Open Geospatial Data,
Software and Standards

ORIGINAL ARTICLE Open Access

Generalized 3D fragmentation index derived from lidar point clouds

Vaclav Petras[1]*, Douglas J. Newcomb[2] and Helena Mitasova[1]

Abstract

Background: Point clouds with increased point densities create new opportunities for analyzing landscape structure in 3D space. Taking advantage of these dense point clouds we have extended a 2D forest fragmentation index developed for regional scale analyses into a 3D index for analyzing vegetation structure at a much finer scale.

Methods: Based on the presence or absence of points in a 3D raster (voxel model) the 3D fragmentation index is used to evaluate the configuration of a cell's 3D neighborhood resulting in fragmentation classes such as interior, edge, or patch. In order to incorporate 3D fragmentation into subsequent conventional 2D analyses, we developed a transformation of this 3D fragmentation index into a series of 2D rasters based on index classes.

Results: We applied this method to a point cloud obtained by airborne lidar capturing a suburban area with mixed forest cover. All processing and visualization was done in GRASS GIS, an open source, geospatial processing and remote sensing tool. The newly developed code is also publicly available and open source. The entire processing chain is available and executable through Docker for maximum reproducibility.

Conclusions: We demonstrated that this proposed index can be used to describe different types of vegetation structure making it a promising tool for remote sensing and landscape ecology. Finally, we suggest that processing point clouds using 3D raster methods including 3D raster algebra is as straightforward as using well-established 2D raster and image processing methods.

Keywords: 3D raster, Voxel model, Spatial pattern, Lidar, Raster algebra, Spatial indices

Background

Data acquired by airborne lidar have transformed how the Earth's surface and vegetation structure are mapped and analyzed leading to many applications, for example, in terrain modeling and ecosystem studies [1].

Lidar point clouds have been used not only to map the spatial distribution of vegetation [2–4], but also to analyze the vertical structure of forested and savanna ecosystems [5–8]. With the increasing density of points obtained by the new types of lidar technologies, such as single-photon lidar, which produce orders of magnitude more points [9], there is a need for new techniques that would take advantage of high point densities and provide analyses to support improved ecosystem management.

Many existing methods for 3D point cloud analyses are limited to 2D or 2.5D [7, 10, 11], have been implemented in a specialized lidar-processing software [5, 12, 13], or use custom low-level code [14]. To make advanced analysis of point clouds more general and accessible, we use 3D rasters and associated 3D raster algebra as the basis for developing new methods for lidar data analysis. 3D rasters, also referred to as voxels, voxel models, voxel-based space, or 3D grids, are used in many fields such as soil science [15], geology [16], atmospheric sciences [17], human anatomy [18], and 3D printing [19]. In the fields of remote sensing and geographic information systems, 3D rasters have been used with airborne lidar data to characterize fine-scale bird habitat [12] and with terrestrial lidar to characterize forest canopy fuel properties [20] and detailed tree models [21, 22]. In ecology, spatio-temporal data in 3D rasters have been used to quantify the complexity of simulated population dynamics [23] and 3D rasters representing trees have been used to assess lighting

*Correspondence: vpetras@ncsu.edu; wenzeslaus@gmail.com
The findings and conclusions in this article are those of the author(s) and do not necessarily represent the views of the U.S. Fish and Wildlife Service.
[1] North Carolina State University, Marine, Earth, and Atmospheric Sciences, 2800 Faucette Drive, Raleigh, NC 27695, USA
Full list of author information is available at the end of the article

conditions [24]. Remotely-sensed hyperspectral data have been represented and processed as 3D rasters to extract textures [25].

In order to describe vertical vegetation structure, we define a 3D version of a 2D forest fragmentation index introduced by Riitters et al. [26]. Different spatial indices have been used to describe land cover structure [26–31]. These indices were implemented in various software packages including SPAN software [32], the r.le software package coupled with GRASS GIS [33] and later replaced by different set of modules for GRASS GIS called r.li [34], the well-known FRAGSTATS software package for computing spatial indices [35], the GuidosToolbox software package for the assessment of pattern, connectivity, and fragmentation [36], and the SDMTools R package for species distribution modeling [37]. Jjumba and Dragicevic [38] presented a set of indices for the basic analysis of data represented as 3D rasters. Parrot et al. [39] defined 3D metrics for the analysis of spatio-temporal data in ecology.

The original 2D forest fragmentation index by Riitters et al. [26] was created to characterize the spatial configuration and structure of a forest at a global scale. The presented 3D fragmentation index can be used in applications describing 3D vegetation structure, classifying vegetation types, characterizing fine-scale bird habitat in three dimensions, or describing overall landscape characteristics. We present this new 3D fragmentation index as an example how a 2D index or a 2D filter can be extended into 3D and implemented in a similar way as its 2D version. To use the 3D rasters with established 2D raster processing methods and tools, we also present several methods for converting a 3D raster into a series of 2D rasters.

We provide source code for all the presented methods, which we implemented as modules for GRASS GIS, so that they can be used together with other open source geospatial processing tools [40]. We also provide a repository with all the materials needed to fully reproduce the research presented here using Docker [41].

Methods

First we describe a method for creating a 3D raster (Fig. 1) from a lidar point cloud. Then we review how the forest fragmentation index is defined in 2D. Then we define the 3D fragmentation index and review edge cases. Finally, we show methods for further analyzing 3D rasters using 2D and 3D raster processing methods.

Vegetation structure reconstruction

A 3D raster can be created from a lidar point cloud using a process called binning, rasterization, or voxelization. A value of a 3D cell (voxel) is determined by presence, count, or properties of the points which fall into a 3D space occupied by a given cell [12, 21, 38, 42]. We use point heights relative to the ground surface. Generally, binning produces outputs such as number of points per cell or, if points have values associated with them, a mean of these values or other statistics. We use binning where each cell which contains one or more points is assigned value 1, while empty cells have value 0. Alternatively, we could use a threshold for point count, mean intensity, or a percentage of points within a vertical column [8, 42].

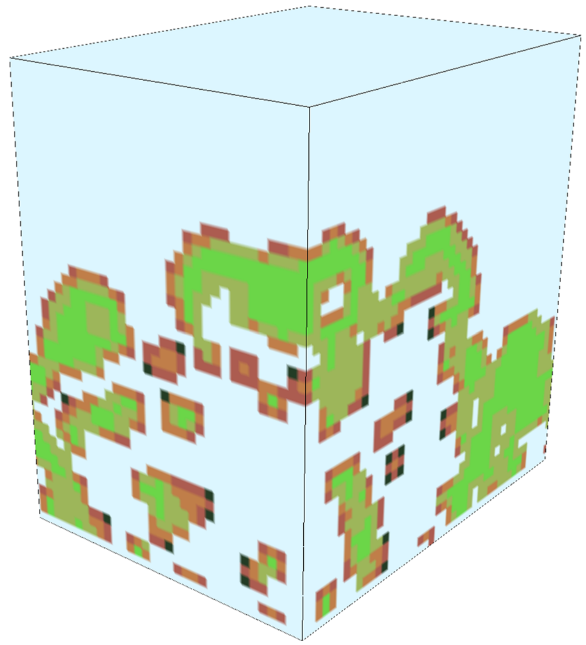

Fig. 1 3D raster example. An example 3D raster which represents the 3D fragmentation index using colors described in Fig. 3. It is a small sample of size 33 × 44 × 46 3D cells

Depending on the point cloud density and resolution of the 3D raster used for binning, the resulting 3D raster may contain a lot of empty cells. In the next step, we reconstruct the shape of the vegetation by assigning value 1 to all cells which have a neighbor with value 1. Neighborhoods can be defined in 2D or 3D with neighbors in different directions where neighboring cells are touching by faces, edges or corners. Alternatively, a larger neighborhood can be considered and then the cells in the neighborhood may not touch the central cell. We use 3D neighborhoods with 27 neighbors touching in any way. An example profile (i.e. vertical slice or transect) of the binning result in relation to the reconstructed vegetation structure is visible in Fig. 2.

