Extending morphological pattern segmentation to 3D voxels

Short Communication

Tarmo K. Remmel

Faculty of Environmental and Urban Change, York University, N430 Ross Building, 4700 Keele Street, Toronto, Ontario, Canada, M3J 1P3, +1-416-736-2100 ext. 22496, remmelt@yorku.ca, https://orcid.org/0000-0001-6251-876X;

Abstract

This short communication introduces the logic, demonstrates its use, and identifies the availability of a new tool that extends the traditional 2D morphological segmentation of binary raster data into the 3-dimensional realm of voxels. A combination of 3-dimensional array data and network graph theory are implemented to facilitate the logical parsing of identified 3-dimensional features into their mutually exclusive constituent morphological classes. All processing is performed in the *R* environment, providing the ability for anyone to perform the demonstrated analyses on their own data. The only input requirement is a binary (1 = feature of interest, 0 otherwise) 3-dimensional array, where each voxel of interest is then classified into classes called *outside*, *mass*, *skin*, *crumb*, *antenna*, *circuit*, *bond, and void* that correspond their 2-dimensional equivalents of *background*, *core*, *edge*, *islet*, *branch*, *loop*, *bridge, and perforation*. An additional class called the *void-volume* identifies voxels belonging to the empty space within the object of interest. The work helps to bring pattern metrics into the 3-dimensional world, particularly given the reliance on adjacency and connectivity assessments.

Extending morphological landscape pattern segmentation to 3D voxels

Introduction

This short communication describes the logic and implementation of a new tool that extends conventional two-dimensional (2D) morphological segmentation (Soille and Vogt 2009), from a grid of pixels, into a three-dimensional (3D) space represented by voxels. Voxels are pixels that have been extruded to have a volume (Popescu and Zhao 2008), which is achieved by extending a pixel's footprint along a third axis. In this paper, to keep things simple, voxels will be used to refer to perfect cubes that are arranged in a regular fashion into a 3D array (imagine a Rubik's cube), with each voxel representing a discrete volume of geographic space. In principle, each of the three dimensions of a voxel may differ.

Spatial data, regardless of its thematic content, is regularly represented in 2D vector or raster formats that provide a representation for the distribution of one or more thematic classes within a geographic extent. These have traditionally been subjected to a variety of numeric and geometric assessments of pattern (Riitters et al. 1995; Uuemaa et al. 2009; Frazier and Kedron 2017). Much of the development has been for 2D data, though some 3D metrics have been developed; these tend to focus on composition rather than configuration, thus ignoring the importance of adjacency (Kedron et al. 2019). Some 2D landscape concepts, such as porosity (Remmel, 2018) and fragmentation (Fahrig et al., 2019), could translate relatively easily into 3D (Petras et al., 2017), but overall implementation of 3D metrics has been slow, likely due to the complexity of measuring 3D adjacencies. Advances in the 3D domain have typically involved voxel analysis of lidar point clouds (Wu et al., 2013) or the inference of 3D

parameters through estimation or modelling (Liu et al. 2017), but not for morphological segmentation of existing feature representations.

The increasing availability and access to lidar data (Szpakowski and Jensen 2019) and the implementation of Structure from Motion (SfM) for the extraction of 3D point clouds from planar images (Guerra-Hernández et al. 2018), are helping to forge new landscape analytical directions among landscape ecologists and other users of spatial data. These data clouds and volumetric data further emphasize the importance and necessity of 3D analytical tools. The method presented in this short communication achieves the extension of morphological segmentation tools (Vogt et al. 2007; Soille and Vogt 2009) into the third dimension by focusing purely on the adjacency, connectivity, and relative positioning of voxels belonging to a common object. The presented tool segments voxel data into mutually exclusive 3D morphological element classes.

