

A SCRIPT FOR THE AUTOMATIC DELINEATION OF CANOPY GAPS FROM
RASTER IMAGES AND INCORPORATION OF ANALYSIS WITH ONTARIO'S
FRI DATASET

By

Robert Sweeney

An undergraduate thesis submitted in partial fulfillment of the requirements for the
degree of Honours Bachelor of Science in Forestry

NRMT4010, Thesis
Dr. Leni Meyer and Dr. Kevin Crowe

Faculty of Natural Resources Management
Lakehead University

April 25th, 2019

Major Advisor

Second Reader

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for the HBScF degree at Lakehead University in Thunder Bay, I agree that the University will make it freely available for inspection.

This thesis is made available by my authority solely for the purpose of private student and research and may not be copied or reproduced in whole or in part (except as permitted by the Copyright Laws) without my written authority.

Signature: _____

Date: _____ April 26th, 2019 _____

A CAUTION TO THE READER

This HBScF thesis has been through a semi-formal process of review and comment by at least two faculty members. It is made available for loan by the Faculty of Natural Resources Management for the purpose of advancing the practice of professional and scientific forestry.

The reader should be aware that the opinions and conclusions expressed in this document are those of the student and do not necessarily reflect the opinions of either the thesis supervisor, the faculty, or Lakehead University.

ABSTRACT

Sweeney, R. 2019. A script for the automatic delineation of canopy gaps from raster images and incorporation of analysis with Ontario's FRI datasets.

Keywords: ArcPy, canopy gap, CHM, DEM, DSM, FRI, Python, script, tool

This thesis project serves as a tool for resource managers looking to delineate gaps in the forest canopy, where a Canopy Height Model raster is available. Once identified, analysis is performed in order to assess the prevalence and distribution of canopy gaps within a forest stands. Results are written to Ontario FRI data of the corresponding area.

CONTENTS

LIBRARY RIGHTS STATEMENT	ii
A CAUTION TO THE READER	iii
ABSTRACT	iv
TABLES	vii
FIGURES	viii
APPENDICES	ix
ACKNOWLEDGEMENTS	x
INTRODUCTION	1
Objective	2
LITERATURE REVIEW	3
Defining Forest Canopy Gaps	3
Measuring Canopy Gaps	6
Delineating Canopy Gaps Using Script	11
METHODS	13
Modules and Tools	13
Script Overview	17
Canopy Gap Definitions	18
User Defined Functions	19
Script Progression Section 1: Detection of Canopy Gaps	21
Script Progression Section 2: Statistical Analysis	25
Script Progression Section 3: Incorporation with FRI Datum	29
RESULTS	31
Visual Results	31
Testing with Random Numbers	34
Assessment of Gap Prevalence Results	36
Assessment of Gap Clustering Results	38

DISCUSSION	41
Functionality	41
Improving FRI Data Richness	41
Opportunities for Customization	43
Opportunities for Improvement	46
CONCLUSION	51
LITERATURE CITED	53

TABLES

Table	Page
1. Imports made in order for script to run	14
2. Field names and datatypes for fields created in script	30

FIGURES

Figure	Page
1. CHM1 Raster used for development of script	16
2. FRI1 Feature class used for development of script	17
3. User defined function <i>seedcell</i>	19
4. User defined function <i>region_growing</i>	20
5. Hypothetical <i>Array 1</i> used in methods example	21
6. Hypothetical <i>Array 2</i> used in methods example	22
7. Hypothetical “ <i>rgrwF</i> ” array used in methods example	23
8. Hypothetical “ <i>RasterToPoly1</i> ” polygon used in methods example	24
9. Update cursors used to restrict gap polygon size	25
10. Example of <i>intersect</i> method as used in script	26
11. Formula for “Do” calculation used in ANN function	28
12. Formula for “De” calculation used in ANN function	28
13. Formula for “Se” calculation used in ANN function	28
14. Formula for “Z” calculation used in ANN function	28
15. ANN Function implemented in script	29
16. 3D Visualization of CHM1 without gaps identified	32
17. 3D Visualization of CHM1 with gaps identified	32
18. 3D Visualization of CHM2 without gaps identified	33
19. 3D Visualization of CHM2 with gaps identified	33
20. Script tested with random numbers set 1	35
21. Script tested with random numbers set 2	36
22. Example of results in “GapCoverage” field	37
23. Example of results in “NumberOfGaps” field	38
24. Example of “random” gap distribution	39
25. Example of “clustered” gap distribution	40
26. Example of “dispersed” gap distribution	40
27. Example of connected gap identification	49
28. Hypothetical gap shapes leading to errors in clustering classification	50
29. Error associated with Area values used in clustering calculations	51

APPENDICES

Appendix	Page
I: Script	58

ACKNOWLEDGEMENTS

I would like to thank my second reader, Tomislav Sapic for his expertise and guidance throughout the entirety of this project. He offered solutions to the seemingly endless stream of problems encountered during the development of this script.

Additionally, thank you to Dr. Ulf Runesson for taking this project on as a first reader.

INTRODUCTION

Canada's Boreal forests are complex systems, with their health and successional trajectories being heavily influenced by stochastic disturbance events. Depending on the type and severity of these disturbances, forest structure can be changed drastically with the formation of canopy gaps. These gaps have many implications on ecosystem processes including changes in the successional pathways of forest vegetation, alterations in nutrient regimes, and changes in the behavior of inhabiting fauna. This is demonstrated in old-growth forests, where gap dynamics are the drivers of forest succession in the absence of large scale disturbance (Vepakomma et al. 2007). More generally, alterations in light availability caused by canopy gaps have led to a reestablishment of shade-intolerant tree species, instead of the expected transition to late successional species dominance (Bergeron 2000). Increased decomposition of foliar litter has been shown to increase abundance and availability of carbon, nitrogen, and phosphorous in canopy gap soils (Ni et al. 2018). Lastly, wildlife habitat diversity as well as species diversity increase following the formation of canopy gaps due to increased structural heterogeneity (Muscolo 2014). The vast number of implications that canopy gaps have on forest communities is a testament to the importance of understanding these entities when examining forest ecosystems.

The availability of spatially overt data on forest conditions, composition, and availability of natural resources within a management unit is critical if forest managers are to make informed forest management decisions affecting large geographical regions (Desclée et al. 2006). Providing an accurate representation of the canopy gaps present in a management unit is one way of providing decision makers with this information. As of

now, Ontario's Forest Resource Inventory data lacks quantitative descriptions of forest canopy gaps. Furthermore, qualitative descriptions are limited to the "HORIZ" field, representing horizontal stand structure (OMNR 2009). The script developed in this project will serve managers in the automatic delineation of forest canopy gaps, and improve the richness of available FRI data through both quantitative and qualitative analysis of the identified gaps.

The prevalence of canopy gaps within a forest stand is measured through their abundance and size. Additionally, their dispersion within a stand will also be assessed. With the confirmation of LiDAR data acquisition for the Province of Ontario (Kuttner 2018), this tool may be further improved upon as the quality of available spatial data is improved with higher resolution LiDAR.

OBJECTIVE

The objective of this thesis project is to develop a tool for use by forest managers that will allow for accurate automatic delineation of forest canopy gaps using publicly available datasets. In addition to this, I intend to improve my familiarity and proficiency with the Python programming language.

As the objective of this thesis is to develop a tool, as opposed to answer a research question I have no hypothesis.

LITERATURE REVIEW

DEFINING FOREST CANOPY GAPS

Forests are highly complex systems subject to stochastic variation and disturbance. As such, the formation and persistence of canopy gaps is a highly variable process, leading to inconsistencies in their expression. They vary in their size, shape, frequency, and ecological effects (Muscolo et al. 2014; Schliemann and Bockheim 2011). Thus, assigning them a static definition is difficult. Several descriptions have been offered within associated scientific literature -- some present subjective descriptions, while others offer objective requirements.

Qualitative Definitions

In some cases, all that is required is a qualitative depiction to describe a canopy gap. This appears more commonly in studies concerned with the ecological implications of forest canopy gaps, as opposed to works assessing physical attributes such as their sizes and orientations (Schliemann and Bockheim 2011). Depending on the purpose and location of a study, the exact definition of what constitutes a “canopy gap” may vary. An initial definition was offered in Runkle (1981), who described them as “the ground area in a canopy opening extending to the bases of trees surrounding the opening”. This definition was later changed to be descriptive of what became known as ‘expanded gaps’ (Runkle 1982), when Brokaw (1982) defined a canopy gap as “The vertical projection of the canopy opening from the forest canopy to the ground”. As holes in the forest canopy are generally a result of one or more trees dying to disturbance (Whitmore 1989), some authors have included formation due to natural disturbances within their qualitative descriptions (Vepakomma 2008; Vehmas et al. 2011; Muscolo et al. 2014).

For studies focused on forest gap dynamics, Zhu et al. (2015) suggest an abstract definition of a “contiguous open area of sufficient size to create forest understory conditions that are functionally different from smaller canopy openings”. This may be in the availability of light, nutrients, moisture, for example. While this definition appears thorough and practical, it may prove difficult to apply on the ground.

In some cases, such as studies interested in temporal variability of canopy gaps, it may be important to subdivide distinguished canopy gaps into one of two classifications identified by Schneider and Larson (2017). They differentiate between developmental, and edaphic gaps. Developmental canopy gaps result from the natural succession of forested land, due to overstory tree death. These gaps will vary in size and shape over time (Vepakomma et al. 2008). Alternatively, edaphic canopy gaps are effectively permanent as they result from site conditions inhibiting tree establishment, such as wetlands or rock outcrops (Schneider and Larson 2017). The above descriptions offer guidelines for the identification of canopy gaps, however they are subject to an inherent need for user interpretation due to a lack of qualitative requirements.

Quantitative Definitions

To automate the delineation of forest canopy gaps, quantitative values must be included in the description of said gaps. Again, these values may vary between projects, as well as type and height of forest (Vepakomma et al. 2008). A critical regeneration height refers to a maximum trees height value at which a point may be considered a canopy gap. These heights may be absolute or relative (Vepakomma 2008). For example, Brokaw (1982) suggests a lower limit of 2 m, Bonnet et al. (2015) uses 3 m, Nakashizuka et al. (1995) uses 15 m, while 5 m is used by Seidel et al. (2015) and

Vepakomma et al. (2008), with the latest adding the additional requirement of 80% of perimeter trees being 10m tall. In some cases, a relative critical threshold is used, such as the requirement of regenerating vegetation to be one-half to two-thirds of the height of the surrounding canopy (Tyrrell and Crow 1994). In Ontario, the 3 m height requirement for a stand to be declared “Free-to-Grow” is perhaps a reason to use 3 m as a critical threshold.

Gap area is another metric by which a gap may be quantified. Again, lower and upper area limits vary between authors. Generally, an upper limit of 1000m² is set for a canopy break to be considered a gap (Schliemann and Bockheim 2012; Yamamoto 1992). This value is used as beyond this size canopy breaks exhibit characteristics that are deemed substantially different from those exhibited by smaller canopy gaps, in terms of increased light penetration, and soil temperatures (Zhang and Zak 1995; Muscolo et al. 2014). Lower limits can vary heavily, largely due to the quality of data being used for their identification. For example, Schneider and Larson (2017) offer no minimum area, while Getzin et al. (2012) use 1 m², Vepakomma (2012) uses 5 m², and Bonnet et al. (2015) and Koukoulas and Blackburn (2004) use 50 m². In some cases, it may be important to exclude expected gaps based on their shape, as long narrow gaps may exhibit properties that vary from those seen in rounder gaps, due to their perimeter to area ratio and subsequent shading (Gray et al. 2002). For this reason, minimum gap widths may be applied, such as 1.5 m or 2 m as used in Vepakomma (2012) and Bonnet et al. (2015), respectively. Ideally, a program written to identify canopy gaps should allow for user inputs to set upper and lower bounds, between which a break in the forest canopy may be considered a “gap”.

MEASURING CANOPY GAPS

Gap detection and delineation of its boundaries using any technique is a complex task (Vepakomma et al. 2008). As such, many methods have been developed and attempted to offer quantitative metrics regarding forest canopy gaps. It is important to note that the definitions used to identify gaps, as discussed above, will have a substantial impact on the effectiveness of the measurement technique being used (Nakashizuka et al. 2015). Additionally, the ‘optimal’ technique has been shown to vary based on forest type, structure, and location (Bagaram et al. 2018). As technologies have progressed over time, new remote approaches to canopy gap identification have been developed in lieu of the traditionally used manual methods.