2D forest fragmentation index

According to Riitters et al. [26], the 2D forest fragmentation index is defined for a 3 × 3 window in the following way. The ratio of forested cells P_f is defined as:

$$P_f = \frac{\sum_{i=-1}^{i=1} \sum_{j=-1}^{j=1} p_{i,j}}{3 \cdot 3}$$

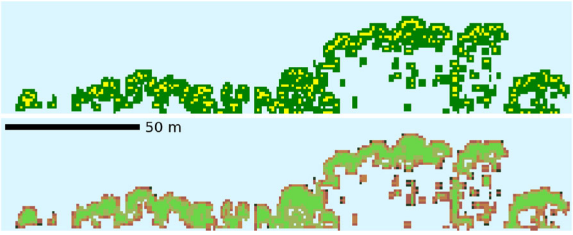

Fig. 2 Selected profile. A selected profile (vertical slice) of 3D raster representing the reconstructed 3D vegetation structure (*top*) and fragmentation index profile (*bottom*). The yellow color represents cells which contain lidar points from the original point cloud. The green color represents the reconstructed vegetation. The colors in the bottom profile represent the fragmentation index as described in Fig. 3. Vertical and horizontal scales in this figure are the same. The position of the profile is shown in Fig. 5 in 2D and in Fig. 6 in 3D space

where P_f is the value of a central cell of a moving window in the resulting raster, $p_{i,j}$ is a cell value and i and j are indices in the moving window. The value for non-forested cells is 0 and value for forested cells is 1.

Further, number of cell pairs e_1 where both cells are forested is:

$$e_1 = \sum_{j=-1}^{j=1} \sum_{i=-1}^{i=0} p_{i,j} \wedge p_{i+1,j} + \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=0} p_{i,j} \wedge p_{i,j+1}$$

The operator $\wedge$ is logical AND, which we define so that it yields 1 when both cells have value 1 and 0 otherwise.

Similarly, the number of cell pairs e_2 where at least one cell is forested is defined as follows:

$$e_2 = \sum_{j=-1}^{j=1} \sum_{i=-1}^{i=0} p_{i,j} \vee p_{i+1,j} + \sum_{i=-1}^{i=1} \sum_{j=-1}^{j=0} p_{i,j} \vee p_{i,j+1}$$

The operator $\vee$ is logical OR, which we define so that it yields 1 when at least one of the cells has value 1 and 0 otherwise.

The ratio of fully versus partially forested cell pairs P_{ff} is defined as:

$$P_{ff} = \frac{e_1}{e_2}$$

The final classification used to create the index is described in the next section.

3D fragmentation index

For a 3D raster and an arbitrary window size, we define the ratio of filled cells P_f as follows:

$$P_f = \frac{\sum_{i=-r}^{i=r} \sum_{j=-c}^{j=c} \sum_{k=-d}^{k=d} p_{i,j,k}}{l\,m\,n}$$

where $p_{i,j,k}$ is a cell value at the position i, j, k in the moving window, $l \times m \times n$ are the window proportions in each direction, and r is defined as:

$$r = \frac{l-1}{2}$$

where l must be an odd integer. The values c and d are defined in the same way using m and n respectively. If we assume isotropic environment, the window proportions in all directions would be equal, i.e. $l = m = n$.

The number of cell pairs e_1 where both cells are filled is defined as follows:

$$e_f = \sum_{i=-r}^{i=r} \sum_{j=-c}^{j=c} \sum_{k=-d}^{k=d-1} p_{i,j,k} \wedge p_{i,j,k+1}$$

$$e_g = \sum_{k=-d}^{k=d} \sum_{i=-r}^{i=r} \sum_{j=-c}^{j=c-1} p_{i,j,k} \wedge p_{i,j+1,k}$$

$$e_h = \sum_{j=-c}^{j=c} \sum_{k=-d}^{k=d} \sum_{i=-r}^{i=r-1} p_{i,j,k} \wedge p_{i+1,j,k}$$

$$e_1 = e_f + e_g + e_h$$

The number of cell pairs e_2 where at least one cell in a pair is filled is defined in the similar fashion:

$$e_2 = \sum_{i=-r}^{i=r} \sum_{j=-c}^{j=c} \sum_{k=-d}^{k=d-1} p_{i,j,k} \vee p_{i,j,k+1} + \ldots$$

Then the P_{ff} coefficient is computed in the same way as in 2D:

$$P_{ff} = \frac{e_1}{e_2}$$

We also define a difference D_P which is used later on:

$$D_P = P_f - P_{ff}$$

Finally, we use the following set of rules based on Riitters et al. [26] to classify P_f, P_{ff}, and D_P into the index classes F:

$$F = \begin{cases} \text{patch}, & \text{if } P_f < p_l \\ \text{transitional}, & \text{if } P_f >= p_l \wedge P_f > t_l \\ \text{edge}, & \text{if } P_f > t_l \wedge D_P < 0 \\ \text{perforated}, & \text{if } P_f > t_l \wedge D_P > 0 \\ \text{interior}, & \text{if } P_f = 1 \\ \text{undetermined}, & \text{if } P_f > t_l \wedge D_P = 0 \end{cases}$$

where p_l is the patch limit where the index changes from patch to transitional and t_l is the transitional limit where the index changes from edge, perforated or undetermined to transitional. Riitters et al. [26] sets $p_l = 0.4$ and $t_l = 0.6$. We use these values in this study and we provide them as defaults to the user in the software implementation, however, users can change the values if needed. The cell is marked as interior when $P_f = 1$. A graphical representation of relation between P_f and P_{ff} and the index classes (i.e. index values) is depicted in Fig. 3.

To accommodate uncertainty in floating point computations and to allow the user to loosen the requirements

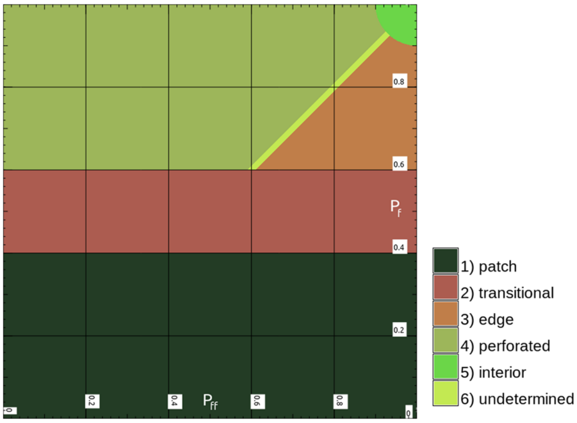

Fig. 3 Fragmentation index classification schema. The P_f variable increases in the y-axis direction and the P_{ff} variable increases in the x-axis direction. The class boundaries are set according to Riitters et al. [26] with the exception of interior and undetermined classes which are exaggerated for visualization purposes. The exterior class (denoted with *light blue color* in this manuscript) is not included as there are no values associated with it in the P_f-P_{ff} space. Each of the selected colors has different lightness allowing for gray-scale printing and limited color-blind safeness

for the interior cells, we introduce the interior limit i_l and optionally use the following condition:

$$|P_f - 1| < i_l$$

For some applications, it may be more appropriate to define the interior cells as any cells whose P_f and P_{ff} falls into a circle around the point $P_f = 1, P_{ff} = 1$ as show in Fig. 3. The condition for the interior is the following:

$$\left(P_{ff} - 1\right)^2 + \left(P_f - 1\right)^2 < i_l^2$$

We modify conditions for the classes that can overlap with the interior class, namely the edge, perforated and undetermined classes, because the interior class takes priority. For example, the modified condition for the edge class is the following:

$$P_f > t_l \wedge D_P < 0 \wedge \neg\text{interior}$$

where ¬ is logical NOT, which we define in the way that it yields 1 when the input value is 0 and vice versa.

Similarly, the general case may require that the undetermined class is wider. Therefore, we also allow the use of the following condition for the undetermined class:

$$|D_P| < u_l$$

where u_l is the undetermined limit which specifies how wide is the undetermined area around the line $P_f = P_{ff}$. Furthermore, overlapping classes, namely the edge and perforated classes, need an additional condition similar to the one for interior to exclude areas included in undetermined class.

Horizontal slices

Most raster and image processing algorithms operate on 2D rasters, not 3D rasters. 2D rasters are easier to combine with other 2D data and more suitable for creating printed or on-line 2D maps. Therefore, it is necessary to convert a 3D raster into 2D representations so that different approaches can be adopted based on the information one wishes to preserve or highlight.

The basic conversion involves splitting the 3D raster into horizontal slices, which will be represented as a series of 2D rasters. Each 2D raster represents a slice at a certain depth in the original 3D raster. This approach preserves the information about the relative height based on the order of the 2D raster in the series. The resulting series of 2D rasters can then be processed as any other series or used as image bands in subsequent analysis.

Number of cells per vertical column with a given class

A 3D raster which contains class numbers (i.e. categorical 3D raster) can be converted to a series of 3D rasters where the values in each 3D raster indicate the presence (denoted by 1) or absence (denoted by 0) of a certain class. The next step is counting the number of cells that have a class present for each vertical column. The resulting value for one class r is defined as:

$$r_{i,j} = \sum_{k=0}^{k=d} p_{i,j,k}$$

where $p_{i,j,k}$ is a value of the 3D raster (0 or 1), i, j are indices in the horizontal directions and k is an index in the vertical dimension which has values from 0, which is the minimum vertical index, to d, which is the maximum vertical index for the given 3D raster.

