{A}, \mathcal{T})$ ; // Center ADS on template
20        $\mathcal{C} \leftarrow \text{getNextScans}(\mathcal{A}, \text{side}, N_R)$ ; // Get the next  $N_R$  scans in  $\mathcal{A}$ 
21        $\mathcal{V} \leftarrow \text{getAltimetricProfile}(\mathcal{C}, \mathcal{G})$ ;
22       if size ( $\mathcal{V}$ ) >  $N_P$  then
23          $c \leftarrow \text{getTemplateCenter}(\mathcal{T})$ ; // Get template center
24          $\mathcal{P} \leftarrow \text{detectPlateau}(\mathcal{V}, c)$ ; // See Algo. 1 and Sec. 5.2
25         if  $\mathcal{P} \neq \emptyset$  and isConsistent ( $\mathcal{P}, \mathcal{T}$ ) then
26            $\mathcal{R} \leftarrow \mathcal{R} + \mathcal{P}$ ; // Add the obtained tracking to result
27            $\mathcal{T} \leftarrow \text{getRoadParameters}(\mathcal{P})$ ; // Update road template
28            $N \leftarrow 0$ ;
29         else
30            $N \leftarrow N + 1$ ;
31            $\mathcal{T} \leftarrow \mathcal{T} + \text{getRoadDrift}(\mathcal{R}, \text{side})$ ;
32       else
33         /*  $N$  is not incremented in occluded areas */
34          $\mathcal{T} \leftarrow \mathcal{T} + \text{getRoadDrift}(\mathcal{R}, \text{side})$ ;
35       while  $\mathcal{C} \neq \emptyset$  and  $N < N_F$ ;
36       side  $\leftarrow \text{side} + 1$ ;
37    $\mathcal{R} \leftarrow \text{cleanRoadSection}(\mathcal{R})$ ; // Final road section cleaning

```

The template is used to enforce spatial consistency between adjacent plateaux using tolerance thresholds for each attribute (`isConsistent` primitive in Algo. 2). The adaptive directional scan is continuously centered on the last plateau template. The tracking stops when reaching grid bounds or after N_F (set to 5 here) successive detection failures. In case of sparse profiles (featuring less than 6 points), this counter is not incremented so that occluded areas can be crossed. When a denser profile is met, the extension goes on again if a reliable plateau is detected.

Finally a post-processing validation is performed. The plateau density is checked in order to discard road sections with too much missing plateaux (less than 40% allowed). Moreover section tails (last sequence of adjacent plateaux) are also iteratively withdrawn if they are too short (10 plateaux at least). This last test contributes to remove most of wrong road ends.

6 Experiments

6.1 Usability test on a large-scale LiDAR set

At our knowledge, neither LiDAR test set, with raw data and forest road ground truth, nor concurrent approach code is publicly available for testing and comparing. This is certainly due to a large variability between different acquisition contexts and terrain configurations and maybe also to the huge data storage required. Nevertheless, the available data set used to validate the method is quite large, covering the whole Fossard mountain. Tiles corresponding to surrounding urbanized valleys were not considered as the road model used is not adapted to this context. The remaining part features 313 tiles (about 78 km²) of forested mountain area, with about 755 millions of points for a mean density of 9.65 points / m². But the repartition is not homogeneous because of the presence of many tight conifer plantations.

Processing such a large amount of tiles exceeds the memory capacity of standard computers. Therefore, blocks of tiles are successively processed for each step, until the whole area is completely covered. For the seed selection step, resulting road seeds are classified into an array according to the DTM tile they belong to. In this step, the block must include a one-tile border to avoid the splitting of many blurred segments between two adjacent tiles during edge detection stage. For the road extraction step, buffers are used to collect point tiles. Tiles are processed one by one, but road sections may have a long extent, so that a two-tiles border is required. The whole data set is traversed using a specific path minimizing tile loading operations. This parcelling strategy allows the automatic processing of quite large data sets, at the price of an execution time increase because it performs more tile loading operations, and border tiles are processed multiple times in the first step.

In the present case, the whole Fossard processing takes **1747 s** on Intel i7 processor, 840 s for the seed selection step and 907 s for the final road extraction from raw data. Most output roads are visually correct (see a detail on Fig. 5).