Manual methods of canopy gap measurement

Traditionally, the measurement of forest canopy gap size has been done using manual measurement techniques which require extensive field work. Several methods have been employed to accomplish this task. Most methods focus on the traditional “canopy gap” definition, that is the projection of a two-dimensional shape from the canopy to the ground (Schliemann and Bockheim 2011). There are three main strategies for taking two-dimensional measurements: 1) Making just two perpendicular measurements if the gap is assumed to be of uniform, elliptical shape, 2) Several measurements may be taken from the center point if the gap is assumed to be of an irregular shape, 3) Photographic data may be collected along with measurements used for establishing a scale to be used in photographic analysis (Schliemann and Bockheim 2011). The first two methods have been employed in several studies (Stewart et al. 1991; Runkle 1992; Bartemucci et al. 2002), however they are touted as being time

consuming and physically challenging in difficult terrain (Betts et al. 2005). Furthermore, Betts et al. (2005) claims that for small gaps, or those with heavy understory vegetation, the identification of canopy gaps may be difficult, resulting in inconsistencies in acquired data. The third method, incorporating photographic data in canopy gap measurement, was employed by Hu and Zhu (2009). They used a series of hemispherical photographs combined with horizontal measurements at multiple heights to quantify gap sizes. However, a criticism of this method is the variability in heights at which the horizontal measurements were taken (Seidel et al. 2015). A final, simplistic measure of gap shape is a comparison of gap width to surrounding canopy height, however this method is only suggested when the gap in question is of a regular, rounded shape – often seen in man-made canopy gaps (Gray et al. 2002). Unfortunately, the results of ground surveys may vary based on the surveyor's definition of what constitutes a gap, leaving consistency to be desired (Nakashizuka et al. 1995). Ultimately, the manual measurement of forest canopy gaps through field work may appear straightforward, but the high time and labour costs, combined with the subjective nature of canopy gap measurement makes it sub-optimal.

Another physical characteristic of canopy gaps that has been studied is their distribution, and implications with respect to a landscape (Betts et al. 2005). This has been accomplished in several ways. Lorimer and Frelich (1989) took an unconventional approach, of inventorying tree species, age, and size for all trees greater than 1.4 m in height at each of 70 random plot locations. Working under the assumption that the Northern-hardwood-hemlock system they were studying only enables canopy trees to have germinated in or released from canopy gaps, the ages of measure trees allowed

them to identify the timing of gap formation and gap-causing canopy disturbance over time. A simpler method has been employed in studies such as (Brokaw 1982), and (Runkle 1982), which utilize strip transects (Schliemann and Bockheim 2011). By recording the length along each transect that falls within a canopy gap compared to the transects total length, Runkle (1982) calculated the fraction of the entire forest in gaps. In addition, he recorded the size and shape of each gap encountered to establish overall gap extent and size distribution (Runkle 1982).

A final manual method of canopy gap delineation is offered by Kathke and Bruelheide (2010). They used old aerial photographs, georeferenced in ArcGIS 9.0, to locate canopy gaps and allow for comparisons to be made between time periods. Their method included manual analysis of the photographs, followed by the drawing of polygons within the ArcGIS software. These vector files were then used to provide spatial statistics on the gaps themselves. While all the manual methods for canopy gap detection may accomplish their purpose, they may all be criticized for being time and labour intensive, and for lacking reliability due to inconsistencies in technique (Schneider and Larson 2017). For this reason, remote, automated method for canopy gap delineation may prove superior.

Remote sensing based methods of canopy gap measurement

The automation of forest canopy gap delineation may serve to save time when compared to employing traditional manual techniques (Betts et al. 2005). Many options for this remote analysis are available, however the best methods may vary based on forest type and data availability. A key component in determining what data may be reasonably used when employing a method is its resolution. Coarser data makes it

increasingly difficult to identify and accurately quantify smaller gaps (Malahlela et al. 2014).

The identification of canopy gaps using remote sensing has been occurring for several decades (Foster and Reiners 1986; Koukoulas Blackburn 2004; Torimaru et al. 2012). Unfortunately, their use is often limited to the two-dimensional projection definition offered by Brokaw (1982), and are unable to measure the size of the “extended gap”. One successful approach to identifying canopy gaps was the semi-automated method proposed by Betts et al. 2005. By applying a coarse filter to a high-resolution Digital Elevation Model (DEM) they could identify gaps through an analysis of the differences between DEM’s before and after applying the filter. A large difference seen in the same relative location following filter implies a particularly low return surrounded by many, higher points. This is indicative of a canopy gap (Betts et al. 2005). A notable feature of this program is that it requires user input to “train” the algorithm to identify gaps correctly, per the user. This was accomplished by allowing for the adjustment of the filter size. Again, this method fails to meaningfully measure gap shape, due to the relatively low resolution of available data.

The advent of LiDAR (light detection and ranging) has attracted attention in the field of natural resources management (Vepakomma et al. 2008). LiDAR sensors produce very dense and accurate point clouds that can be used to measure three dimensional structures of forest canopy and sub-canopy, including estimates of vegetation height, cover, canopy structure (St-Onge et al. 2004). This form of data is very popular for the measurement of canopy gaps due to its accuracy and dense coverage. Lidar data may be analyzed in the form of a point cloud, or a canopy height

model (CHM) – similar to a DEM, but measuring distances above the ground irrespective of elevation. Using a canopy opening algorithm created by Churchill et al. (2017), Schneider and Larson (2017) used point cloud Lidar data to measure both regular and extended canopy gaps. This was accomplished by providing the option to remove forest canopy returns between the initially detected gaps, identified by ground returns, and surrounding tree boles. Gaulton and Mathus (2010) compared Lidar point cloud and CHM data for use in identifying canopy gaps in plantations. When compared to manually measured canopy gaps in the same location, the point cloud data set identified gaps with 78% accuracy, while the CHM data provided 75% accuracy. Hunter et al. (2015) used multi-temporal point cloud Lidar data to successfully follow canopy growth, shrinkage, closure, and creation over time. An object-based image analysis (OBIA) approach has recently been adopted when using CHM-style Lidar data. This method requires the object be segmented into “objects” of similar value for study, as opposed to pixel-by-pixel analysis. The high accuracy and resolution that Lidar data provides makes it the best option for topological analysis, which is often the foundation of automated methods of canopy gap identification and measurement.

Measuring gap shape

It has been made evident by the literature that simply measuring the size of forest canopy gaps and recording their locations is insufficient in addressing many other ecologically important dynamic characteristics of gaps (Vepakomma et al. 2008). The shape of a canopy gap plays a role in determining the impacts said gap will impose on the forest from an ecological perspective. For example, rounder gaps will allow for more photosynthetically active radiation to reach the forest floor due to a reduction in shadow

cover (Canham 1988). Generally, gap shapes are classified based on similarities to a standard geometric form (Seidel et al. 2015), however this method has been criticized heavily due to the inaccuracies associated with estimation (Schliemann and Bockheim 2011). Many geometric forms have been used to describe canopy gap shape, including “dumb-bell” (Oldeman 1978), and “ellipses” (Runkle 1981). Salvador-Van Eysenrode et al. (1998) offered 17 different options. In some cases, irregularly shaped gaps are divided into several smaller sections for measurement (Green 1996). Attempts have also been made to measure gap shape by assessing the fractal dimensions expressed by surrounding canopy trees. While this has potential to be highly accurate, it is highly time consuming (Zhu et al. 2014). Finally, there are some rather simplistic methods for quantifying gap shape. Several studies have used “gap ratio”, that is the gap diameter compared to the height of surrounding canopy trees as a measure of three-dimensional shape and size (Spies et al. 1990; Schliemann and Bockheim 2011). Alternatively, Gap Shape Complexity Index (GSCI) offers a value representing the relative complexity of a shape compared to a circle, where shapes with a higher value are “more irregular”. Finally, a simple comparison of shape perimeter to area offers results like those received from calculating a GSCI value (Vepakomma et al. 2008). Again, the inherent variability associated with canopy gap development prevents the determination of a single “best” method for measurement. This remains true for the assessment of their shape.

DELINEATING CANOPY GAPS USING SCRIPT

The automation of any manual process serves to reduce the amount of time and money spent performing the task. The development of customized software can facilitate this automation. Python, an object-oriented programming language, is quickly

growing in popularity particularly for new programmers due to its intuitive nature (Kuhlman 2012). Being a general-purpose programming language, Python is suited for interactive work and quick development of “one-off” programs known as scripts (Esri 2018) A common approach to remote sensing based forest canopy gap detection is to perform a topological analysis. Within Python there are several modules available that may facilitate this type of investigation.

ArcPy is a Python module that was developed to provide an efficient way to perform geographic data analysis, conversion, management, and map automation. Through ArcPy, a user may access various geoprocessing tools found within Esri’s ArcGIS software, in addition to other functions, classes, and modules (Esri 2018). Depending on the approach taken and the desired outcome when analyzing forest canopy gaps, one may find value in ArcPy due to its capacity to manage geographical data, and subsequent compatibility with ArcGIS.

A sensible way to display topological data is through a two-dimensional dataset containing the height value at a given location, such as a DEM or CHM. NumPy, another Python module, is a fundamental package for scientific computing within the language (NumPy 2018). The primary object used within NumPy is its n-dimensional array. This is a way to store and manipulate data in any number of dimensions (NumPy 2018). An obvious function for this object for the delineation of forest canopy gaps is the ability to perform computations on individual cells with respect to surrounding values. The capacity to iterate over large sets of data, such as those representing forest topology, makes NumPy an efficient and powerful module for the analysis of multi-dimensional data. As a division of SciPy, a Python-based ecosystem of open-source

software, NumPy is highly compatible with other packages contained within this network (SciPy Developers 2019). SciPy offers several data analysis packages referred to as SciPy Toolkits (Scikits). For example, the *generic filter* Scikit allows for computations on a given cell, based on the data surrounding it (SciPy Community 2014). The SciPy module is a powerful tool for data processing, particularly for data stored in a two-dimensional form, such as an array.

As of now, there is limited use of the Python programming language in the scientific literature surrounding forest canopy gap delineation. One relevant program is SEGMA – a Python program used for the delineation of tree crowns from airborne lidar CHM's (St-Onge 2018). Although the program segments tree crowns, the implementation of topological analysis through Python programming, as well as gathering an understanding of the modules used may prove useful in establishing an approach to the delineation of the gaps themselves. While ArcPy is not used by St-Onge, the SciPy module is. Examples include the use of a Gaussian filter to remove abnormalities from the dataset prior to any computation, as well as statistical analysis' utilizing NumPy's n-dimensional array functionality (St-Onge 2018). An understanding of the SEGMA program may prove valuable due to the similarities in data used (CHM's) as well as the modules used.

METHODS

MODULES AND TOOLS

The program was written in the Python 2.7.10 programming language. This script relies on functionalities contained within several external Python modules and tools. Table 1 lists all of the imports made at the beginning of the script and their

respective uses. Due to the continued development of the script, any references to code lines are relevant only to the Figures presented. Should the user wish to alter the script, the relevant line content should be located, rather than the line number.

Table 1: List of all imports made in order for script to run

Tool or Module	Purpose
Numpy	Array analysis
Arcpy	Geospatial processing
Arcpy (env)	Setting arcpy work environment
Arcpy Spatial Analyst toolbox	Accessing tools from the Spatial Analyst toolbox in ArcMap
Scipy	Image analysis
Scipy.stats	Quantitative image analysis
os	Accessing functions dependant on the user machine's operating system
Tkinter	Access user interface tools
Math	Statistical computing

DATA

The purpose of the program is to analyze raster datasets representing a forest canopy height model (CHM), identify canopy gaps based on specific criteria, determine their geographic extent, quantitatively and qualitatively describe them, and add the descriptions to the Ontario forest resource inventory (FRI) GIS dataset (OMNR 2009).

The Python program was developed using a sample CHM raster dataset as well as an Ontario FRI feature class encompassing the same area. CHMs can be produced from two publicly available datasets, a digital elevation model (DEM) and a digital surface model (DSM). A DEM is representative of the elevation of bare ground, while a DSM represents the elevation of the highest detectable surface at a given point whether it be bare ground, or a vertical obstruction. The DEMs used were taken from the OMNRF's North Western Ontario Orthophotography Project (NWOOP). This dataset has a ground sampling distance (GSD) of 2 meters, and comes in 1 km by 1 km tiles (OMNRF 2018). The DSM data used was of 5 m GSD, derived from Ontario's eFRI ADS linear scanner imagery (GOO 2010).