Generally, this can be viewed as a 3D moving window which has as many depths as the 3D raster, but only 1 × 1 dimensions in the horizontal directions. In this regard, we also provide an alternative definition of r which uses a moving window that has non-trivial dimensions in the horizontal direction. The resulting value for a given 2D cell is defined as:

$$r = \sum_{k=0}^{k=d} \sum_{i=-r}^{i=r} \sum_{j=-c}^{j=c} p_{i,j,k}$$

where k is defined in the same way as in the previous equation and i, j are indices of the moving window.

The value which is the result for the central cell is then assigned to the corresponding cell of the 2D raster. This also creates a continuous raster from classified (i.e. categorical) data such as the fragmentation index. Given the fragmentation index, we can, for example, measure the number of cells within the patch class and define a measure of patchiness based on that.

For the fragmentation index applied to vegetation structure, the resulting value for one class r is defined as:

$$r_{i,j} = \sum_{k=b_{i,j}}^{k=t_{i,j}} p_{i,j,k}$$

where $b_{i,j}$ is a value of a surface which limits the computation from the bottom and $t_{i,j}$ is a surface value which limits it from the top. The surface is a 2D raster with values corresponding to cell indices rather than coordinates. The top surface t is, for example, the top of the vegetation, which is the case for this study. The bottom surface b can be the ground surface. The equation can be modified to take into account the actual heights instead of depth indices.

We also define q as a relative count of cells with a given class. The relative count q is defined as follows:

$$q_{i,j} = \frac{r_{i,j}}{t_{i,j} - b_{i,j}}$$

For the purpose of this study we replace $b_{i,j}$ with 0 when we apply this method to the fragmentation index because we take into account all vegetation above the ground level and we use heights relative to the ground. An example at Fig. 4 shows the differences in between the absolute and relative counts and count without the surface constraint.

Once we have relative count for each of the classes, we can also determine the most common (i.e. dominant) class for each vertical column by finding a class with maximum q.

Results

For this study we used an airborne lidar point cloud. The study site, depicted in Fig. 5, is a 16 hectare (38 acre) area on North Carolina State University's campus. The data were collected during leaf-off conditions in January 2015 by the North Carolina Floodplain Mapping Program. The point cloud was classified by the data provider. We used only points classified as ground (class 2) and vegetation (classes 3, 4, and 5).

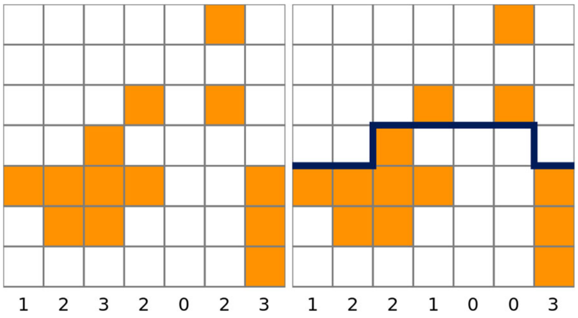

Fig. 4 Vertical class count principle. The two images show an example profile (vertical slice) of a 3D raster. The orange raster cells denote the cells with a class of interest and the empty cells are all the other classes. The number under each column (vertical column in a 3D raster) is count of the orange cells. The *left image* shows the basic example, while the right image shows just the counts of cells under an example surface shown in *blue*. Relative counts for the *right image*, i.e. counts divided by the total count of cells under the surface, would be 1/3, 2/3, 3/4, 1/4, 0, 0, and 1

Reconstructed vegetation structure

The computations were performed with cubical cells with approximately 0.9 m (3 feet) edges. With this data it was necessary to reconstruct the vegetation structure as some parts did not have enough cells with at least one point. The average point density is 2.0 points per 2D cell (0.9 m × 0.9 m) and the point density in 3D is 0.044 points per 3D cell (0.9 m × 0.9 m × 0.9 m). However, as a profile (vertical slice) in Fig. 2 shows, the structure of the vegetation emerges after the reconstruction step with a 3 × 3 × 3 neighborhood.

Fragmentation index

A profile of the 3D raster representing the fragmentation index in Fig. 2 shows areas with different structures and distributions of the index classes using 3 × 3 × 3 neighborhood. The low vegetation contains only a few exterior cells under the top of the canopy, while the higher vegetation has a lot of exterior cells in some areas and higher numbers of transitional and edge cells in other areas (likely indicating different vegetation type, not only height). The middle part of the profile in between the low and high vegetation is very dense, resulting in a lot of interior and perforated cells. Figure 6 shows the profile from Fig. 2 placed into the 3D raster together with an orthophoto.

The fragmentation index profile shows the relation of the index to the basic structure of the vegetation. Similarly, horizontal slices at different heights (depths) shown in Fig. 7 reveal the correspondence between the simple presence of the points in the given horizontal slice and the fragmentation index classes. However, the fragmentation index also considers the 3D neighborhood and describes which areas have continuous vegetation (i.e. interior class) and which areas are characterized by presence of separated or partially separated cells (i.e. edge, transitional, and patch classes). For example, Fig. 7 shows that the north and north-east parts of the study area are characterized by a low number of interior cells and a high number of transitional cells in the lower elevation range, while having a high number of interior cells in the upper elevation range. Similar differences can be also observed for the inner and boundary sections of a forested area in the south-east corner where the index suggests higher trees with dense canopy in the middle and lower, similarly structured vegetation on the boundary.

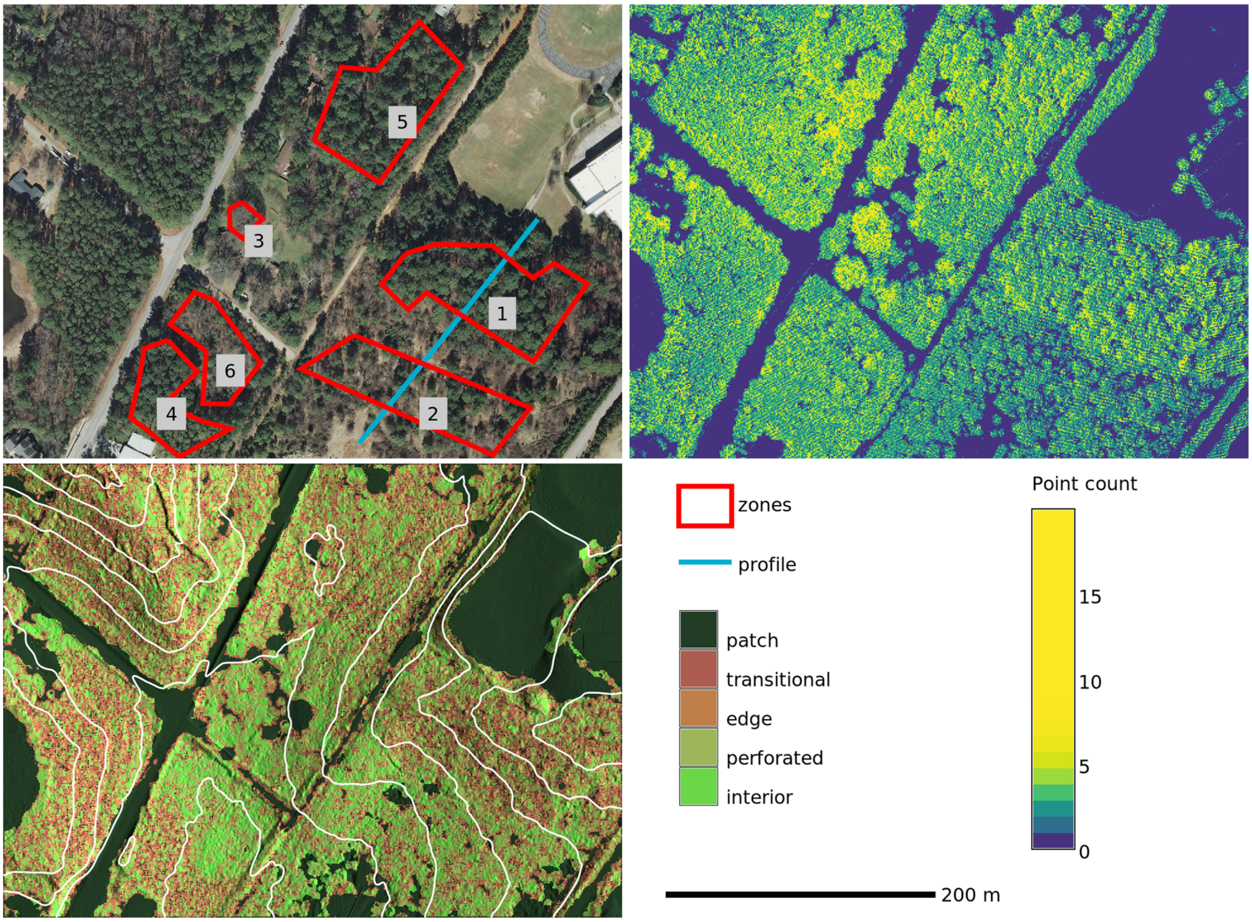

Fig. 5 Overview of the study area. The *top left figure* shows orthophoto, zones 1–6 used for zonal statistics plots (*red*), and a sample profile (*blue*) oriented from south-west to north-east. The *top right figure* is point density computed as number of lidar points in a 3 m × 3 m 2D cell. The *bottom left figure* shows one of the results, a dominant fragmentation index class in a vertical column. The *bottom left* dominant class figure is combined with topography shading and contours in *white*