Sector	D (pts/m ²)	T (s)	R (%)	P (%)	F (%)
Saint-Mont	8.62	28.70	67.4616	74.8994	70.9862
Gris-Mouton	6.86	30.18	69.8697	88.1259	77.9431
Grand-Rupt	7.47	27.55	73.0137	78.3710	75.5976
Tête des Cuveaux	8.95	32.62	68.7293	80.4931	74.1475

Table 1: Road extraction performance measures

Because the simple road drift estimation procedure does not allow to predict rapid direction changes, many tight curves are left undetected. Most false detections occur in areas with low height gradient where the road model is not well adapted, or along some talwegs, old road relicts or cultivation terraces. More generally, output filtering should be added to deliver more accurate data such as road width or center line.

6.2 Performance test on smaller size areas

A finer evaluation was performed on four distinct 4×4 tile sets (4 km² each). Two of them include arranged areas for walking with some wood clearings and a large variety of tracks (*Saint-Mont* and *Tête des Cuveaux*) near the urbanized Mosel valley. The other ones (*Gris-Mouton* and *Grand-Rupt*) correspond to classical wood exploitation sectors, the last one featuring more standardized roads. During this long and tedious task, for each sector, most salient roads were carefully delineated in DTM views. Only the central line was manually extracted as the road width could not be easily estimated whatever the sector type. The achieved set of polylines constitutes the road ground truth.

For each extracted road section, missing plateaux were interpolated between detected ones. The set D of the covered pixels by these road sections was collected. It was compared to the set G_L of the covered pixels by the ground truth polylines to get a recall measure R , and to a 28 pixels dilation of G_L to get a precision measure P . This dilated set, called G_W , is assumed to enclose the real road set, including calibrated roads up to twelve meters wide, but also to take into account possible delineation errors. A F-measure F was also computed as the harmonic mean of R and P . All these measures are given by Eq. 4.

$$R = \frac{\overline{D \cap G_L}}{\overline{G_L}}, \quad P = \frac{\overline{D \cap G_W}}{\overline{D}}, \quad F = \frac{2 * R * P}{R + P} \quad (4)$$

Point density D , execution time T , recall R , precision P and F-measure F are reported in table Tab. 1, and Fig. 5 gives the results on one of the sectors. Comparing the mean execution time for these tests (7.44 s / km²) to the one observed for whole Fossard processing (22.33 s / km²) gives an idea of the parcelling strategy overcost. Achieved accuracy results are consistent with local sector features, and quite correct if we consider the diversity of processed road types and most of all the variability of their geometric characteristics along their run. In addition, the supervised mode may be used to complete most of missing road sections.

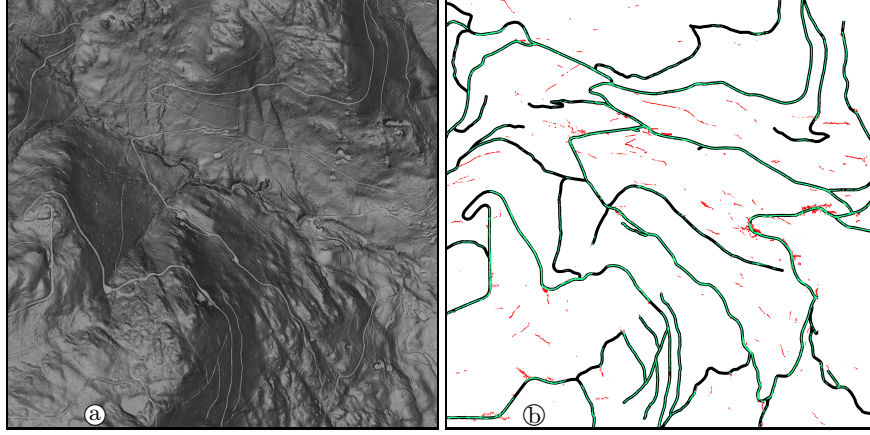


Fig. 5: Results on Grand-Rupt sector; (a) DTM map; (b) precision map: dilated ground truth in black, good detection $D \cap G_W$ in green, false detection in red.