Morphological spatial pattern segmentation (Vogt et al. 2007; Soille and Vogt 2009) provides an intuitive and tangible alternative to landscape metrics. During morphological segmentation, pixels are assigned to mutually exclusive categories based on the structural role they play in relation to other pixels. The work presented here builds on the foundational work introduced with the Morphological Spatial Pattern Analysis (MSPA) generic image analysis framework that is described by living documents available from https://forest.jrc.ec.europa.eu/en/activities/lpa/mspa/. Morphological segmentations can be used to summarize spatial patterns (Ye et al. 2020) or feed further analyses with software environments such as the GuidosToolbox (Vogt and Riitters 2017). Morphological segmentation allows the consistent parsing of each cell's contribution to the overall structure of an object and maintains consistency in the minimum mapping unit for analysis. Therefore, the challenges of deciding among

landscape metrics (Riitters et al. 1995; Cushman et al. 2008), identifying metric value expectations (Remmel and Csillag 2003; Neel et al. 2004), or the handling of challenges stemming from non-normal metric distributions to compare landscapes (Remmel and Fortin 2013) can be avoided, and their misuses (Li and Wu 2004) minimized.

True 3D data from lidar and SfM are contributing to a growing data stockpile with *x,y,z* coordinates and are relatively easily represented as volumetric data with voxels as the base spatial unit. It is also possible to consider the *z*-axis as time rather than a vertical axis (*x,y,t*), and thus use a 3D morphological segmentation to facilitate temporal pattern analysis. This extends the potential of feature or phenomenon based change tracking with hyper-local measures of configuration (Remmel 2020), pattern-based identification of landscape types with multi-thematic data (Nowosad and Stepinski 2021), or to extend forest disturbance recovery analyses (Pflugmacher et al. 2014; White et al. 2017).

The 2D segmentation of binary raster patterns into primary morphological elements (Soille and Vogt 2009) results in a minimum of seven foreground (core, edge, islet, branch, bridge, loop, and perforation) and three background classes (background, border-opening, and core-opening) which are described in the MSPA documentation. MSPA can return up to 25 classes, but this level of segmentation is not considered for implementation here currently. This type of segmentation allows secondary assessments of variable states and distributions within classes or comparisons among morphological elements, where the morphology acts as a structural factor, or grouping variable. The advancement presented in this short communication is the extension of this partitioning logic to 3D, resulting in nine corresponding volumetric morphological classes (Table 1). The new 3D classes extend the primary seven foreground classes and two of the background classes.

Table 1. A mapping of 2D morphological element terminology onto corresponding 3D classes.

2D Morphological Element	3D Morphological Element
Background	Outside
Core	Mass
Edge	Skin
Islet	Crumb
Branch	Antenna
Loop	Circuit
Bridge	Bond
Core-Opening	Void-volume
Perforation	Void

Methods

Implementation of the 3D morphological segmentation is achieved by *morph3d*[1], and associated functions written for *R* (R Core Team 2021) that also depend on critical helper functions imported from packages *rgl* (Adler and Murdoch 2021), *reshape2* (Wickham 2007), *stringr* (Wickham 2019), and *igraph* (Csárdi and Nepusz 2006). The only input required is a 3D binary array where 1 = voxels belonging to a feature, class, or phenomenon of interest; all other voxels must be coded as 0 (Figure 1a). The features of interest can form any number of contiguous clusters of theoretically any number of voxels. Though in print, Figure 1 is static, all visualizations produced by *morph3d* are interactive 3D environments that can be rotated, zoomed, and examined from virtually any observation point, providing functionality that greatly improves the visualization and interpretation of 3D data.

[1] *morph3d* is a complete software tool and will be available via CRAN (Comprehensive R Archive Network) for free and open distribution in the package *morph*. A copy of the code can also be obtained from the author.