The (CHM) was produced using the *Raster Calculator* (Esri 2019h) tool available in ArcGIS. By subtracting the DEM from the DSM within a forested area, the user is left with a CHM. This raster dataset provides cell values (in meters) representative of the highest detectable point above the ground, or the ground elevation if no vertical obstructions are present. Within the script negative values are removed from the CHM, as any negative values are inherently a result of error. The negative heights attained in the sample CHM's used were rare, and of negligibly small values. Figure 1 is the CHM generated as an example, for use in developing this program, referred to going forward as "*CHMI*". Cells of lighter colour are representative of higher elevation values, generally representing forest canopy, while darker cells represent lower elevations. The resolution of this raster dataset is 5 m.

In addition to the raster elevation model, the script uses an FRI feature class for the same geographic area to access the Ontario FRI information regarding the forest

stands within the area of interest. Information such as stand areas are used in calculations within the script. In addition, calculations are written to the FRI feature class attribute table, offering an easily interpretable method of presenting results. An example of an FRI file can be seen in Figure 2; This FRI feature class will be referred to as “*FRII*”. The area shown is the same as that of the CHM seen in Figure 1.

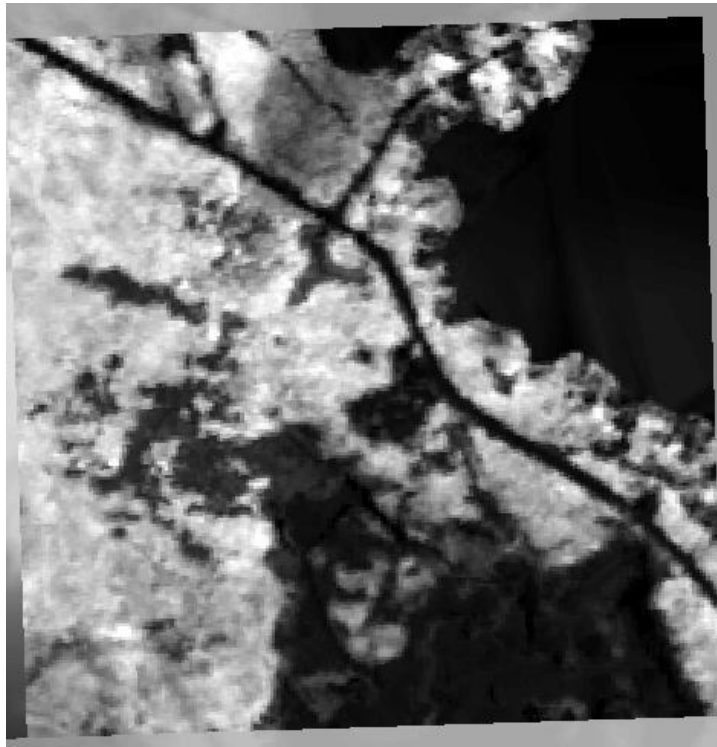


Figure 1: CHM Generated from subtracting a DEM from a DSM of the same area, representing an area west of Thunder Bay. Resolution of 5 m.

Data source: (GOO 2010; OMNRF 2018)

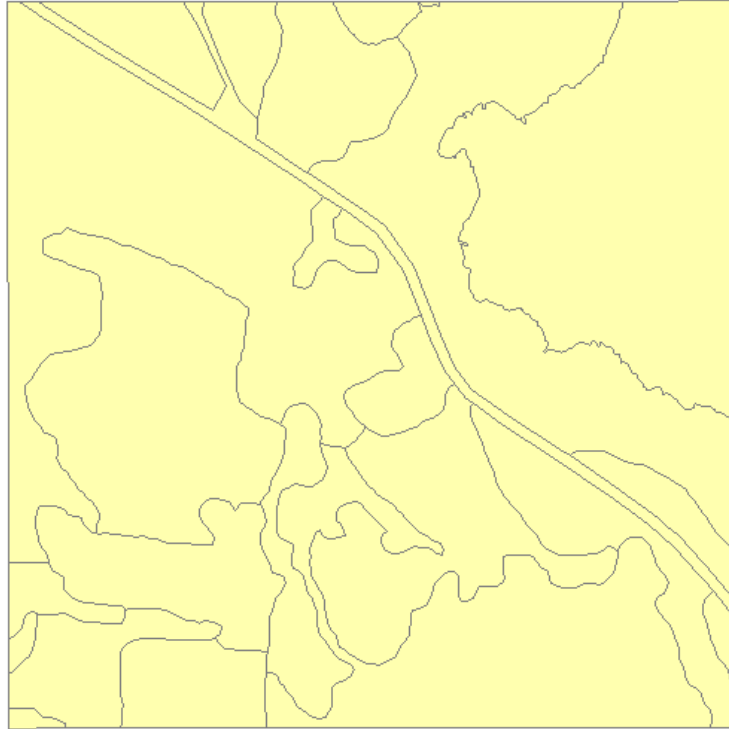


Figure 2: Polygon feature class “FRI1” required as script input.

Data source: (OMNR 2009).

SCRIPT OVERVIEW

The script intends to serve resource managers through the automatic delineation and description of canopy gaps within an area of interest. It may be separated into three major sections: gap detection, statistical analysis, and incorporation into Ontario’s FRI data. To delineate gaps in the forest canopy, topological analysis is used. Two user-defined functions, described in detail later, are implemented. They are named *seedcell* and *region_growing* and work together in the delineation of canopy gaps. Second, statistical analysis is performed on identified gaps in order to assess their dispersion within a forest stand. This is done through the approach used in ArcMap’s *Average Nearest Neighbour* (ANN) tool (Esri 2019b). Finally, information regarding the prevalence of gaps within forest stands, as well as their dispersion is added to the

provided FRI feature class attribute table. A polygon feature class is also created, with each feature representing a single canopy gap.

CANOPY GAP DEFINITIONS

A literature review of studies concerned with forest canopy gaps determined that the definition used for a “canopy gap” changes based on the focus and intent of a study. As such, this program was developed in a way that allows for customization by the user. Values that may be customized include critical regeneration heights, as well as maximum and minimum gap area requirements. A critical regeneration height refers to the height at which vegetation within the canopy gap excludes it from being described as a gap. These values may be absolute or relative to the surrounding canopy height. For the development of this script, the absolute value of 3 meters, as suggested by Bonnet et al. (2015), was used. Other absolute regeneration heights found within the literature include 2 m, 5 m, and 15 m (Brokaw 1982; Seidel et al. 2015; Nakashizuka et al. 1995). This absolute critical regeneration value was used in combination with a relative value of one half of the height of surrounding canopy trees, up to a height of 5 m, implemented by Tyrell and Crow (1994). Thus, any forested area below a height of 3 m is considered a gap, while vegetation ranging from 3 m – 5 m may be considered gaps only if the gap is bordered by trees greater than double their height. Should the critical regeneration values be changed by the user, both the *seedcell* and *region_growing* functions will need to be updated. The only requirement offered by the OMNRF within their FRI data specifications for a canopy gap is a hole in the canopy greater than 16 m² in area (OMNR 2009). The option to include gap area requirements will be discussed further.

USER DEFINED FUNCTIONS

seedcell

The user defined function *seedcell* (Figure 3) is the first step in canopy gap detection, and is used to locate “seed cells” – which are cells that are definitely representative of a canopy gap based on user requirements. It works by applying a 3x3 focal statistics filter over a subject cell within a NumPy array, and running calculations based on the neighbouring values. If a cell meets the assigned critical regeneration threshold requirements (being less than 3 m, or 5 m if the greatest neighboring value is over 10 m), it is marked as a seed cell in the output array. Otherwise, the cell receives a ‘NoData’ value of -9999 in the output array. It was designed in a way that allows for new users to adjust canopy gap definition requirements with ease, discussed later.

```

.      ###Locating Seed Cells###
- def seedcell(in_filter):
.      [a,b,c,d,e,f,g,h,i] = in_filter
.      maxFilter = max(a,b,c,d,e,f,g,h,i)
.      if -9999 in in_filter:
.          return -9999
70 elif ((e <= 3) or (e <= maxFilter/2)) and (e <= 5):
.          return maxFilter
.      else:
.          return -9999
.

```

Figure 3: User defined function *seed cell*.

region_growing

The second function used in canopy gap detection is *region_growing*. It serves to determine the true extent of canopy gaps, outside of the extent captured by the 3x3 focal statistics filter, i.e., away from the canopy trees and towards the center of the gap. Figure 4 shows the function itself, as well as its iterative implementation. For each cell previously identified as a seed cell, the function looks outward to neighbouring cells.

Using a minimum value of 0 meters, and maximum value dependent on the user's definition of a canopy gap (3 m, or 5 m should the surrounding canopy be greater than 10m tall, in this example), the function identifies neighboring cells that fall between these thresholds and marks them as gap cells.

```

    .                               ###Region Growing Function###
    .  def region_growing(img, seed, minthr, maxthr, s):
    .      img[seed] = minthr
    .      thring = (img < maxthr) & (img >= minthr)
    .      lmap, _ = label(thring, structure = s)
    .      lids = np.unique(lmap[seed])
    .      region = np.zeros(img.shape, np.bool)
    .      for lid in lids:
    .          region |= lmap == lid
    .      return region
    .
    .      #Define structure for area in which Region Growing will occur
    .      s = [[1,1,1], [1,1,1], [1,1,1]]
    .
    .      #Performing multi-index region growing function
    .      rgrwF = np.zeros(Array1.shape)
    .      it = np.nditer(Array2, flags=['multi_index'])
    .
    .      100 while not it.finished:
    .          r,t = it.multi_index
    .          it.iternext()
    .          if Array2[r,t] > 0 and Array2[r,t] < 10 :
    .              maxThresh=(Array2[r,t]*(0.5))
    .          elif Array2[r,t] > 10:
    .              maxThresh=5
    .          else:
    .              it.iternext()
    .          110 continue
    .
    .          rgrw = region_growing(Array1, (r,t),0,maxThresh,s)
    .          rgrwF = np.where(rgrw,1,rgrwF)
    .          it.iternext()
    .
    .

```

Figure 4: User defined function *region_growing*

Testing with random numbers

In order to ensure the script functions properly with respect to canopy gap identification, two test runs were completed using raster files consisting of random numbers of varying ranges, as to mimic gaps within a forest stand. Each test consisted of one range of values being used to represent a forest back ground (background range),

and a second range used within a test area in which gaps may be located (test range). The test range consisted of ten 8x8 cell squares, and an 8x60 cell rectangle.

SCRIPT PROGRESSION SECTION 1: DETECTION OF CANOPY GAPS

Detecting seed cells

To begin, the CHM raster is converted to a NumPy array, using the `arcpy.RasterToNumPyArray` function (Esri 2019i), in which array values represent the heights presented by the CHM; this is called “*Array 1*”. A down-sized example of a hypothetical *Array 1* is shown in Figure 5. This array is then altered to remove any negative values found within the CHM, giving them a value of 0. As mentioned, this is done due to negative values being a result of error.

10.0m	10.0m	10.0m	10.0m	10.0m
10.0m	A: 8.6m	B: 7.4m	C: 8.1m	10.0m
10.0m	D: 6.1m	E: 4.5m	F: 3.1m	10.0m
10.0m	G: 2m	H: 3m	I: 8.9m	10.0m
10.0m	10.0m	10.0m	10.0m	10.0m

Figure 5: Depiction of a hypothetical *Array 1*, with blue cells representing a focal statistics filter used when iteratively applying the *seedcell* function.

The user defined function *seedcell* is then iterably applied to every cell found within *Array 1*. This function evaluates every cell to determine whether it meets the

criteria required to be designated as a “seed cell”, described previously. Parallel with examining *Array 1* and detecting seed cells in it, *Array 2* is created and cross-populated in grid positions identical to the positions in *Array 1*. If a cell is determined to be a seed cell in *Array 1*, it is given, in *Array 2*, the value of its tallest neighboring cell in *Array 1*. Otherwise it receives, in *Array 2*, a ‘NoData’ value of -9999.

NoData	NoData	NoData	NoData	NoData
NoData	NoData	NoData	NoData	NoData
NoData	NoData	NoData	F: 10.0m	NoData
NoData	G: 10m	H: 10.0m	NoData	NoData
NoData	NoData	NoData	NoData	NoData

Figure 6: Depiction of a hypothetical *Array 2*, with orange cells representing “seed cells”, given the value of their tallest neighboring cell in Figure 5.