7 Conclusion

This paper presents a new framework only based on mathematical morphology and discrete geometry tools to automatically extract forest roads in mountainous areas from LiDAR DTM and point cloud. In a first step, road seeds are selected in the shaded DTM using RORPO filter and FBSD blurred segment detector. Then a road extraction step is performed from provided seeds using only LiDAR raw data. It relies on the analysis of point profiles on each side of the seed position to detect road plateaux. Not only based on the interpolated DTM, this approach is more aware of the heterogeneous point repartition, giving better chance to cross occluded areas. The extraction framework was successfully tested on a large-scale LiDAR data set. Good execution time was obtained (**about 22 s / km²**). A comparison to selected ground truth in four smaller areas pointed out good accuracy measures (**around 70%**) regarding the quite simple road model used.

Achieved results showed the good potential of the used discrete geometry notions (blurred segment, adaptive directional scan) to process raw LiDAR data, and relevant seeds could be efficiently obtained using RORPO and FBSD. To get more effective results, it is left possible to combine this tools with more standard signal processing ones. In particular, a finer estimation of the extended road drift may help to detect rapid changes of direction, so that better aligned directional scans could be provided to extend the road section in tight curves. Some output filtering could also be added to deliver finer road width and position information. Moreover the integration of additional data such as low vegetation points could bring more clues to discriminate road sections from similar structures such as talwegs or ancient cultivation terraces. Also this new framework could probably be easily adapted to other application contexts, for instance urban structures detection and tracking from on-board sensors of smart vehicles.

Acknowledgements. This work was realized in the scope of SolHoM interdisciplinary project of Université de Lorraine. LiDAR data were acquired in the scope of PCR AGER project (*Projet collectif de recherche — Archéologie et Géoarchéologie du premier Remiremont et de ses abords*), and left available to SolHoM project.

References

1. Clode, S., Kootsookos, P., Rottensteiner, F.: The automatic extraction of roads from LiDAR data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **34**(B7) (2004)
2. David, N., Mallet, C., Pons, T., Chauve, A., Bretar, F.: Pathway detection and geometrical description from ALS data in forested montaneous areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **38**(part 3/W8), 242–247 (2009)
3. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. *Computers and Graphics* **30**(1), 30–36 (February 2006). <https://doi.org/10.1016/j.cag.2005.10.007>
4. Even, P., Ngo, P.: Live extraction of curvilinear structures from LiDAR raw data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **V-2**, 211–219 (XXIV ISPRS Congress 2020). <https://doi.org/10.5194/isprs-annals-V-2-2020-211-2020>
5. Even, P., Ngo, P., Kerautret, B.: Thick line segment detection with fast directional tracking. In: *Proc. of 20th International Conference on Image Analysis and Processing*. LNCS, vol. 11752, pp. 159–170. Springer, Trento, Italy (September 9–13 2019). https://doi.org/10.1007/978-3-030-30645-8_15
6. Ferraz, A., Mallet, C., Chehata, N.: Large-scale road detection in forested mountainous areas using airborne topographic lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing* **112**, 23–36 (February 2016). <https://doi.org/10.1016/j.isprsjprs.2015.12.002>
7. Klette, R., Rosenfeld, A.: *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco (2004)
8. Liu, Q., Kampffmeyer, M., Jenssen, R., Salberg, A.B.: Road mapping in LiDAR images using a joint-task dense dilated convolutions merging network. In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019)*. pp. 5041–5044. Yokohama, Japan (July 28 – August 2 2019). <https://doi.org/10.1109/IGARSS.2019.8900082>
9. Merveille, O., Naegel, B., Talbot, H., Najman, L., Passat, N.: 2D filtering of curvilinear structures by ranking the orientation responses of path operators (RORPO). *Image Processing On Line* **7**, 246–261 (2017). <https://doi.org/10.5201/ipol.2017.207>
10. Salberg, A.B., Trier, Ø.D., Kampffmeyer, M.: Large-scale mapping of small roads in lidar images using deep convolutional neural networks. In: Sharma, P., Bianchi, F. (eds.) *Scandinavian Conference on Image Analysis (SCIA)*. LNCS, vol. 10270, pp. 193–204. Springer, Tromsø, Norway (June 12–14 2017). https://doi.org/10.1007/978-3-319-59129-2_17
11. White, R.A., Dietterick, B.C., Mastin, T., Strohman, R.: Forest roads mapped using LiDAR in steep forested terrain. *Remote Sensing* **2**(4), 1120–1141 (2010). <https://doi.org/10.3390/rs2041120>