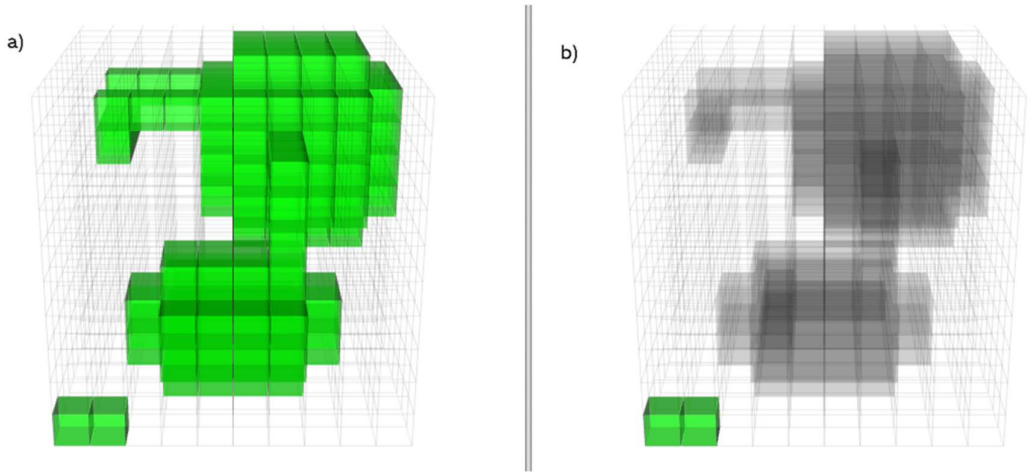

Figure 1. A volumetric spatial feature of interest shown in green (a) and its partitioning into discrete objects (b). Each voxel is tracked with a unique identification code and each discrete object in (b) is also coded with a separate unique object identifier and corresponding colour.

The segmentation logic described here follows an algorithm that guarantees mutual exclusivity in the classification of each voxel into a morphological class. To initialize the process, each voxel is assigned a unique ID and then the complete volume is inset into a larger array (that has been expanded outward in each direction by 1 voxel and whose IDs are initialized with 0; this ensures that all feature voxels in the input data will have a proper edge identified in subsequent processing. To tabulate all pairs of adjacent voxels (six neighbours in 3D), the original array is shifted 1-voxel dimension in each of the 6 possible directions (up, down, left, right, forward, backward), each time recording the voxel ID pairs formed between the original shifted data and the larger array of unshifted data. Collectively this is summarized into a table containing a listing of all neighbours for each voxel. The table is reprocessed to produce a matrix having two columns, such that each row contains two voxel IDs (as column values) that form a single neighbour-pair of voxels that exist within the volume. All duplicate pairs are reduced to a single entry since the directionality of the neighbour structure is not

important (e.g., voxel neighbours 101—100 and 100—101 are identical). This matrix of neighbouring pairs is then converted into an undirected network graph (Rahman 2017) with tools provided in the *igraph* package for *R*. Graph theory allows voxels to be considered neighbours with other voxels if they share a face. Mapping, maintaining, and assessing 3D connectivity is explicitly recorded within graphs that permit spatial associations to be easily tested and for morphology to be assessed.

Morphological processing commences in a sequence of eight main steps. First, voxels with IDs of 0 are deleted from the graph which is subsequently decomposed to produce a sub-graph for each disconnected sub-object within the input data (e.g., for each object depicted in Figure 1b). Unique object IDs are assigned to the voxels of each separate object to allow counting of the individual objects within the dataset and to associate individual voxels with unique and discrete objects. The green feature in Figure 1a is comprised of 2 discrete objects (Figure 1b) and are independently processed. The ability to view such data in an interactive 3D viewer allows occluded voxels to be viewed by rotating the view.