Region growing from seed cells

As mentioned, there will be many instances in which a gap extends beyond the contiguous area detected as a seed cell. In order to identify non-seed cells that also represent a canopy gap, a region growing technique is used. For every seed cell found within *Array 2* (Figure 6), *Array 1* is cross referenced in a manner that allows all neighbouring cells to be checked in a spiral pattern. Cells are checked according to the following rule:

$$H_x \leq 3 \text{ OR } H_x \leq (1/2 H_s)$$

H_x Is the height value stored in *Array 1* surrounding a seed cell (CHM value), and H_s is the height value of a seed cell in *Array 2*. If a neighbouring cell is determined to be a gap cell, all of the cells adjacent to the new gap cell are checked, and so on. Returning to Figure 6, one can see that cells F, G, and H were identified as seed cells, which will automatically become gap cells. Following the use of the *region_growing* function, cell E would also be identified as a gap cell, due to it being less than $(1/2 H_s)$ (5 m in this case). From here all seed cells and newly detected gap cells are stored in a new array of the same spatial structure (named *rgrwF* in the script, in Figure 4), where non-gap cells retain a value of NoData. Figure 7 continues with the example used in Figures 5 and 6, showing what would be produced as *rgrwF* in this hypothetical situation, where green cells are considered to be gaps.

0	0	0	0	0
0	0	0	0	0
0	0	E: 1	F: 1	0
0	G: 1	H: 1	0	0
0	0	0	0	0

Figure 7: Hypothetical *rgrwF* array created, where gap cells are highlighted in green, and non-gap cells receive “NoData” values

Creating a feature class representing gap polygons

The *rgrwF* array is converted to a raster (*rgrw_rst_INT*) using the *arcpy.NumPyArrayToRaster* tool (Esri 2019g), in which all gap cells are given a value of 1 and non-gap cells are given a value of 0. Finally, a polygon feature class is created through ArcPy's *RasterToPolygon_conversion* tool (Esri 2019j), in which all adjacent gap cells in the output polygon feature class are merged into a single unique feature, representing a gap. This feature class is identified as *RasterToPoly1* in the script, while the output is renamed to "Name of CHM" + "Gaps" using the ArcPy's *Rename_management* tool (Esri 2019k). Figure 8 represents the gap feature that would be exported as an entry to the *RasterToPoly1* feature class created in this hypothetical example.

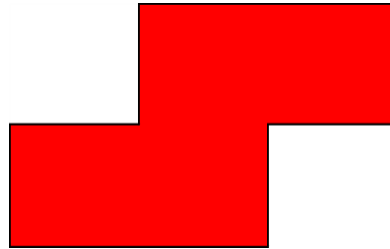


Figure 8: Hypothetical *RasterToPoly1* geometry resulting from *rgrwF* array shown in Figure 7.

Depending on the intended use of the script, the user may wish to add an upper or lower area requirement on gap polygon returns. Functionality was added to allow for this through the use of an Update Cursor (Esri 2019l). The approach to adjusting gap size limitations will be discussed further.

```

140 #This will delete any entries greater a given area value (1000m2)
    with arcpy.da.UpdateCursor("RasterToPoly1", "Shape_Area") as cursor:
        for row in cursor:
            if row[0] > 1000: #Update or delete this to change maximum gap size limit
                cursor.deleteRow()
    -
    -
    #This will delete any entries less than a given area value (0m2)
    with arcpy.da.UpdateCursor("RasterToPoly1", "Shape_Area") as cursor2:
        for row in cursor2:
            if row[0] < 0: #Update or delete this to change minimum gap size limit
150         cursor.deleteRow()
151

```

Figure 9: Update cursors used to allow user to place lower and upper limits on gap area

SCRIPT PROGRESSION SECTION 2: STATISTICAL ANALYSIS

The newly developed polygon feature class representing canopy gaps is used alongside the input FRI feature class to perform statistical analysis on gap prevalence and distribution within forest stands. For all analyses, only the portion of a gap that falls within the subject FRI polygon is used. This is accomplished by using ArcPy's *Intersect* method available for polygon geometry objects (Esri 2019d). This creates a series of new geometry objects equal to the shape of the spatial intersection between the overlapping objects. An example of this is shown in Figure 10, where only green polygon areas are analyzed within the selected FRI polygon. A nested Cursor approach (Esri 2019c) is used in order to access information stored within the FRI and gap output feature classes simultaneously, ultimately writing to the FRI feature class attribute table.

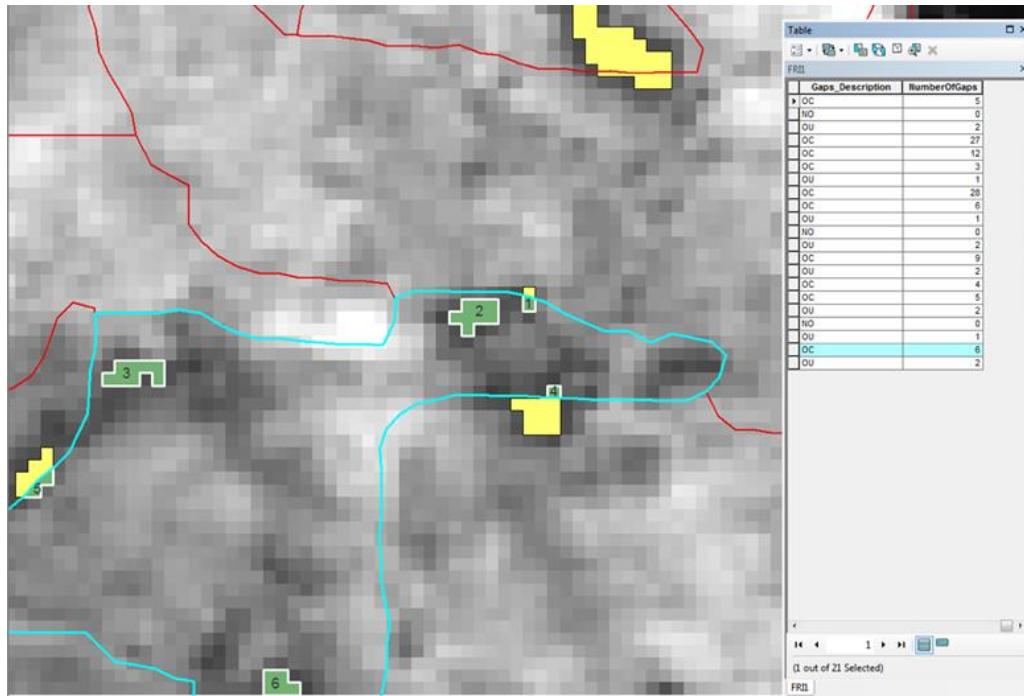


Figure 10: FRI stand with 6 gap portions

Assessing gap prevalence

The prevalence of canopy gaps within a forest stand is assessed in two ways – through the number of gap segments, as well as their cumulative area. A *Search Cursor* is applied to each polygon within the FRI feature class, and the number of gap polygon segments that fall within this polygon is counted using ArcPy’s *GetCount_management* tool (Esri 2019e). In accordance with the Ontario FRI Technical Specifications document, the number of canopy gaps may also be assigned a qualitative description. As per the “HORIZ” field within Ontario’s FRI data, describing horizontal forest structure, stands with 1 or 2 canopy gaps are described as “OU” for “openings uncommon”, whereas stands with more than 2 canopy gaps are described as “OC” for “openings common” (OMNR 2009).

The area encompassed by canopy gaps within a forest stand may also be of interest to forest managers. A new *Search Cursor* is employed to gather the areas of each individual gap portion within a stand and sum them. Dividing this value by the total stand area provides a value of canopy gap coverage (%) for the entire stand.

Assessing gap dispersion

The assessment of canopy gap dispersion within a stand requires the implementation of a more complex statistical analysis. The methodology used in ArcMap's *Average Nearest Neighbour* tool was adapted for use in this script. This tool serves to express the degree of clustering being exhibited by a series of features within a given area (Esri 2019b). The equations presented in Figures 11-14 are used in order to determine the certainty with which the dispersion of canopy gaps within a given area may be deemed "clustered" "random" or "dispersed". Three variables are used within the calculations: " d_i " is the distance between feature i and its nearest neighbour feature, " n " is the number of gap features within the subject area, and A is the area representing the extent of gap occurrences. " D_o " is the observed mean distance between each feature and its nearest neighbour, " D_e " is the expected mean distance between features in a random pattern, " SE " is the standard error based on the variables being used. Finally, a Z-Score is calculated for each stand, from which a p value is derived (Esri 2019f). ArcMap's ANN tool uses an " A " value equal to the area of the smallest rectangle that encompasses all point (gap) features being assessed. As such, the user may opt to use a rectangle of this size, or the area of the stand polygon in the script's analysis of canopy gap clustering.

$$\bar{D}_O = \frac{\sum_{i=1}^n d_i}{n}$$

Figure 11: Formula used to calculate “Do” for each forest stand

Source: (Esri 2019f)

$$\bar{D}_E = \frac{0.5}{\sqrt{n/A}}$$

Figure 12: Formula used to calculate “De” for each stand

Source: (Esri 2019f)

$$SE = \frac{0.26136}{\sqrt{n^2/A}}$$

Figure 13: Formula used to calculate “SE” for each stand

Source: (Esri 2019f)

$$z = \frac{\bar{D}_O - \bar{D}_E}{SE}$$

Figure 14: Formula used to calculate the Z-Score for each stand

Source: (Esri 2019f)

The implementation of these equations in the script are shown in Figure 15, where variable *MinDistList* is a list of all d_i values for features within the subject stand. These calculations are only run should more than 1 gap segment be present within a stand.

The Z-Score returned is representative of the number of standard deviations between the sample mean (average minimum distance between each gap polygon and its nearest neighbour) and the population mean (average minimum distance between each gap polygon and its nearest neighbor, had they been distributed randomly (Esri 2019f)). As the number of standard deviations between the population mean and sample mean increases, it becomes less likely that the pattern seen in the sample mean was a result of

random distribution, thus increasing the confidence with which one may reject the null hypothesis. A p value may be derived from a Z-Score, conferring the degree of confidence with which the null hypothesis may be rejected, if at all.

```

    #If more than 1 gap is present in a stand, feature-clustering calculations are performed
    if len(MinDistList) > 1:
        print("stand min dist", SRow[4], sum(MinDistList),len(MinDistList))

        Do = (sum(MinDistList)/float(len(MinDistList)))
        De = ((0.50)/(math.sqrt(len(MinDistList)/SRow[5])))
        SE = (0.26136)/(math.sqrt(float(len(MinDistList)**2/SRow[5])))
        ZScore = ((Do-De)/SE)
        pval = scipy.stats.norm.sf(abs(ZScore))*2 #calculates the p val for a two-sided test

```

Figure 15: ANN Equations represented in Python script.

SCRIPT PROGRESSION SECTION 3: INCORPORATION WITH FRI FEATURE CLASS

The final section of the script serves to present relevant information in a user friendly fashion – through the attribute table of the input FRI feature class. As mentioned, a nested cursor approach is used to accomplish this. Several Search Cursors are used within an overarching Update Cursor, to allow fields in the FRI attribute table to be updated based on analyses described above, performed using returns from the search cursors. To begin, four new fields are added to the FRI feature class, after ensuring they have not already been created. This is done using an *if* statement to check for the existence of the field, followed by field creation using the *arcpy.AddField_management* tool (Esri 2019a). The fields created, along with their data types are displayed in Table 2.

Table 2: Name and data type for each new field created in FRI feature class.

Field Name	Data Type
Gap_Clustering	String
GapCoverage	Float
Gaps_Description	String
NumberOfGaps	Short

NumberOfGaps and Gaps_Description

These fields provide insight into gap prevalence within a subject forest stand. First, the “NumberOfGaps” field simply provides the user with the quantity of gap portions within the stand. Referring back to Figure 10, a value of 6 would be assigned to the selected polygon in this field, as 6 gap segments are present. The “Gaps_Description” field offers a qualitative description of the number of canopy gaps in accordance with the approach used in the “HORIZ” field of Ontario’s FRI data. A value of “OC” is used when 3 or more gaps are present, “OU” when 1 or 2 gaps are present, and “NO” when no gaps are present.

GapCoverage

This field simply offers a value (%) of the total area covered by canopy gaps with respect to the area of the entire stand. A return of “50” in this field indicates that exactly half of the stand has been detected as being a gap in the stand’s forest canopy.