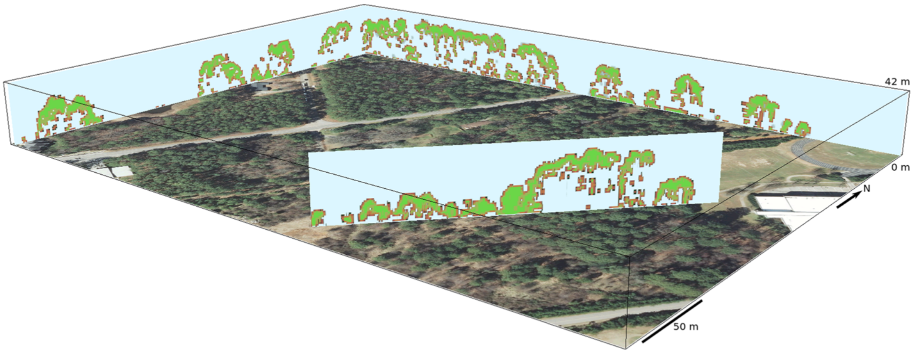

Fig. 6 3D raster profile in 3D perspective view. The image shows a profile of the fragmentation index 3D raster similar to the profile in Fig. 2 positioned in the 3D raster. The are two other additional profiles along the edges of the study area are in the back of the image. The fragmentation index is using the color table form Fig. 3. The bottom of the 3D image shows the orthophoto. The image was created in the GRASS GIS 3D view where profiles and other 3D raster slices can be explored interactively

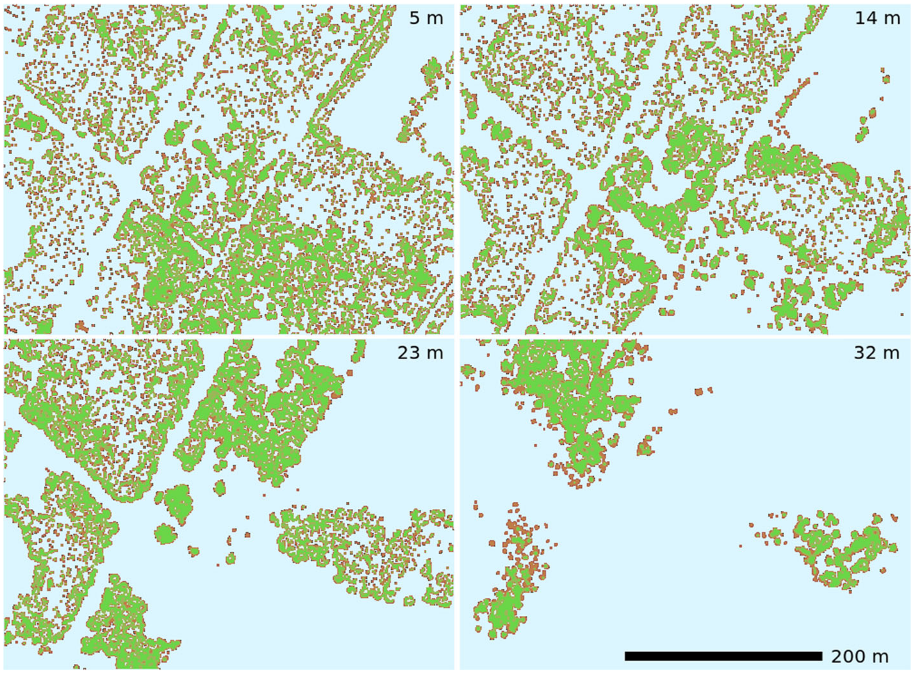

Fig. 7 Horizontal slices of fragmentation index. Each image shows a horizontal slice for a given height. The selected horizontal slices from *left to bottom* are for heights 5 m (15 feet), 14 m (45 feet), 23 m (75 feet), and 32 m (105 feet). The color table is form Fig. 3 with addition of *light blue color* for exterior. The areas in the north and north-east show only limited number of interior and a lot of transitional and edge class for the lower heights, while for the higher height a lot of interior and perforated class is visible. This is different from the area in the south-east corner with a lot of interior in the lowest levels but only exterior in the higher levels indicating dense low vegetation

Manually selected zones in the study area (Fig. 5) show different behaviors for the intermediate values used to compute the fragmentation index. When mean point count per cell n and mean ratios P_f and P_{ff} are shown as functions of height (Fig. 8), we can observe the correlation between them. We also observe that the ratio of P_f and P_{ff} changes significantly, which ultimately leads to different values for the fragmentation index revealing structure not captured by the simple point count.

2D outputs

To visualize the 3D fragmentation index in 2D and to potentially combine it with 2D data, we counted number of cells of a given class in one vertical column (2D cell). There are significant differences between the absolute cell count and the relative cell count, which is the absolute count divided by the number of cells under the vegetation. The absolute count of interior cells in Fig. 9 highlights the same areas as the point density in Fig. 5. However, the

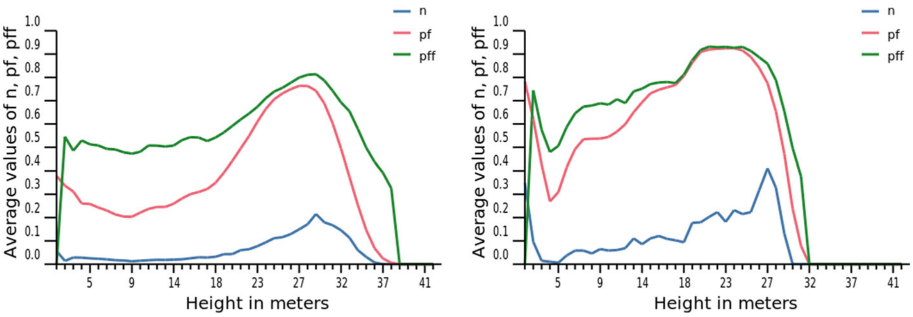

Fig. 8 Index intermediate values. Average values of intermediate 3D rasters of number of points per cell *n* (*blue*), P_f (*red*), P_{ff} (*green*) in zones 1 (*left*) and 3 (*right*) from Fig. 5. The values are averages of all cells for each depth in the given 3D raster and the x-axis shows height in meters. The graph shows correlation of the variables (P_f and P_{ff} are derived from presence and absence of the points), but it also shows how the ratio of P_f and P_{ff} changes for each height and zone. This indicates potential changes in the vegetation structure not captured by number of points alone. The values of P_f and P_{ff} are between 0 and 1 while the small average values for *n* are caused by large number of cells without any points