The second step requires the identification of *mass* voxels. Since these represent the inner core of a 3D object, any mass voxel must be neighboured on all 6 sides by another feature voxel (Figure 2). For any voxel in the graph with degree = 6, a morphology code of 2 is assigned to the corresponding voxel. These morphology codes are recorded in a separate 3D array that will continue to be updated as each morphological class is processed, terminating only when each morphological class has been assessed and each voxel has been assigned one of the 9 possible morphological classes. The third step requires that each unclassified vertex that neighbours a *mass* be classified as *skin* (Figure 2) and given the morphology code of 3. Given the logic to identify *mass* voxels, these two morphological classes are straightforward to identify,

and no *mass* voxels will ever be without full enclosure by *skin* voxels. A flood-filling algorithm is used to distinguish *skin* voxels (external edges) from *void* voxels (internal edges). Computationally, the object of interest is immersed within a volume of -1 coded voxels that differentiates the empty spaces within the structure that remain coded as 0. *Skin* voxels that neighbour voxels coded with -1 remain as *skin*, while those that neighbour voxels coded with 0 become reclassified as *voids*. The voxels forming the hole are further labelled as *void-volume* to identify that they are holes within the object of interest, but not explicitly part of it.

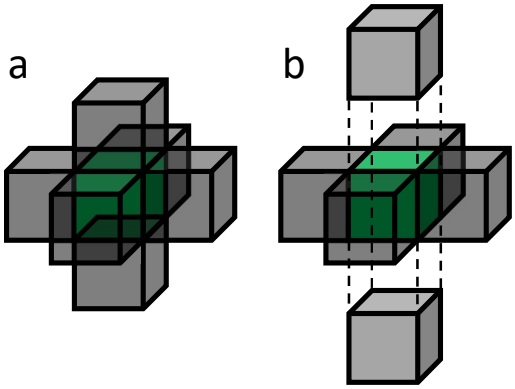

Figure 2. A simple 3D object (a) is comprised of 7 voxels. The green voxel is a mass since it is bounded on all 6 sides by semi-transparent black voxels. The same object is drawn in exploded form (b), more clearly showing the mass voxel at the center.

The fourth step is to identify *crumbs*, those discrete objects that contain no *mass* voxels, and assign them the morphology code 4. Each discrete object is considered in turn and is tested for the presence of *mass* voxels and whether it makes any connections to *skin* voxels. Connections to skin voxels indicate that the object is a one of several possible types of connectors and not a *crumb*; these will be processed in the next set of steps. All remaining voxels that have not been classified into a morphological class by this point are assigned the generic morphology class 5, indicating that they are connectors that need to be further divided into *antenna*, *bond*, or *circuit* classes.

The fifth step is to identify *antenna*, or protrusions from a *mass* that connect to only a single point on a *mass*. A *mass* may have more than one *antenna*, and a discrete object may comprise more than a single *mass*; thus, identification of connected voxels that contain no *mass*, but connect to a *skin* only once, requires an iterative looping and decomposition of the graph and repeated tests of how many voxels in candidate *antennae* connect to *skin*. When the number of connections is one, then the corresponding connected voxels are given the morphology code 6.

The sixth step is to identify *circuit* voxels as those connectors that connect a *mass* to itself along a connected set of voxels. Once *circuit* voxels are identified, the remaining connectors, are by definition, *bond* voxels – those connectors that join two or more discrete *mass* clusters within a single object. All neighbours to the *skin* voxels that are not *mass,* or *antennae,* are added to a list of possible *circuit* voxels and their adjacencies are tested to identify whether connected voxels are also candidates. If these contender voxels are connectors, they are coded as circuits and the remaining connectors are classified as *bond* morphology and given the code 7. The *bond* connectors identify those voxels that connect at least two *masses* within an object.

The seventh step is to assign a morphology code of 1 to all remaining voxels to identify them as being on the *outside* of the feature of interest. The original input data, the corresponding network graph object, the final morphological segmentation, and maps of discrete object and voxel IDs are organized and packaged into a single list object along with summary statistics (Table1) and returned to the user. The summary statistics provide the number of voxels and their relative percentages that have been allocated to each of the morphological classes. Production of a final plot is optional for the interactive viewing the resulting morphological classification (Figure 3). Plots of each morphological element class are also available options.

Table 1. Output summary statistics stemming from the morphological segmentation of the feature presented in Figure 1a.