Gap_Clustering

This field is populated based off of the implementation of the ANN tool described above. Using a confidence level of 95%, this field offers a description of the degree of clustering being exhibited by the gap segments. This is done through the Z-

Score value that is returned. A Z-Score below -1.96 indicates clustering, while one above 1.96 shows dispersion. Any values between these indicates a random dispersal pattern, all with 95% confidence (Esri 2019f). With this in mind, qualitative descriptions of dispersion are applied to each FRI polygon. A value of “c” means the gaps are clustered, “d” means they are dispersed, and “r” means they are randomly distributed. Stands with 1 or 0 canopy gap segments are given a “<Null>” value, as clustering may not be assessed.

RESULTS

VISUAL RESULTS

The script was run on data corresponding to two adjacent forested areas in order to assess the effectiveness of the program at identifying canopy gaps. Figures 16 – 19 offer three-dimensional visualizations of the subject areas, with Figures 16 and 17 showing CHM1 with and without canopy gaps included, and Figures 18 and 19 doing the same for the forest area directly west of CHM1, named CHM2.

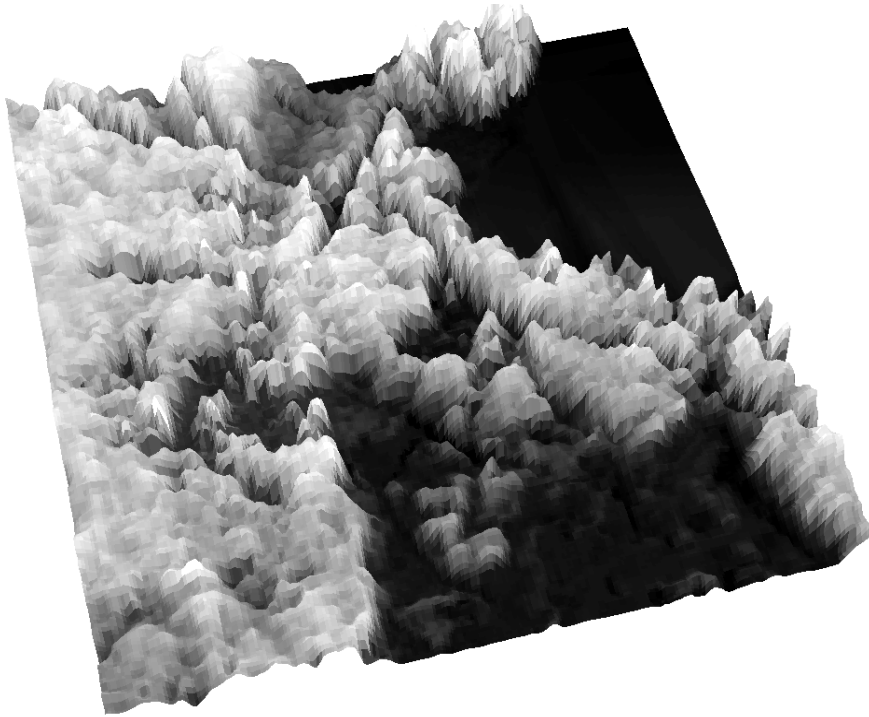


Figure 16: 3d Visualization of CHM1 raster, without canopy gap identification

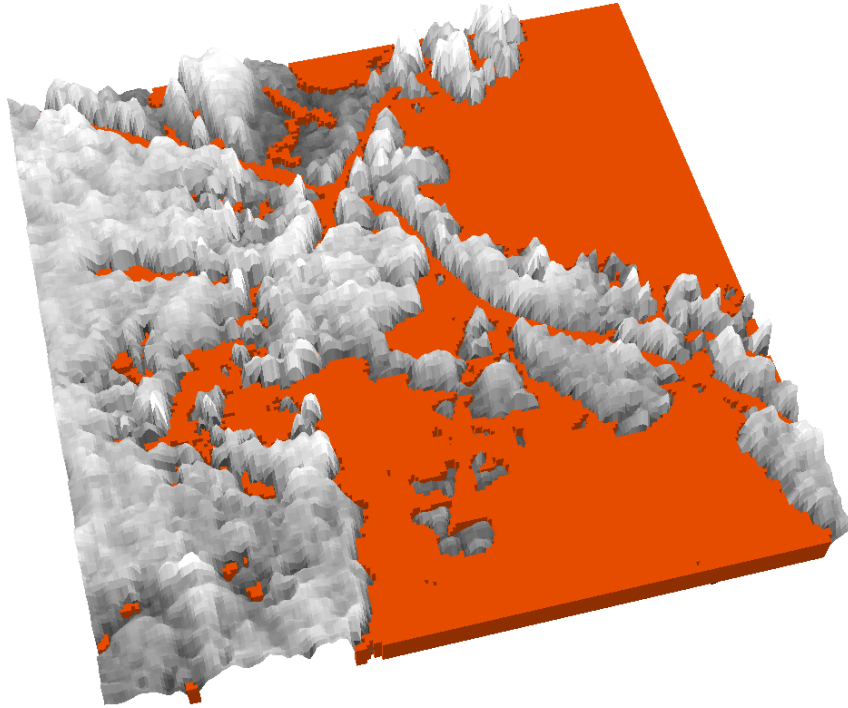


Figure 17: 3d Visualization of CHM1 raster, with canopy gap identification

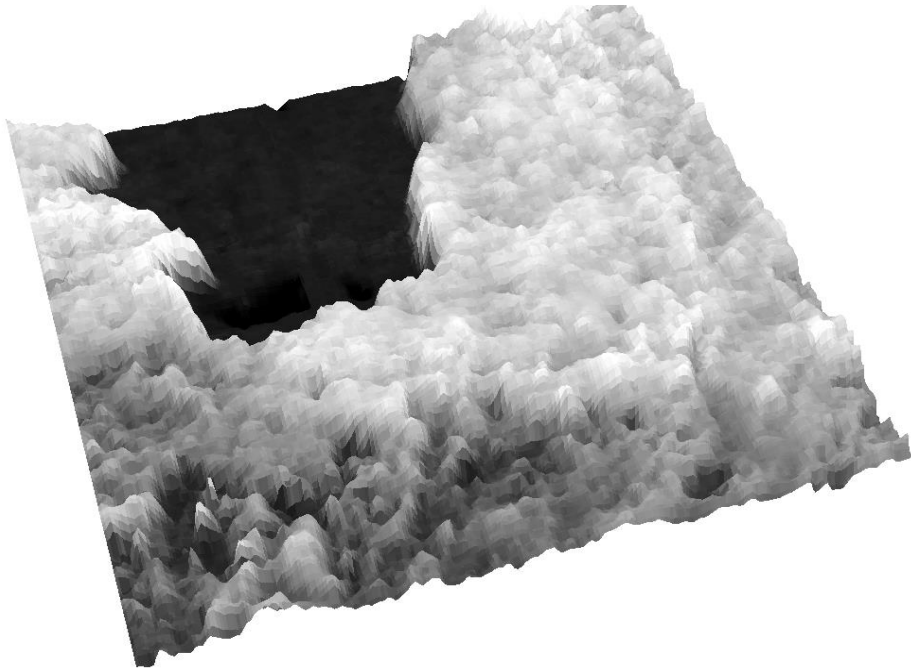


Figure 18: 3d Visualization of CHM2 raster, without canopy gap identification

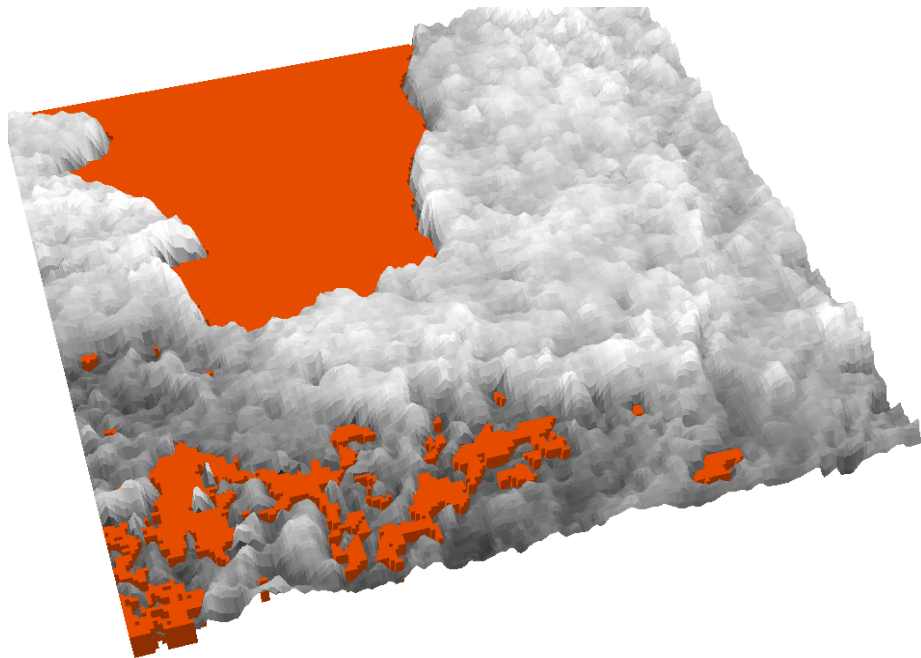


Figure 19: 3d Visualization of CHM2 raster, with canopy gap identification

A visual assessment of these results concludes that the script is identifying canopy gaps within the forest canopy, based on the input definition requirements for a canopy gap. The complete identification of lakes and roads, and interspersed patterns amongst tall forested areas are promising results.

TESTING WITH RANDOM NUMBERS

As mentioned, the efficacy of the script's user defined functions for gap detection (*seedcell* and *region_growing*) was tested using rasters composed of random numbers within defined ranges. The results presented here exemplify the functionality of the script's gap detection capabilities.

First, a back ground range of 5-25 was used in combination with a test range of 0-6. Test values between 0-3 will be detected by default, while values between 3-5 depend on surrounding "canopy" heights, and values between 5-6 will not be detected as gaps. As only background values between 5 and 10 may not qualify a test value between 3 and 6 as a gap, it was expected that the majority of cells within the test range would be detected as gaps, with the exception of test values between 5 and 6. As per Figure 20, one can see that the majority of the test areas are marked as canopy gaps (coloured polygons). In fact, the region growing algorithm extended the detected gap to every cell between 3-5, with only test values between 5-6 being left out. This makes sense, as only a single border cell with a value greater than 10 would trigger the algorithm to grow to every neighbouring cell with a value between 3 and 5. Interestingly, the top left test square was marked as having two distinct gaps, due to a sequence of test values over 5 acting as a barrier for the region growing function. Within the larger test rectangle, there

are two single cell gaps being displayed, due to them being surrounded by cells with values between 5 and 6, not considered as gaps.

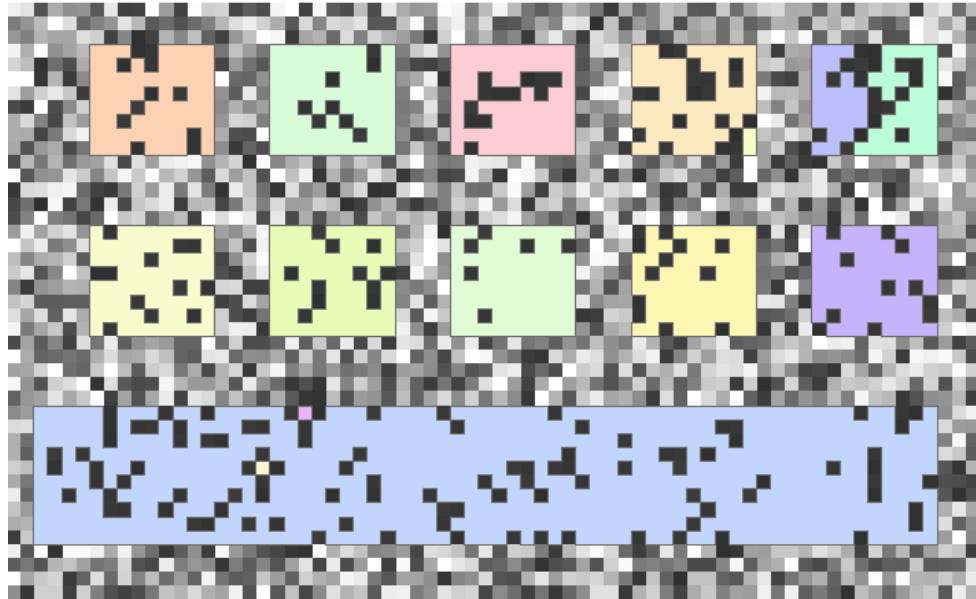


Figure 20: “Gaps” detected by running script on raster of randomly generated test values. Test values ranging from 0-6, background values ranging from 5-25.