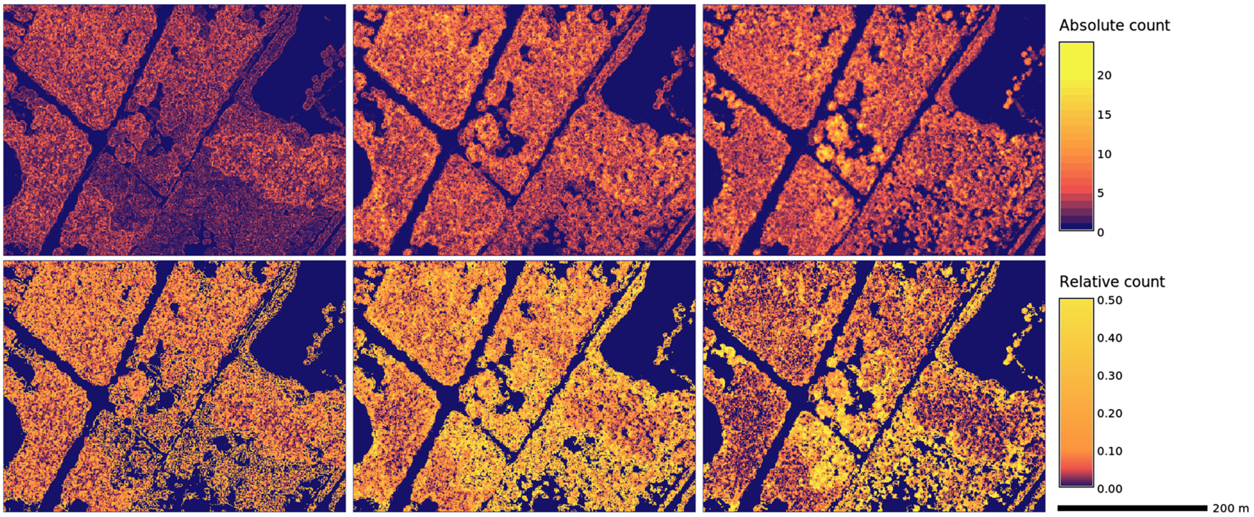

Fig. 9 Absolute and relative counts of edge, perforated, and interior cells. Counts of cells for each selected fragmentation index class; from *left in both rows*: edge, perforated, and interior. The *first row* shows absolute count of cells while the *second row* shows count of cells under the vegetation surface divided by the number of all cells under the vegetation surface. The absolute count of edge cells seems to be associated with higher vegetation, but unlike the view provided by slices in Fig. 7, the vegetation on the edge of the areas is not visible as a separate group. Worth noting is the difference between absolute and relative count of interior cells (*last column*). The absolute count shows the whole south-east quadrant as mostly consistent while the relative count shows it as two distinct groups

relative count of interior cells highlights the distinction between zones 1 and 2 (in south-east quadrant) and shows a clear difference between zones 4 and 6 (in south-west) which was not visible from the point density.

The relative count of interior cells shows distinction between zones 2, 3, and 6 (middle and south-east corner) and the rest of the study area. This is visible in Fig. 9 and also in the plot of absolute and relative counts averaged for each zone in Fig. 10. The same is valid also for absolute count of edge cells but only for zones 2 and 6, while zone 3 has counts similar to the rest of the area.

Although we see generally the same behavior for all the zones in Fig. 10, such as high number of exterior cells, we can also observe more unique behavior for some of the zones. The zones number 2 and 6 have the lowest ratio of exterior cells under the vegetation surface, and the zones 1, 4, and 5 seem to follow the same pattern for both absolute and relative counts.

The dominant class summarizes the 3D fragmentation index raster or the derived 2D rasters into one 2D raster by showing the most common class in a vertical column. Much of the information about the structure is lost, but

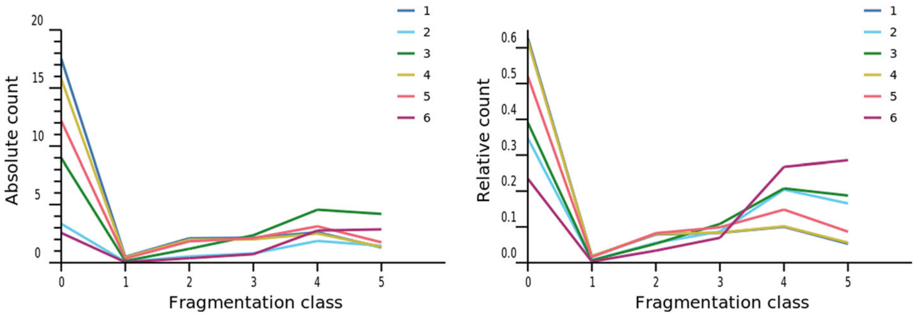

Fig. 10 Absolute and relative counts of fragmentation index classes. Number of cells in each fragmentation class for all selected zones in the study area (delineated in Fig. 5). The counts are only for the points under the vegetation surface (i.e. not including exterior cells above vegetation). The left image is the absolute count and the right image is the relative count, i.e. count divided by the height of the vegetation in cells. The x-axis numbers denotes the index classes: 0 is exterior, 1 patch, 2 transitional, 3 edge, 4 perforated, and 5 is interior as in Fig. 3; the lines in between the numbers have only visualization purpose. The relative count (*right*) shows, for example, zones 4 and 6 as the most different ones from each other, while absolute count (*left*) shows zone 3 (large deciduous tree in the middle) as the most unique one. Both plots show minimum of cells with a patch class; this is mostly due to the choice of the size of the moving window in fragmentation index computation

Fig. 11 shows the main distinctions in between the zones described above such as the differences between zone 1 and 2. However, different patterns, primarily composed of large interior patches, characterize zone 6, while the zone 2, previously similar, now has only small patches of interior cells.

When the number of cells for a given class is plotted as a function of height (Fig. 12), zones 2 and 6 again show different behavior than zones 1 and 4. The zones 1 and 4 have significantly different plots for the interior class, although the perforated class plots significantly overlap.

Software

The 3D (forest) fragmentation index is implemented in a new GRASS GIS module *r3.forestfrag*, which shares most of its code with the 2D version, the *r.forestfrag* module. This was possible thanks to the extensive use of raster algebra in both modules. The *r.forestfrag* module received a major code update as part of this work.

Furthermore, we implemented the counting of cells with a given class (i.e. category counting) in a vertical column of a 3D raster in a new module called *r3.count.categories*. To create profiles of 3D rasters (Fig. 2), we implemented the *r3.profile* module, which slices 3D raster vertically between two given points. Once again we based the code of the *r3.profile* module on its 2D equivalent, the *r.profile* module, which creates a profile from a surface map.

Finally, we prepared a publicly available Git repository hosted on GitHub. The repository contains the data for the study area, scripts to perform the analyses presented here, and details about the dependencies. Using Docker, this repository can be turned into a complete runtime environment to produce the figures (except for Figs. 1, 3 and 6), plots, and all underlying data for this manuscript. We also connected the repository with a continuous integration service, Travis CI, which will show if the basic functionality was broken by any future changes.

Discussion

Metrics for data represented as 3D rasters were presented in the past for plot and artificial data [5, 38] and specific applications [12, 20, 43]. We present a general fragmentation index based on its 2D version [26] and apply it to a sample study area. Although the initial testing of the methods and the software was done on 13 million points, the study area we selected for the manuscript is much smaller and contains only about 900 thousand points to make all the figures quickly reproducible and the data easily distributable as detailed at the end of this section.

Fragmentation index

The individual classes of the 3D fragmentation index can have different meanings. A specific application can

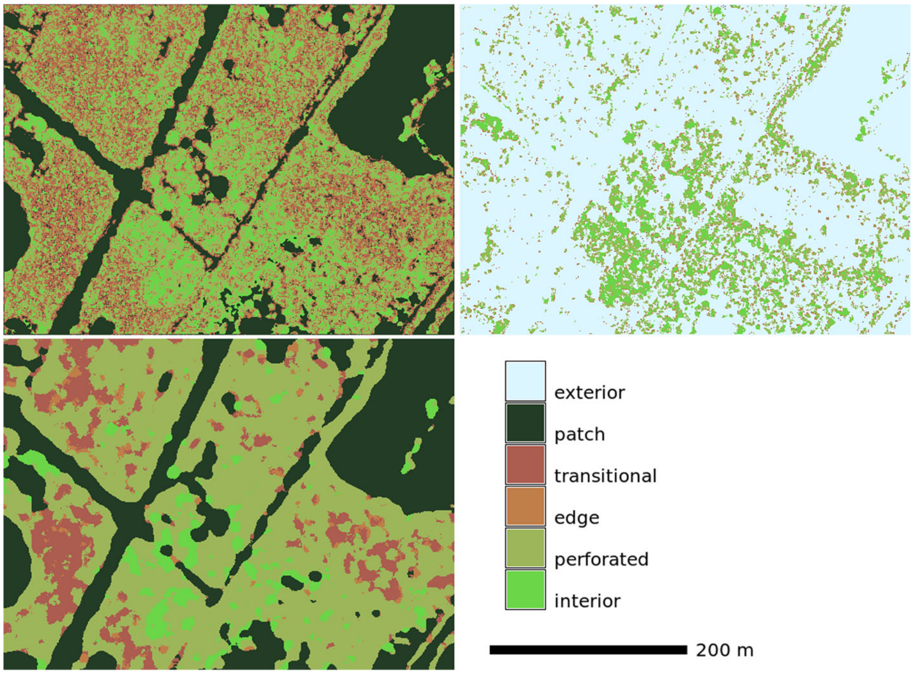

Fig. 11 Dominant fragmentation class. The *top left figure* shows the most common fragmentation class for each vertical column (2D cell) when we don't consider exterior class. The *top right figure* shows the same when we do count the exterior class. The *bottom left figure* shows the most common class in 13 × 13 neighborhood of the 2D cell which removes significant level of detail for the rasters with exterior class, but it shows more clearly the different areas such as the area in south-east quadrant with prevailing transitional class