Code	Description	NVoxels	Percentage
1	Outside	758	75.8
2	Mass	69	6.9
3	Skin	130	13.0
4	Crumb	2	0.2
5	Circuit	18	1.8
6	Antenna	7	0.7
7	Bond	4	0.4
8	Void-volume	2	0.2
9	Void	10	1.0

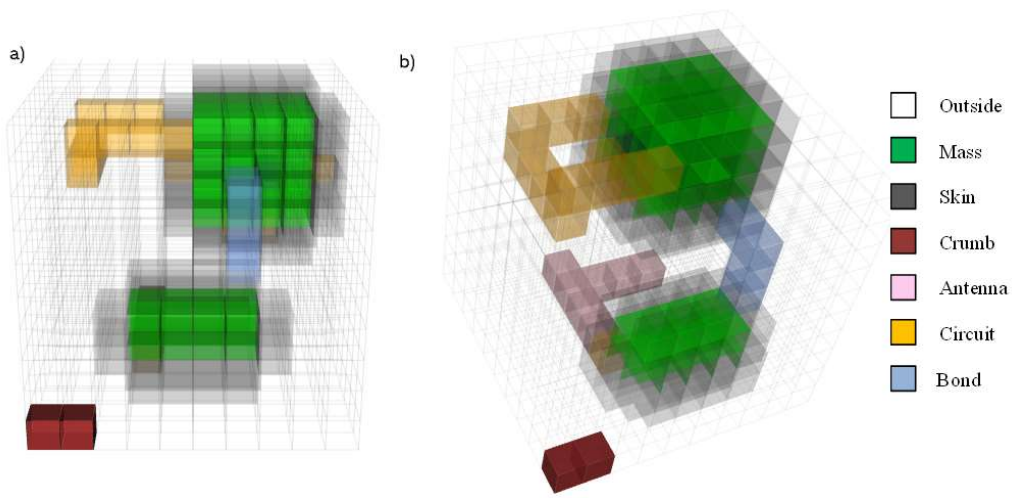

Figure 3. The final 3D morphological segmentation of the data in Figure 1a, shown from two different perspectives (a, b). The *void* and *void-volume* classes are not directly visible because they are contained within the green upper mass cluster.

Each discrete object is stored as a sub-graph with unique voxels identified by unique IDs. Orthogonal connectivity among feature voxels is maintained as undirected neighbours that connect adjacent voxel IDs (Figure 4). This paired representation permits efficient 3D visualization with the shaded voxels across the entire input data while also facilitating rapid adjacency and connectivity testing with existing graph theory tools. The graph theory tools for assessing adjacency for answering questions

such as “is this voxel connected to that voxel?” or “which are this voxel’s spatial neighbours?” are particularly powerful in this context. Graph theory also allows objects with multiple *mass* cores to be quickly assessed together for connector classes that extend outwards from *skin* voxels.