A second test was run with test values ranging from 4-6 and background values ranging from 5-10. This should decrease the number of cells being detected as canopy gaps. As there are no values between 0-3, automatically detected as gaps, the region growing algorithm will extend shorter distances, resulting in more numerous, smaller gap patterns than those seen in the previous test. Furthermore, approximately one half of the test values should be precluded from gap detection altogether as they will fall between 5 and 6. Looking at Figure 21, one can see this pattern play out, first with far fewer test cells being detected as gaps, and many unique gap polygons being created, as opposed to few large ones.

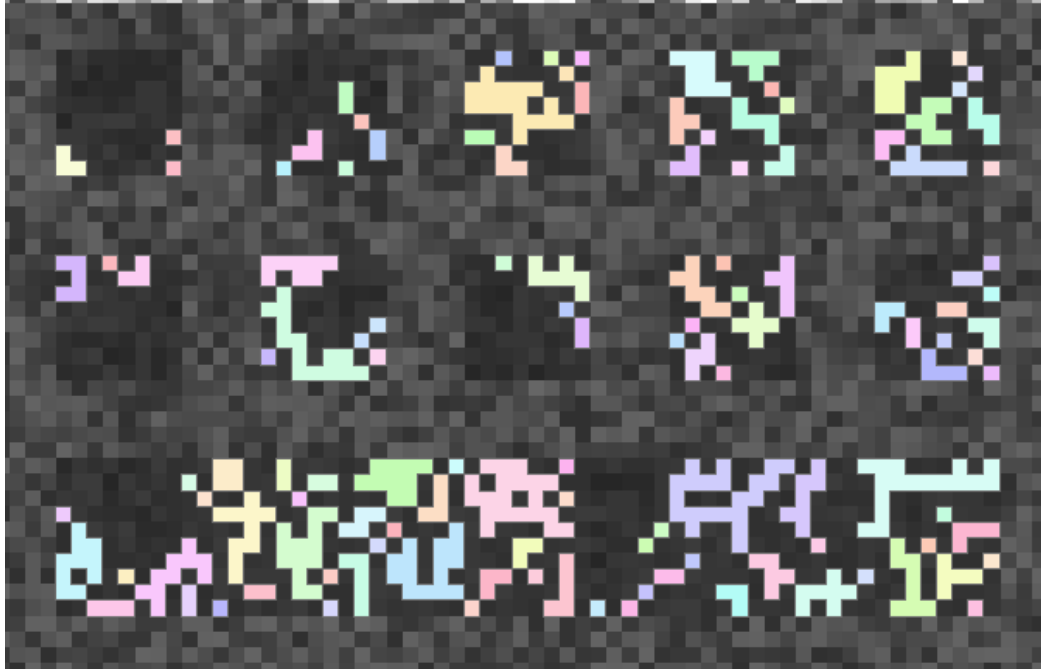


Figure 21: “Gaps” detected by running script on raster of randomly generated test values. Test values ranging from 4-6, background values ranging from 5-10.

ASSESSMENT OF GAP PREVALENCE RESULTS

The presence of canopy gaps within forest stands is measured through the outputs to 3 fields; “GapCoverage”, “NumberOfGaps”, and “Gaps_Description”.

The information returned in the “GapCoverage” did not contain any errors. As the field is populated simply by dividing the total area of all gaps by the stand area, it is unlikely to encounter error. The example in Figure 2 shows the “GapCoverage” field being properly populated. With the red numbers representing the area of gap segments within the selected FRI polygon (totaling 629m² when rounded), dividing this by the stand area of 24306m², one is left with a resulting gap coverage of approximately 2.59%.

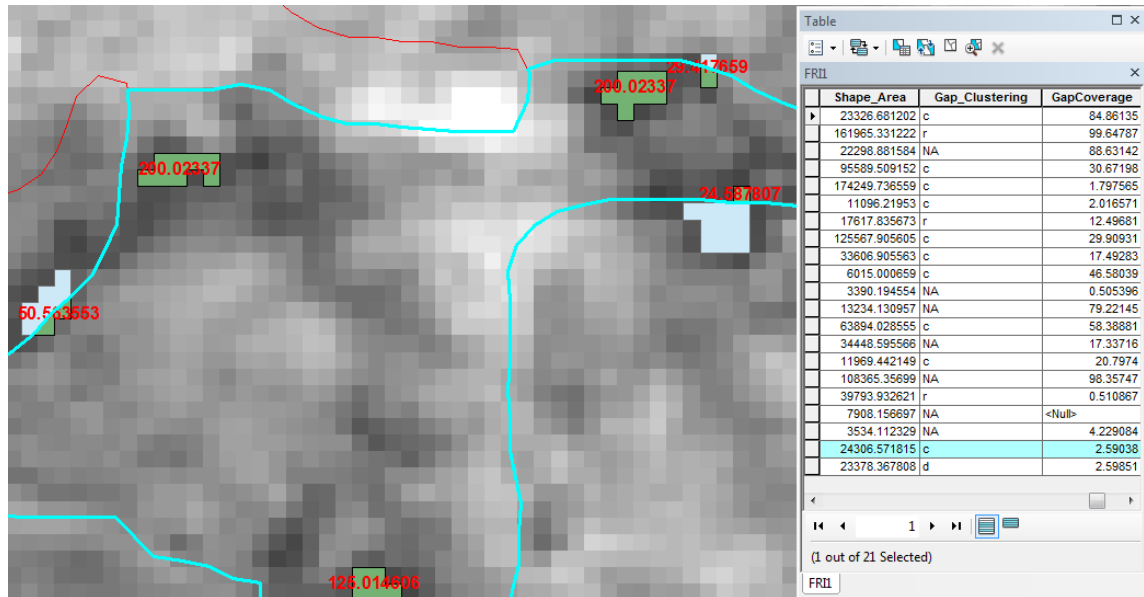


Figure 22: Proof of “GapCoverage” field being populated correctly.

As the “NumberOfGaps” and “Gaps_Description” field provide the user with related information, their results may be assessed together. First, the “NumberOfGaps” field is functioning correctly, in that it is properly identifying every gap segment within a given FRI polygon as a unique gap feature. Proof is offered in Figure 23, where a total of 4 gap segments are illustrated within an FRI polygon, which then has its “NumberOfGaps” field populated accordingly. In this same Figure, one may also see that the “Gaps_Description” field is being filled appropriately. FRI Polygons with 0 gaps are given a “Gaps_Description” value of “NO”, while those with 1 or 2 gaps are labelled “OU”, and polygons with more than 2 gaps are given “OC”. The accuracy of the “Gaps_Description” entries with respect to the information provided on gap prevalence in the FRI feature classes upon acquisition will be discussed later.

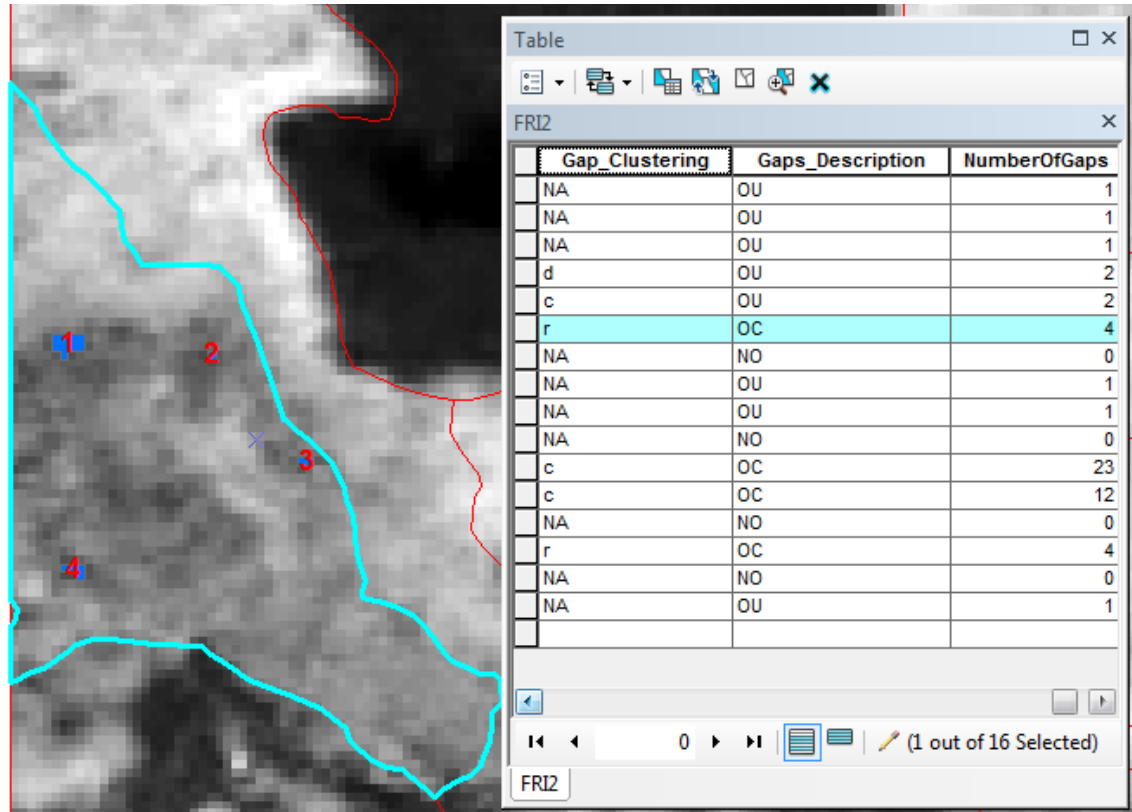


Figure 23 “NumberOfGaps” and “Gaps_Description” fields functioning properly.

ASSESSMENT OF GAP CLUSTERING RESULTS

Due to the complex nature of the calculations used to calculate canopy gap clustering, multiplied by the complex nature of the forest vertical and horizontal structure, providing absolute proof of the script’s efficacy was not practically feasible. However, some examples may be provided to exhibit the results acquired in different situations. First, Figure 24 provides an example of a random gap distribution. This is sensible as there is no clear pattern to the gap location, as they occur sporadically throughout the stand area. It is likely the two small segments towards the center of this figure that preclude these gaps from being classified as dispersed due to their closer proximity.

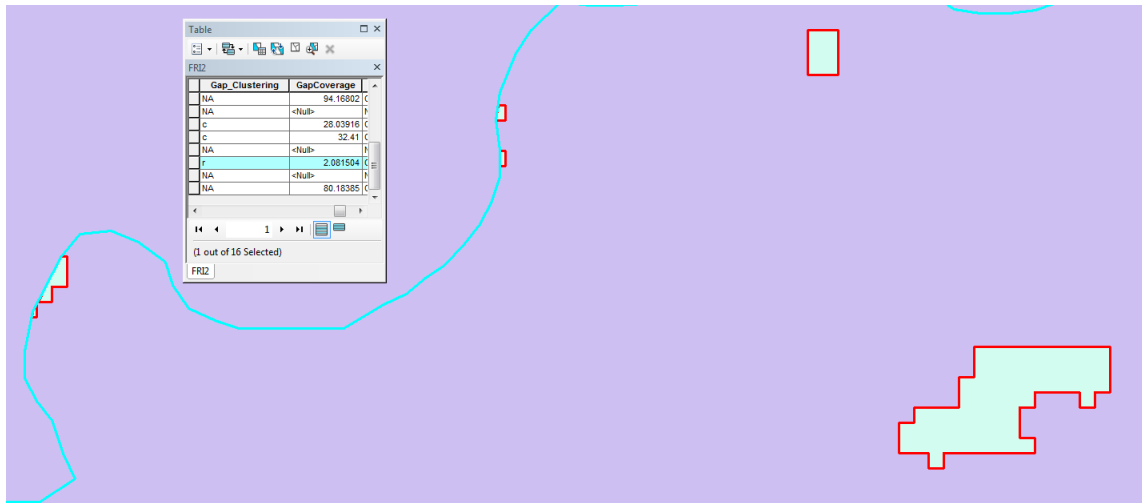


Figure 24: Example of a “random” gap distribution.

Looking to Figures 25 and 26, one can see examples of gap polygon distributions leading to designations as clustered and dispersed, respectively. In Figure 25, three green ellipses enclose areas in which gaps appear to be occurring in close proximity, thus leading the ANN algorithm to detect them as being clustered, with 95% confidence. Alternatively, Figure 26 shows a stand in which gaps appear along the stand’s edges, quite distant from each other. This lead to the stand’s designation as dispersed.