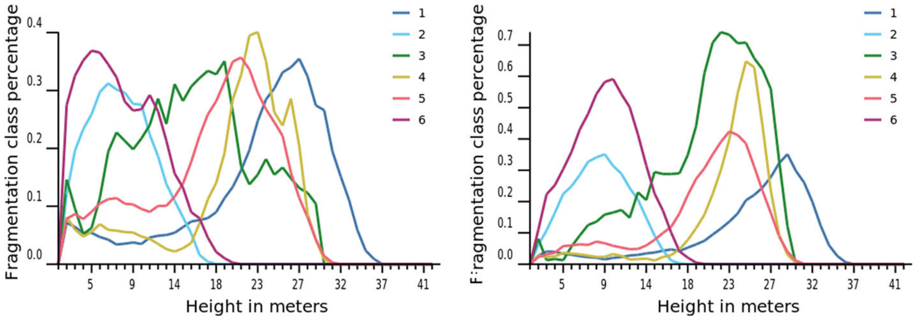

Fig. 12 Perforated and interior class percentage. The plots capture the dependency of fragmentation class percentage on height in meters for selected zones in the study area. The perforated class plot (*left*) shows two main groups of distributions. The shape of the curve for zone 3 is likely caused by the small area of the zone. The interior class plot (*right*) shows the same two main groups with smoother curves in comparison to perforated class plot

assign ranks or weights to individual classes based on their importance for a particular use case. The interior class can be important for forest structure study, but certain bird species habitat may be associated with perforated or edge classes. Alternatively, P_f and P_{ff} variables or their ratio can be used directly, leaving out the classification completely.

We applied the 3D fragmentation index on a 3D raster containing zeros and ones which was based on presence or absence of lidar points in a cell or surrounding cells in a window 3 × 3 × 3 at resolution 0.9 m. Alternatively, the 3D raster with zeros and ones can be derived from cells falling above or below some threshold for point count, total or mean intensity per cell, or their combination. The window size may be altered as well depending on how many cells contain at least one point. However, larger window size could cause creation of artificial interior cells. The 3×3×3 window creates an envelope around the cells with points which is subsequently classified as edge. The overall structure is not influenced as visible from the profiles in Fig. 2, but we must be careful when using the results in applications where the exact position of the edge cell matters. With dense enough point clouds, for example from waveform or single-photon lidars, there isn't any need for a moving window and structure reconstruction step.

The choice of resolution is influenced by the application and point density. We used small cubical cells to capture details in tree canopy or understory, but for studying individual trees or vegetation patches, it may be appropriate to choose different vertical and horizontal resolutions. Using a coarser resolution may also remove the need for the vegetation structure reconstruction step because when the point density is high enough, the cells with points will start to touch each other and there won't be any gaps for structure reconstruction to fill. On the other hand, no window at all in combination with sparse point cloud would lead to potentially incomplete model of the tree structure, no interior cells, and many transitional and patch cells. Again, the significance of this depends on the context in which the resulting fragmentation index is used.

To compute the fragmentation index, we again used 3 × 3 × 3 window. The size of the window together with the vegetation structure reconstruction causes only insignificant number of cells to be classified as patch as visible from Fig. 10. Larger window size would yield larger number of patch cells and it would change the overall fragmentation result as well [26, 29, 44]. The horizontal proportions of the window should be kept the same, i.e. the 2D projection of the window should be square. However, the size in the vertical direction can be different because, depending on the choice of resolution, the vertical relations may be different than the horizontal ones. In other words, we expect the horizontal relations to be isotropic, but in the vertical direction we may encounter anisotropy depending on what the vertical dimension in the 3D raster represents.

The choice of the window size for the fragmentation index depends on the application and using multiple window sizes may be appropriate for regressions and classifications.

The 3D fragmentation index depends on lidar pulses penetrating through the top of the canopy. When the canopy is dense, lidar pulses may not penetrate it resulting in no points under the canopy and limited applicability of 3D raster methods. The penetration also depends on the sensors used. If the point cloud comes from processing unmanned aerial system (UAS) imagery, it typically captures only the top of the canopy, while the fragmentation index works on a full 3D raster (as opposed to surface represented by 3D cells). However, UAS equipped with lidar may provide a more complete representation of the vegetation.

We applied the index strictly on vegetation, specifically different types of forest and groups of trees using lidar point cloud where the points were already classified.

However, using the fragmentation index together with the technique of counting cells of a given class per vertical column under a given surface, we could support classification of the point cloud because, for example, the buildings would be characterized by high number of exterior cells under the surface and only one patch or transitional cell right below the surface.

Processing of 3D rasters

In the context of point cloud and land cover analysis, we can now take advantage of combining the 3D raster data and processing with the 2D data so that the techniques currently in use can be enhanced by the explicit information about the 3D structure. We described and demonstrated the 3D and subsequent 2D analyses on the presented 3D fragmentation index which can be combined with the commonly used 2D (or 2.5D) metrics such as canopy height, point density, mean intensity, some of the more specific point cloud measures such as canopy-relief ratio [45], or spectral and hyperspectral data. However, alternative approaches exist, for example, 2D rasters can be extruded into 3D rasters (with the same value for all depths) and further used in 3D computations fully preserving the 3D relationships captured in the other 3D rasters.

The 3D version of the 2D forest fragmentation index, similarly to the 3D Moran's I presented by Jjumba and Dragicevic [38], shows that 2D indices can be transfered to 3D while keeping similar semantics as the 2D version. Additionally, we can observe that 3D raster algebra and 3D raster tools in general can be used for analysis in the same manner as the 2D equivalents. For example, the raster algebra expressions to determine fragmentation class based on P_f and P_{ff} values is the same for 2D and 3D and also the P_f value is computed in a similar way; only the expression for P_{ff} is different in 2D and 3D due to additional members and third value for referencing the neighboring cells.

The processing times and disk space will be typically higher for 3D rasters simply because there is one more dimension to take care of in comparison to 2D rasters, however the parallelization of the computations can be as straightforward as in 2D. In general, 3D rasters can be used for development of new remotely sensed data processing techniques and subsequent landscape ecology methodologies and measures such as landscape indices or connectivity metrics. Specifically 3D rasters in GRASS GIS were used in past for modeling of evaporation processes in 3D [46].

Reproducibility

We not only provide newly written source code and use open source software, but we also prepared a repository for a full reproducibility which is considered equal with *replicability*, *repeatability*, and *recomputability* [47] for the purpose of this manuscript. The source code is a necessary part of the method description [48, 49] and, together with documentation, a step towards re-usability. In addition to it, the use of open source software for the dependencies, most notably GRASS GIS, makes reproducibility possible [50] and opens the whole underlying computational environment for review.

However, the full and easy reproducibility is possible only when the whole processing chain and environment is shared. Thanks to Docker and the way we prepared the repository with data and code, our processing environment and the results can be reproduced on any computer [41]. The results cannot be reproduced within 10 minutes on a standard workstation as required by Schwab et al. [51] for *easily reproducible result* because of the environment building and processing time. However, obtaining the result requires up to 10 minutes of preparation, depending on whether Docker is installed on the computer or not, and the total time including building and processing is within an hour.

Conclusions

In the fields of remote sensing, GIS, and landscape ecology, scientists often process remotely sensed data as 2D rasters (images). Comparing to 2D rasters, 3D rasters are less common, despite the fact they explicitly preserve the 3D relations in the data such as lidar point clouds.

We show how 3D rasters can be utilized in a general remote sensing and GIS environment to process lidar point clouds. We used a point cloud, which is dense enough for reconstructing the structure of vegetation. The 3D raster we obtained represents the spatial information about the structure of vegetation in all three dimensions.

Well-known concepts such as moving windows and indices used in 2D processing in remote sensing or landscape ecology can be also applied in 3D. The process of doing so is very straightforward. We redefined an existing 2D forest fragmentation index as a general 3D fragmentation index, which we used to describe 3D vegetation structure. This index is represented as a 3D raster, however we show how to use it in 2D to leverage common 2D processing techniques.

The newly developed code is open source and is implemented in a well-established GIS and remote sensing software GRASS GIS. Standard GRASS GIS tools, most notably 3D raster support, were used for rest of the analysis showing how GRASS GIS can be leveraged for processing lidar point clouds in three dimensions as well as in two dimensions.

Availability of data and materials

We used a suite of existing and newly developed software to implement the new methods and perform the analyses. We used publicly available data and aim for full reproducibility of the results presented in this study.