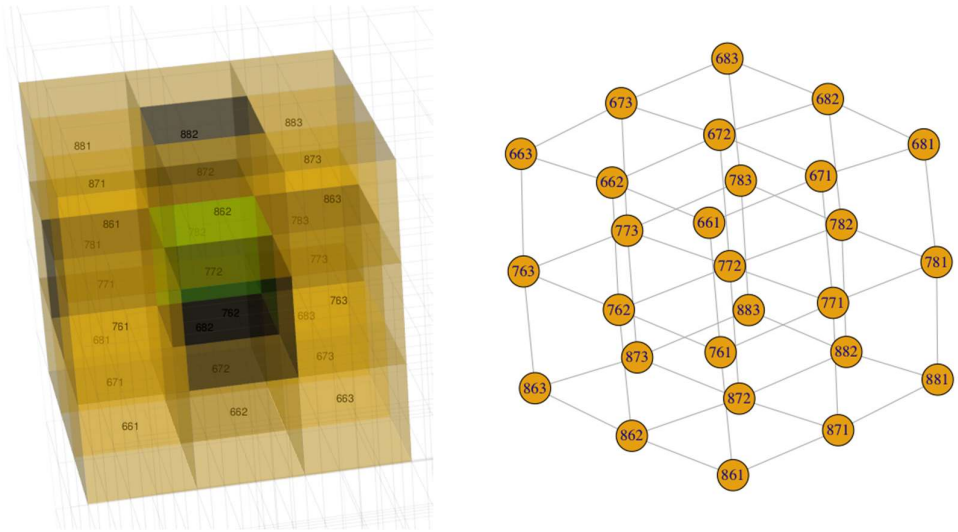

Figure 4. An example of a discrete object is shown here enlarged (left), with its corresponding graph (right). The numbers represent the unique voxel IDs, and while the two views are rotated relative to each other, the topological relationships are maintained.

From Figure 4, it is evident that the mass voxel is 772, and that from the graph representation, its skin neighbours must be 773, 771, 782, 762, 672, and 872 (which can be discerned from the block figure but with greater difficulty). The network does not connect to the outside voxels, which were coded as zeros (0) initially and deleted from the overall network graph that contained all sub-objects. This simplicity also allows the number of discrete objects to be counted and processed separately. Figure 5 shows the individual morphological elements for the feature of interest in Figure 1, such that the individual components are visible without being occluded by the simultaneous display of all elements simultaneously.

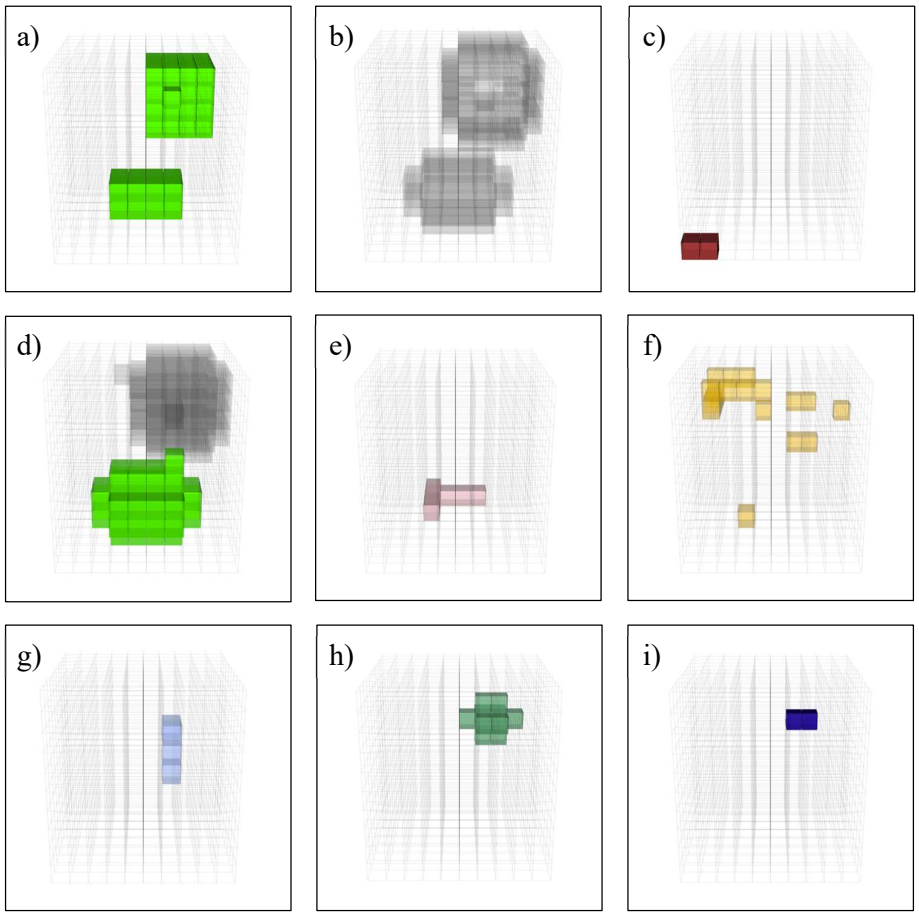

Figure 5. Individual plots for the morphological elements: (a) *mass*, (b) *skin*, (c) *crumb*, (d) expanded *mass* cores – used for distinguishing *circuits* from *bonds*, (e) *antenna*, (f) *circuit*, (g) *bond*, (h) *void*, and (i) *void-volume*.