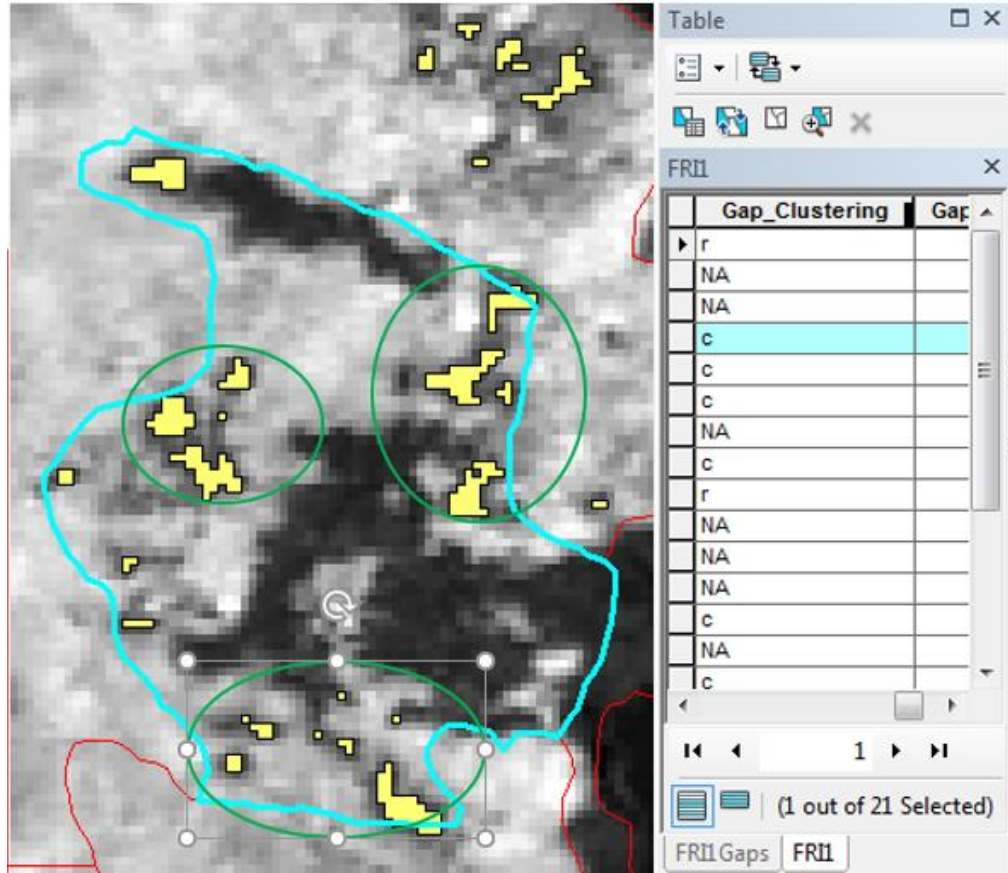


Figure 25: Example of gap distribution classified as “clustered”

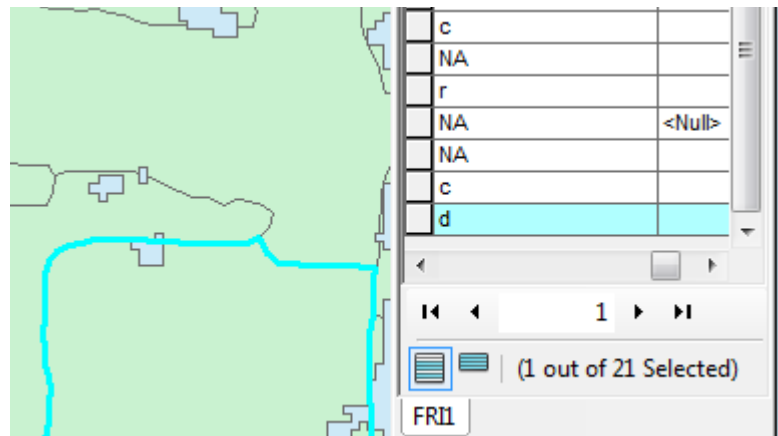


Figure 26: Example of gap distribution classified as “dispersed”

DISCUSSION

FUNCTIONALITY

The presented script offers a tool for the delineation of forest canopy gaps using topological analysis. As per the three-dimensional visualizations offered in Figures 17 and 19, gap detection does appear to be effective. The script may prove useful to resource managers or researchers under a variety of circumstances. The application of a script such as this is a stark improvement over traditional manual methods of gap delineation in terms of temporal and monetary costs. Should a contemporary dataset be available for an area of interest, a user may apply this tool to be given an understanding of where canopy gaps are occurring within the subject area, as well as their prevalence and dispersion, prior to beginning field work. Certainly this would streamline operations concerned with said gaps. An additional benefit of an open-source script such as this are the opportunities for customization. While the majority of the script acts as a framework for automated information processing, the most “important” aspect – canopy gap detection – can be manipulated and perhaps fine-tuned to meet specific needs rather simply. While specific examples of this will be discussed below, this customizability may serve to broaden the applicability of this script as a wider range of specific uses become available through small modifications to the program.

IMPROVING FRI DATA RICHNESS

Four fields are added to the information available by default within Ontario’s FRI spatial data. Due to the broad number of applications that this data serves, the

ability to add relevant information to what is already available will certainly prove useful to some, at least in niche cases if not more universally.

“NumberOfGaps” and “Gaps_Description”

The only information regarding horizontal forest structure with respect to canopy gaps within the FRI data is the “HORIZ” field (OMNR 2009). The OMNR offers six possible values to describe this horizontal structure, however only two of these six refer to canopy gaps, with four describing species distributions across the stand. Additionally, as only one of these six values may be attributed to each stand, one of the four codes describing species distribution appear to be used far more commonly than those describing gap frequency. Within the two data sets used in testing this script, species distribution codes were used 25 times, compared to just 4 cases in which gap frequency was described. As such, offering these fields is sure to improve the consistency of available data. Although only the “NumberOfGaps” field is really required for this practically, the addition of the “Gaps_Description” field was made for the sake of uniformity with the existing data.

“GapCoverage”

This field provides an idea as to the pervasiveness of canopy gaps within forest stands. Managers may be interested in information such as this when prioritizing harvest scheduling. As heavy canopy gap presence is indicative of declining forest health, harvesting in blocks with high gap coverage may want to be prioritized. While perhaps somewhat limited due to the coarseness of the data used in testing, the results given by running this script are adequate. Should more precise data be available to the user, results will only improve.

“Gap Clustering”

The addition of this field adds information that is completely novel to the FRI dataset. A user may find this information useful should their intent be determining a reason for canopy gaps within a stand. For example, a stand where the gaps are heavily clustered may imply that a discrete disturbance event or site condition is responsible for the openings, whereas a random gap dispersion may be indicative of more sporadic disturbances like wind throw, or death due to old age. The manual adaptation and implementation of the *Average Nearest Neighbour* tool offered within ArcMap (Esri 2019b) was imperfect due to the nature of the tool’s intended use. These imperfections will be discussed later. However, upon visual assessment the results attained do appear to be accurate enough to be useful. Ultimately, these results must be interpreted with a critical eye, as there is no real method for determining error beyond subjective inference. Ultimately, while the function is not optimized within the context of the data used in this script, the addition of data where none previously existed should be seen as a benefit.

OPPORTUNITIES FOR CUSTOMIZATION

Vegetation height values associated with gaps

With this script relying on topological analysis to identify canopy gaps, establishing a critical regeneration height is crucial in the application of the script. This script utilizes an absolute and relative critical regeneration height. For use in Ontario, an absolute height value of 3 m was used as a default due to this being a requirement in declaring a forest stand to be “forested”, while a critical threshold was added for the

sake of adding complexity to its potential for customization. Values chosen heavily influence the ability of the scripts results to be repeated or compared with alternative results (Barden 1989). However, the ability to run a script such as this multiple times with different values in a short period of time is a testament to its efficacy as a tool. While a 3 m vegetation height was used in the scripts development, different users may find different values to be more applicable. With elk showing a heavy preference for browse vegetation between 1 m and 2 m in height (Rounds 1979), one concerned with locating areas suitable for elk feeding within a forest may opt to restrict gap detection to gaps within this height range.

Altering the critical regeneration height may also have a use in determining the cause of a canopy gap. As per Schneider and Larson (2017), a differentiation between edaphic and developmental canopy gaps may be useful to some. With edaphic gaps being restricted from regeneration due to ground conditions such as exposed bedrock, and developmental canopy gaps actively regenerating, it is possible that height values stored within the CHM could be used to differentiate between these gap types. For example, a threshold height of 0.5 m may be used. Below this height an assumption may be made concluding no regeneration is occurring. This height could be customized by the user, but should offer a buffer above absolute 0 to account for errors in the CHM data.

Critical regeneration thresholds may be altered by the user within the *seedcell* and *region_growing* functions presented in Figures 3 and 4, respectively. Within Figure 3, line 70 may be updated to alter critical regeneration height requirements. The

region_growing function shown in Figure 4 must be updated in parity with the *seedcell* function. This is done by updating code lines 104-107 as shown in Figure 4.

Limitation of gap area values

To include or remove identified gaps based on their area is a simple task, requiring a modification to just a single line within the script. A tentative maximum area limit of 1000 m² was used in the development of the script as environmental conditions within gaps of this size have been found to vary significantly from those found in gaps of smaller sizes (Schliemann and Bockheim 2011). This is expressed in terms of increased light penetration and soil temperatures (Zhang and Zak 1995; Muscolo et al. 2014). Thus, studies looking into topics such as gap dynamics (Spies et al. 1990; Stewart et al. 1991; Salvador-Van Eysenrode et al. 1998) may opt to limit gap size based on these physical characteristics. Of course, this upper limit is specific to the study subject and can be altered accordingly. Alternatively, removing the upper canopy area limit may be advantageous in certain situations as well. For example, a study looking at overall gap coverage within a German study area (Kathke and Bruelheide 2010) could have applied this technique to determine the amount of unforested area in their study area.

Due to the limited availability of suitable data, a raster with 5 m resolution was used in developing this script. As such, no lower area limit was employed, as the smallest detectable gap was 25 m². However, should finer resolution data be available, one could remove gap polygons smaller than a given area very easily as well. This may allow a user to ignore results that are not useful in their specific case. For example, gaps between 20 m² and 80 m² in interior douglas-fir forests were too small to initiate gap-

phase regeneration, the preliminary stage of canopy gap-induced forest succession (Zustovic 2015). For use in this region, one may opt to preclude gaps of this size in their results. Should the user wish to compare the results of running this script with the information available in an FRI data set, a minimum area of 16 m² may be used, as this is a limitation imposed by the OMNRF (OMNR 2009). Ultimately, placing limitations upon canopy gap areas being identified and analyzed by the script is certain to alter results. It should be determined by the user the optimal application of the script for their given situation and goal.

User customization of the limitation on gap sizes that are to be identified is accomplished by updating two lines of code. Looking back to Figure 9, line 143 may be updated to impose a maximum area limit, while line 149 enforces a minimum gap area.

OPPORTUNITIES FOR IMPROVEMENT

User Interface

In its present state, the user is only prompted for three inputs: the directory (geodatabase) in which their data is located, and the names of their respective CHM and FRI files entered as raw text inputs. While navigation to the geodatabase through the user's file tree is a sensible method for directory selection, selection of data could perhaps be improved through a drop-down menu. Furthermore, the opportunities for customization discussed above must be accessed through manual alteration of variables within the script. Alternatively, these variables could be presented to the user following the selection of their data. This could be accomplished in a single window with five spaces for inputs: absolute vegetation height limit, relative vegetation height limit, a

maximum and minimum gap area, and a drop-down to select stand area or area extent to be used in the clustering calculations, discussed later. While a clean user interface could improve usability, it was not a priority within this project.

Creation of multiple layers

Throughout the script four raster files and one polygon feature class are created, with all four rasters being deleted at the end. As the creation and deletion of files are computationally expensive, this inevitably slows down the script. While the output feature class is required, there is potential that NumPy arrays could have been used in some cases to store the information being stored in the raster files. Particularly when working with larger datasets, increasing the relative speed at which a program runs is a benefit.

Bare ground versus regenerating gaps

As presented previously as a use for having a customizable minimum regeneration threshold, the script could be modified to give an idea regarding the type of canopy gap being identified. By assuming that a canopy gap under a critical height is being prevented from re-establishment, the program could be altered to list these gaps as edaphic gaps, as opposed to a regenerating developmental gap. This information could be carried into the output gap polygon feature class in a new field. While the accuracy of this sort of metric would require field work to be assessed, it may be of use to some as a preliminary observation.