Newly developed software

We implemented the 3D version of the forest fragmentation index as a GRASS GIS module called *r3.forestfrag*. We developed the *r3.count.categories* module to count cells with a given class (i.e. category) and we also developed the *r3.profile* module to create profiles (i.e. vertical slices) from 3D rasters. The *r3.profile* module is written in C programming language, while *r3.forestfrag* and *r3.count.categories* modules are in Python. All three modules are distributed under the terms of the GNU General Public License 2 [52] or higher (GNU GPL). The modules are now available in the GRASS GIS Add-ons repository. The documentation for the modules is available online [53].

Existing software

The rest of the study used GRASS GIS [54], its graphical user interface, and modules which can run in command line. The most important modules used in this study were *r3.in.lidar* [42] and *r3.to.rast*. GRASS GIS is platform independent, is supported on all common desktop operating systems and can run on servers and clusters. For our study we used a desktop computer with Ubuntu [55]. GRASS GIS is licensed under GNU GPL and, therefore, does not impose any special restrictions for use by non-academics. GRASS GIS is written in C and Python. The *r3.in.lidar* module requires libLAS library for reading lidar point clouds in LAS format.

Data

The lidar point cloud data used in the study are from the North Carolina Floodplain Mapping Program and are available through North Carolina's Spatial Data Download website [56]. We used tile LA_37_20079301_20160228 from phase 3 from January 2015.

Code for full reproducibility

The Git [57] repository with data for the study area and scripts in Bash for running the analyses is hosted on GitHub and publicly available [58] under GNU GPL. The repository contains a Dockerfile so that Docker [59] can be used to create the exact environment used to run the analyses for this study. The repository's web page contains instructions how to download it and the two commands needed to reproduce all of the data and figures presented in this study. The status of the code can be reviewed at Travis CI [60], a continuous integration service. A snapshot of the repository is also available as Additional file 1.

Additional file

Additional file 1: Code for full reproducibility. This is a snapshot of the repository [58] provided for full reproducibility as a 7z archive file. It contains a Dockerfile, so by using Docker it can be turned into runtime environment for processing the data and creating the figures and plots. The detailed instructions are in the readme file. The code for the newly developed modules is included. (7Z 10,850 kb)

Abbreviations

2D: Two dimensional, two dimensions; 2.5D: Two and half dimensional, two and half dimensions; 3D: Three dimensional, three dimensions; GIS: Geographic information system; GNU: GNU's Not Unix!; GNU GPL: GNU General Public License; UAS: Unmanned aerial system

Acknowledgements

We are grateful to the GRASS GIS developer and user community for developing and maintaining GRASS GIS software package. We acknowledge Emmanuel Sambale, Stefan Sylla, and Paulo van Breugel who originally implemented the *r.forestfrag* module in GRASS GIS. We would like to also acknowledge the creators of the open source software used for this study, including, but not limited to, Python, GNU Compiler Collection (GCC), Linux, Ubuntu, ImageMagick, Kile, LaTeX, Docker, Travis CI, and Git. We acknowledge GitHub, Travis CI, and Overleaf services. We also acknowledge Paul Tol for the color-blind safe and printer-friendly color scheme for qualitative data [61] used in Figs. 8, 10 and 12. We acknowledge the authors of the viscm tool [62], which we used to design the color table for the fragmentation index (Fig. 3). We also acknowledge the authors of Matplotlib color tables viridis and plasma which we used in Fig. 5 and modified for Fig. 9. Finally we also acknowledge Brendan A. Harmon and Anna Petrasova for providing feedback on the text of the manuscript.

Authors' contributions

VP developed the methods, case study, processed data and prepared the manuscript. HM and DJN provided critical revisions to the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

[1]North Carolina State University, Marine, Earth, and Atmospheric Sciences, 2800 Faucette Drive, Raleigh, NC 27695, USA. [2]U.S. Fish and Wildlife Service, 551-F Pylon Drive, Raleigh, NC 27606, USA.

Received: 7 December 2016 Accepted: 22 March 2017
Published online: 20 April 2017

References

1. Lefsky MA, Cohen WB, Parker GG, Harding DJ. Lidar Remote Sensing for Ecosystem Studies. BioScience. 2002;52(1):19–30.
2. Strîmbu VF, Strîmbu BM. A graph-based segmentation algorithm for tree crown extraction using airborne LiDAR data. ISPRS J Photogrammetry Remote Sensing. 2015;104:30–43.
3. Kobal M, Bertoncelj I, Pirotti F, Dakskobler I, Kutnar L. Using Lidar Data to Analyse Sinkhole Characteristics Relevant for Understory Vegetation under Forest Cover—Case Study of a High Karst Area in the Dinaric Mountains. PLOS ONE. 2015;10(3):1–19.
4. Tang H, Brolly M, Zhao F, Strahler AH, Schaaf CL, Ganguly S, Zhang G, Dubayah R. Deriving and validating Leaf Area Index (LAI) at multiple spatial scales through lidar remote sensing: A case study in Sierra National Forest, CA. Remote Sensing Environ. 2014;143:131–41.
5. Morsdorf F, Mårell A, Koetz B, Cassagne N, Pimont F, Rigolot E, Allgöwer B. Discrimination of vegetation strata in a multi-layered Mediterranean

forest ecosystem using height and intensity information derived from airborne laser scanning. Remote Sensing Environ. 2010;114(7):1403–15.