Discussion

Any spatial feature that can be represented as a 3D binary array of voxels can now be automatically segmented into its constituent morphological elements with this tool, thereby extending the capability previously available only for 2D data. The algorithm will assign each voxel of interest to one of 9 mutually exclusive morphological elements (*outside*, *mass*, *skin*, *crumb*, *antenna*, *circuit*, *bond, void-*

volume, or void). The implementation of this segmentation allows 3D data to be partitioned into categories analogous to their 2D counterparts. These structural classes can be used to describe the relative contributions to shapes, objects, or features, but also be used to partition further analysis within and among these classifications. Examples of ecologically-relevant analyses include comparing species richness between *mass* and *skin* classes, testing whether *mass* habitat is connected to other *mass* habitat, or comparing *mass*-to-*skin* ratios among objects.

Implementation of the 3D morphological segmentation procedure is in an open and free software environment and the code is being made publicly available via CRAN (https://cran.r-project.org/) and is available from the author. While variations exist on how the segmentation functionality is applied and displayed, most users will be able to perform the segmentation by providing the one required binary 3D array of feature voxels (e.g., *incube*) and making a single function call:

```
outobj <- morph3d(DATACUBE=incube, FINALPLOT=TRUE)
```

The output object, *outobj* in this case, is a list object that contains the input data cube, the network graph object, 3D array objects for unique voxel IDs and discrete object IDs, the morphological segmentation as an array of 3D voxels, and a data frame containing the summary statistics of the segmentation among other state variables. The output ensures that related data remain together and connected, allowing for context and simplicity in further analyses, while ensuring an organized workspace.

Declarations

Funding

This work was funded by a Natural Sciences and Engineering Research Council (NSERC) Discovery Grant to the author

Conflicts of interest/Competing interests

There are no conflicts of interest or competing interests

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Consent for publication

As the only author of this short communication, I provide consent for publication.

Availability of data and material

Not applicable.

Code availability

The code for achieving 3D morphological segmentation is being prepared for free and open distribution on the Comprehensive R Archive Network. The code is also available from the author of this short communication.

Authors' contributions

I conceived the idea of 3D morphological segmentation, developed the logic, wrote the code, and developed this short communication without external assistance.

Acknowledgements

I acknowledge the valuable and constructive feedback from two anonymous reviewers whose advice and suggestions helped improve the clarity of this manuscript. I value discussions with Dr. Connie Ko who was an excellent sounding board for ideas. This work was funded through a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada.

References

Adler D, Murdoch D (2021) rgl: 3D visualization using OpenGL. Version 0.105.22URL https://CRAN.R-project.org/package=rgl

Csárdi G, Nepusz T (2006) The igraph software package for complex network research. InterJournal Complex Systems 16959

Cushman SA, McGarigal K, Neel MC (2008) Parsimony in landscape metrics: strength, universality, and consistency. Ecological Indicators 8:691–703

Frazier AE, Kedron P (2017) Landscape Metrics: Past Progress and Future Directions. Current Landscape Ecology Reports 2:63–72. https://doi.org/10.1007/s40823-017-0026-0

Guerra-Hernández J, Cosenza DN, Rodriguez LCE, et al (2018) Comparison of ALS- and UAV(SfM)-derived high-density point clouds for individual tree detection in Eucalyptus plantations. International Journal of Remote Sensing 39:5211–5235. https://doi.org/10.1080/01431161.2018.1486519