6. Fisher JT, Erasmus BF, Witkowski ET, Aardt J, Wessels KJ, Asner GP. Savanna woody vegetation classification–now in 3-D. Appl Vegetation Sci. 2014;17(1):172–84.
7. Davies AB, Asner GP. Advances in animal ecology from 3D-LiDAR ecosystem mapping. Trends Ecol Evol. 2014;29(12):681–91.
8. Kumar J, Weiner J, Hargrove WW, Norman SP, Hoffman FM, Newcomb D. Characterization and classification of vegetation canopy structure and distribution within the Great Smoky Mountains National Park using LiDAR. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW). Atlantic: IEEE; 2015. p. 1478–1485. doi:10.1109/ICDMW.2015.178.
9. Swatantran A, Tang H, Barrett T, DeCola P, Dubayah R. Rapid, high-resolution forest structure and terrain mapping over large areas using single photon lidar. Sci Rep. 2016;6:28277. doi:10.1038/srep28277. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4916424/.
10. Priestnall G, Jaafar J, Duncan A. Extracting urban features from LiDAR digital surface models. Comput Environ Urban Syst. 2000;24(2):65–78.
11. Höfle B, Mücke W, Dutter M, Rutzinger M, Dorninger P. Detection of building regions using airborne LiDAR - A new combination of raster and point cloud based GIS methods. In: Geospatial Crossroads @ GI_Form '09 - Proceedings of the Geoinformatics Forum. Salzburg: (2009). p. 66–75.
12. Sasaki T, Imanishi J, Fukui W, Morimoto Y. Fine-scale characterization of bird habitat using airborne LiDAR in an urban park in Japan. Urban Forestry Urban Greening. 2016;17:16–22.
13. Levick SR, Setterfield SA, Rossiter-Rachor NA, Hutley LB, McMaster D, Hacker JM. Monitoring the distribution and dynamics of an invasive grass in tropical savanna using airborne LiDAR. Remote Sensing. 2015;7(5):5117–32.
14. Wu B, Yu B, Yue W, Shu S, Tan W, Hu C, Huang Y, Wu J, Liu H. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. Remote Sensing. 2013;5(2):584–611.
15. Veronesi F, Corstanje R, Mayr T. Mapping soil compaction in 3D with depth functions. Soil Tillage Res. 2012;124:111–8.
16. Hickin AS, Kerr B, Barchyn TE, Paulen RC. Using ground-penetrating radar and capacitively coupled resistivity to investigate 3-D fluvial architecture and grain-size distribution of a gravel floodplain in northeast British Columbia, Canada. J Sedimentary Res. 2009;79(6):457–77.
17. Nativi S, Blumenthal B, Habermann T, Hertzmann D, Raskin R, Caron J, Domenico B, Ho Y, Weber J. Differences among the data models used by the Geographic Information Systems and Atmospheric Science communities. In: Proceedings of American Meteorological Society–20th Interactive Image Processing Systems Conference, Seattle (WA); 2004.
18. Caon M. Voxel-based computational models of real human anatomy: a review. Radiat Environ Biophys. 2004;42(4):229–35.
19. Hiller J, Lipson H. Design and analysis of digital materials for physical 3D voxel printing. Rapid Prototyping J. 2009;15(2):137–49.
20. García M, Danson FM, Riano D, Chuvieco E, Ramirez FA, Bandugula V. Terrestrial laser scanning to estimate plot-level forest canopy fuel properties. Int J Appl Earth Observation Geoinformation. 2011;13(4):636–45.
21. Gorte B, Winterhalder D. Reconstruction of laser-scanned trees using filter operations in the 3D raster domain. Int Arch Photogrammetry, Remote Sensing Spatial Inform Sci. 2004;36(Part 8):2.
22. Brolly G, Király G, Czimber K. Mapping forest regeneration from terrestrial laser scans. Acta Silvatica et Lignaria Hungarica. 2013;9(1):135–46.
23. Parrott L. Quantifying the complexity of simulated spatiotemporal population dynamics. Ecol Complexity. 2005;2(2):175–84.
24. Douglas E, Martel J, Cook T, Mendill C, Marshall R, Chakrabarti S, Strahler A, Schaaf C, Woodcock C, Liu Z, et al. A Dual-Wavelength Echidna Lidar for Ground-Based Forest Scanning. In: Proceedings of SilviLaser 2012: First Return, 12th International Conference on LiDAR Applications for Assessing Forest Ecosystems. Vancouver: 2012; p. 361.
25. Tsai F, Chang CK, Rau JY, Lin TH, Liu GR. 3D computation of gray level co-occurrence in hyperspectral image cubes In: Yuille AL, Zhu S-C, Cremers D, Wang Y, editors. Energy Minimization Methods in Computer Vision and Pattern Recognition: 6th International Conference, EMMCVPR 2007, Ezhou, China, August 27–29, 2007. Proceedings. Berlin, Heidelberg: Springer; 2007. p. 429–40. Springer.
26. Riitters K, Wickham J, O'Neill R, Jones B, Smith E, et al. Global-scale patterns of forest fragmentation. Conserv Ecol. 2000;4(2):3.
27. McGarigal K, Cushman S, Regan C. Quantifying terrestrial habitat loss and fragmentation: A protocol. Amherst, MA: University of Massachusetts, Department of Natural Resources Conservation. 2005;113. https://www.treesearch.fs.fed.us/pubs/52866.
28. Cushman SA, Gutzweiler K, Evans JS, McGarigal K. In: Cushman SA, Huettmann F, editors. The Gradient Paradigm: A Conceptual and Analytical Framework for Landscape Ecology. Tokyo: Springer; 2010, pp. 83–108.
29. Hurd JD, Wilson EH, Lammey SG, Civco DL. Characterization of forest fragmentation and urban sprawl using time sequential Landsat imagery. In: Proceedings of the ASPRS Annual Convention, St. Louis, MO; 2001. p. 2001.
30. McGarigal K, Tagil S, Cushman SA. Surface metrics: an alternative to patch metrics for the quantification of landscape structure. Landscape Ecol. 2009;24(3):433–50.
31. Riitters KH, O'neill R, Hunsaker C, Wickham JD, Yankee D, Timmins S, Jones K, Jackson B. A factor analysis of landscape pattern and structure metrics. Landscape Ecol. 1995;10(1):23–39.
32. Turner MG. Spatial and temporal analysis of landscape patterns. Landscape Ecol. 1990;4(1):21–30.
33. Baker WL, Cai Y. The r.le programs for multiscale analysis of landscape structure using the GRASS geographical information system. Landscape Ecol. 1992;7(4):291–302.
34. Rocchini D, Petras V, Petrasova A, Chemin Y, Ricotta C, Frigeri A, Landa M, Marcantonio M, Bastin L, Metz M, Delucchi L, Neteler M. Spatio-ecological complexity measures in GRASS GIS. Computers & Geosciences. 2016. ISSN 0098-3004, http://dx.doi.org/10.1016/j.cageo.2016.05.006.
35. McGarigal K, Cushman SA, Neel MC, Ene E. FRAGSTATS: Spatial Pattern Analysis Program for Categorical Maps. Computer software program produced by the authors at the University of Massachusetts, Amherst. 2002. http://www.umass.edu/landeco/research/fragstats/fragstats.html.
36. Vogt P. GUIDOS: tools for the assessment of pattern, connectivity, and fragmentation. In: EGU General Assembly Conference Abstracts, vol. 15. Vienna. 2013; p. 13526.
37. VanDerWal J, Falconi L, Januchowski S, Shoo L, Storlie C. SDMTools: tools for processing data associated with species distribution modelling exercises.–R package, ver. 1.1-20. 2014. https://CRAN.R-project.org/package=SDMTools.
38. Jjumba A, Dragićević S. Spatial indices for measuring three-dimensional patterns in a voxel-based space. J Geograph Syst. 2016;18(3):183–204. doi:10.1007/s10109-016-0231-0.
39. Parrott L, Proulx R, Thibert-Plante X. Three-dimensional metrics for the analysis of spatiotemporal data in ecology. Ecol Inform. 2008;3(6):343–53.
40. Neteler M, Bowman MH, Landa M, Metz M. GRASS GIS: A multi-purpose open source GIS. Environ Modell Softw. 2012;31(0):124–30.
41. Boettiger C. An introduction to Docker for reproducible research. ACM SIGOPS Oper Syst Rev. 2015;49(1):71–9.
42. Petras V, Petrasova A, Jeziorska J, Mitasova H. Processing UAV and lidar point clouds in GRASS GIS. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2016;XLI-B7:945–952.
43. Vetter M, Höfle B, Hollaus M, Gschöpf C, Mandlburger G, Pfeifer N, Wagner W. Vertical vegetation structure analysis and hydraulic roughness determination using dense ALS point cloud data–a voxel based approach. Int Arch Photogrammetry, Remote Sensing Spatial Inform Sci. 2011;38:5.
44. Riitters KH, O'neill R, Jones K. Assessing habitat suitability at multiple scales: a landscape-level approach. Biol Conserv. 1997;81(1):191–202.
45. McGaughey RJ. FUSION/LDV: Software for LIDAR data analysis and visualization. Department of Agriculture, US, Forest Service, Pacific Northwest Research Station: Seattle, WA, USA. 2009;123(2). http://forsys.cfr.washington.edu/fusion/FUSION_manual.pdf.
46. Ciolli M, De Franceschi M, Rea R, Zardi D, Zatelli P. Modelling of evaporation processes over tilted slopes by means of 3D GRASS raster. In: Proceedings of the Open Source GIS-GRASS Users Conference 2002. Trento: 2002.
47. Fehr J, Heiland J, Himpe C, Saak J. Best practices for replicability, reproducibility and reusability of computer-based experiments exemplified by model reduction software. AIMS Math. 2016;1(Math-03-00261):261.
48. Ince DC, Hatton L, Graham-Cumming J. The case for open computer programs. Nature. 2012;482(7386):485–8.

49. Morin A, Urban J, Adams P, Foster I, Sali A, Baker D, Sliz P. Shining light into black boxes. Science. 2012;336(6078):159–60.
50. Lees JM. Open and free: Software and scientific reproducibility. Seismological Res Lett. 2012;83(5):751–2.
51. Schwab M, Karrenbach N, Claerbout JF. Making scientific computations reproducible. Comput Sci Eng. 2000;2(6):61–7.
52. Free Software Foundation: GNU General Public License, Version 2. https://www.gnu.org/licenses/gpl-2.0.en.html Accessed 23 Nov 2016.
53. GRASS GIS Contributors Community: GRASS GIS Add-ons Repository. https://grass.osgeo.org/grass70/manuals/addons/ Accessed 23 Nov 2016.
54. GRASS Development Team: Geographic Resources Analysis Support System (GRASS GIS) Software. Open Source Geospatial Foundation. http://grass.osgeo.org Accessed 23 Nov 2016.
55. Canonical: Ubuntu. http://ubuntu.com/ Accessed 23 Nov 2016.
56. NC Floodplain Mapping Program: North Carolina's Spatial Data Download. https://rmp.nc.gov/sdd Accessed 15 Sep 2016.
57. Git. https://git-scm.com/ Accessed 20 Nov 2016.
58. Petras V. 3D Fragmentation Index Repository. https://github.com/wenzeslaus/forestfrag3d Accessed 27 Nov 2016.
59. Docker. https://www.docker.com/ Accessed 20 Nov 2016.
60. Travis CI – Forest Fragmentation 3D Repository Status. https://travis-ci.org/wenzeslaus/forestfrag3d Accessed 27 Nov 2016.
61. Tol P. Paul Tol's Notes – Palettes and Templates. https://personal.sron.nl/~pault/ Accessed 25 Nov 2016.
62. Smith NJ, van der Walt S. GitHub viscm Repository. https://github.com/matplotlib/viscm Accessed 26 Nov 2016.