Kedron P, Zhao Y, Frazier AE (2019) Three dimensional (3D) spatial metrics for objects. Landscape Ecol 34:2123–2132. https://doi.org/10.1007/s10980-019-00861-4

Li HB, Wu JG (2004) Use and misuse of landscape indices. Landscape Ecology 19:389–399

Liu M, Hu Y-M, Li C-L (2017) Landscape metrics for three-dimensional urban building pattern recognition. Applied Geography 87:66–72. https://doi.org/10.1016/j.apgeog.2017.07.011

Neel MC, McGarigal K, Cushman SA (2004) Behavior of class-level landscape metrics across gradients of class aggregation and area. Landscape Ecology 19:435–455

Nowosad J, Stepinski TF (2021) Pattern-based identification and mapping of landscape types using multi-thematic data. International Journal of Geographical Information Science 1–16. https://doi.org/10.1080/13658816.2021.1893324

Pflugmacher D, Cohen WB, Kennedy RE, Yang Z (2014) Using Landsat-derived disturbance and recovery history and lidar to map forest biomass dynamics.

Remote Sensing of Environment 151:124–137. https://doi.org/10.1016/j.rse.2013.05.033

Popescu SC, Zhao K (2008) A voxel-based lidar method for estimating crown base height for deciduous and pine trees. Remote Sensing of Environment 112:767–781

R Core Team (2021) R: a language and environment for statistical computing. Version 4.0.4. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Rahman MdS (2017) Basic Graph Theory. Springer International Publishing, Cham

Remmel TK (2020) Distributions of hyper-local configuration elements to characterize, compare, and assess landscape-level spatial patterns. Entropy 22:420. https://doi.org/10.3390/e22040420

Remmel TK, Csillag F (2003) When are two landscape pattern indices significantly different? Journal of Geographical Systems 5:331–351

Remmel TK, Fortin M-J (2013) Categorical, class-focused map patterns: characterization and comparison. Landscape Ecology 28:1587–1599

Riitters KH, O'Neill RV, Hunsaker CT, et al (1995) A factor analysis of landscape pattern and structure metrics. Landscape Ecology 10:23–39

Soille P, Vogt P (2009) Morphological segmentation of binary patterns. Pattern Recognition Letters 30:456–459. https://doi.org/10.1016/j.patrec.2008.10.015

Szpakowski D, Jensen J (2019) A Review of the Applications of Remote Sensing in Fire Ecology. Remote Sensing 11:2638. https://doi.org/10.3390/rs11222638

Uuemaa E, Antrop M, Roosaare J, et al (2009) Landscape metrics and indices: an overview of their use in landscape research. Living reviews in landscape research 3:1–28

Vogt P, Riitters K (2017) GuidosToolbox: universal digital image object analysis. European Journal of Remote Sensing 50:352–361. https://doi.org/10.1080/22797254.2017.1330650

Vogt P, Riitters KH, Estreguil C, et al (2007) Mapping spatial patterns with morphological image processing. Landscape Ecology 22:171–177. https://doi.org/10.1007/s10980-006-9013-2

White JC, Wulder MA, Hermosilla T, et al (2017) A nationwide annual characterization of 25 years of forest disturbance and recovery for Canada using Landsat time series. Remote Sensing of Environment 194:303–321. https://doi.org/10.1016/j.rse.2017.03.035

Wickham H (2007) Reshaping data with the reshape package. J Stat Soft 21:1–20. https://doi.org/10.18637/jss.v021.i12

Wickham H (2019) stringr: Simple, Consistent Wrappers for Common String Operations. Version R package version 1.4.0URL https://CRAN.R-project.org/package=stringr

Ye H, Yang Z, Xu X (2020) Ecological Corridors Analysis Based on MSPA and MCR Model—A Case Study of the Tomur World Natural Heritage Region. Sustainability 12:959. https://doi.org/10.3390/su12030959