Improvements to clustering algorithm

The imperfect implementation of ArcMap's *Average Nearest Neighbour* algorithm stems from two major points: the use of polygons instead of points as subjects for analysis, and the irregular shape of forest stands.

Using polygons as the focus for analysis leads to two issues within the clustering calculations. First, identified gaps that are connected by a corner as opposed to an edge are considered to be independent gaps in the canopy. Figure 27 shows an example of this, where a total of 8 gaps are identified in close proximity. One could argue that Gap 3 and Gap 6 should be included in Gap 7 and Gap 5, respectively. Ultimately, this sways the results of the *Average Nearest Neighbour* algorithm towards clustering, as the minimum distances between all four of these polygons is now 0m, as their nearest neighbours are spatially connected. Ultimately, it would be a subjective decision by the user that should determine whether or not gaps in these situations should be considered as one, or two entities. This issue may be avoided by using higher resolution data, offering a more natural depiction of the gap shape and connectedness.

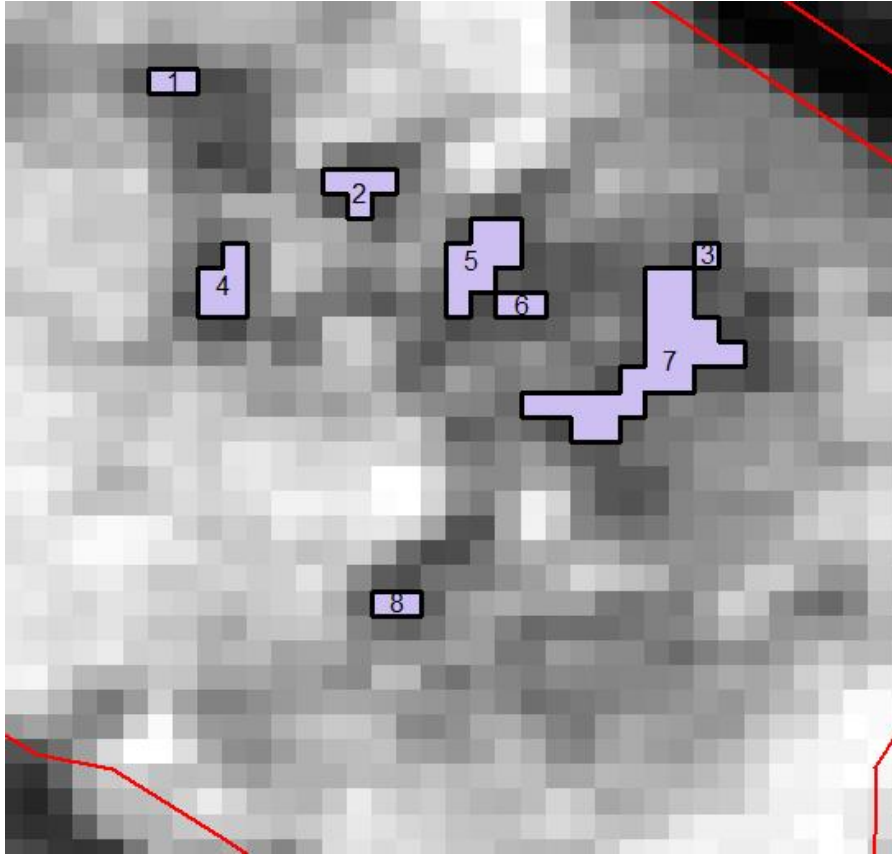


Figure 27: Example of connected gaps being identified as individual gaps.

A second issue arising from using polygons instead of points to run this tool is that it is the shortest distances between two polygons being used in the calculations. While this is the literal interpretation of being a “nearest neighbour” it is not so simple to conclude this is the correct approach in this situation. A hypothetical situation is presented in Figure 28 illustrating the issue that may stem from this. As one can see, the hypothetical gaps are being measured as being 11.2 m apart, while the areas subject to the implications of existing within a canopy gap are closer to 250 m apart. No practical solution to this problem was found.

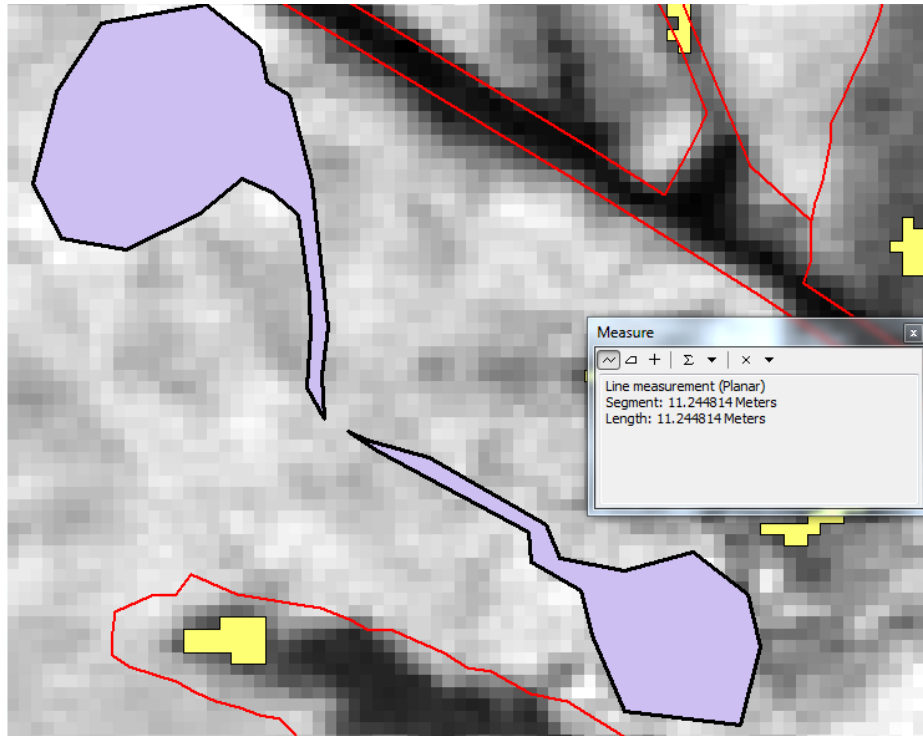


Figure 28: Hypothetical gap shapes leading to errors in clustering calculations

The second factor that leads to error in the *Average Nearest Neighbour* calculations is the irregular shape of forest stands. The algorithm is intended to use the area of the smallest rectangle that completely encompasses all subject points (Esri 2019f). By default, the program uses the area of the FRI polygon in which the subject gaps fall. This could lead to either underestimation or overestimation of the degree of clustering being exhibited, depending on the shape of the stand. Alternatively, the option to use a rectangle encompassing the extent of the gaps is provided. However, this too has associated issues in that an assumption is made that all points (polygons in this case) have an equally likely chance of appearing anywhere within the area (Esri 2019f). In this case this won't work as any areas that are within a different FRI polygon have no chance of containing a gap that is attributed to the subject FRI polygon. Figure 29

illustrates this issue. Should one choose to use the rectangular extent of all gaps as the subject area (shown in purple), it will result in overestimation of area and subsequent clustering as only the area highlighted in cyan has potential to contain the gaps found within this stand. Again, it should be said that the conclusions made regarding gap clustering should be assessed critically, with reference to the output feature class.

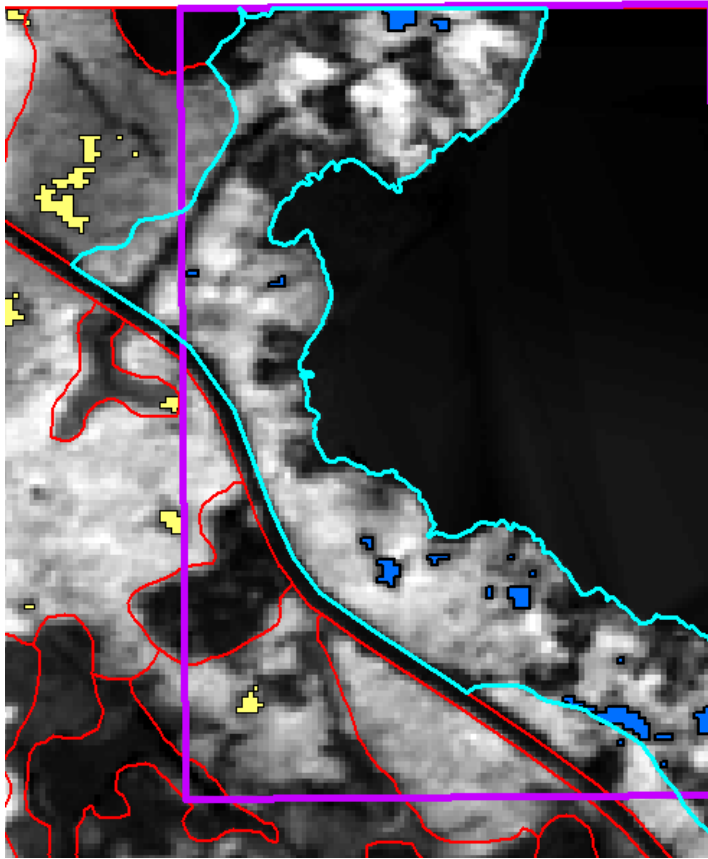


Figure 29: Illustration of errors associated with the areas used in clustering calculations.

CONCLUSIONS

Overall, the script functions as intended. Canopy gaps are able to be adequately identified subject to user inputs, and the statistical analyses function properly. It is my hope that this program will find use amongst researchers or resource managers

concerned with the presence and distribution of canopy gaps across their work areas.

Ultimately, this project served its purpose in teaching me a great deal about the Python programming language, bringing me closer to understanding the breadth of applications script development can have both within, and outside of careers in the natural resources management field.

LITERATURE CITED

- Bagaram, M., D. Giuliarelli., G. Chirici, F. Giannetti., A. Barbati. 2018. UAV remote sensing for biodiversity monitoring: are forest canopy gaps good covariates? *Remote Sensing*. 2018(10):1397-1425.
- Bartemucci, P., K. Coates, K. Harper, E. Wright. 2002. Gap disturbances in northern old-growth forests of British Columbia, Canada. *Journal of Vegetation Science*. 13(5):685-696.
- Bergeron, Y. 2000. Species and stand dynamics in the mixed woods of Quebec's southern boreal forest. *Ecology*. 81:1500-1516.
- Betts, H., L. Brown, G. Stewart. 2005. Forest canopy gap detection and characterization by the use of high-resolution Digital Elevation Models. *New Zealand Journal of Ecology*. 29(1):95-103.
- Bonnet, S., R. Gaulton, F. Lehaire, P. Lejeune. 2015. Canopy gap mapping from airborne laser scanning: an assessment of the positional and geometrical accuracy. *Remote Sens*. 7(9):11267-11294.
- Bressert, E. 2013. *SciPy and NumPy*. O'Reilly Media Inc. Sebastopol, CA. 67pp.
- Brokaw, N. 1982. The definition of treefall gap and its effect on measure of forest dynamics. *Biotropica*. 14(2):158-160.
- Canham, C. 1988. An index for understory light levels in and around canopy gaps. *Ecology*. 69(5):1634-1638.
- Churchill, D., G. Carnwath, A. Larson, S. Jeronimo. 2017. Historical forest structure, composition, and spatial pattern in dry conifer forests of the western Blue Mountains, Oregon. USDA Forest Service. Technical Report. 91 pp.
- Desclée, B., P. Bogaert, P. Defourny. 2006. Forest change detection by statistical object-based method. *Remote Sensing of Environment*. 102(2006):1-11.
- Esri. 2019a. Add Field. <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/add-field.htm>. Viewed: April 1, 2019.
- Esri. 2019b. Average Nearest Neighbor. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-statistics-toolbox/average-nearest-neighbor.htm>. Viewed: April 1, 2019.
- Esri. 2019c. Cursor. <http://desktop.arcgis.com/en/arcmap/10.3/analyze/arcpy-classes/cursor.htm>. Viewed: April 1, 2019.
- Esri. 2019d. Geometry. <http://desktop.arcgis.com/en/arcmap/10.3/analyze/arcpy-classes/geometry.htm>. Viewed: April 1, 2019.
- Esri. 2019e. Get Count. <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/get-count.htm>. Viewed: April 1, 2019.
- Esri. 2019f. How average nearest neighbor works. <https://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/h-how-average-nearest-neighbor-distance-spatial-st.htm>. Viewed: March 15, 2019.

- Esri. 2019g. NumPyArrayToRaster. <https://pro.arcgis.com/en/pro-app/arcpy/functions/numpyarraytoraster-function.htm>. Viewed: April 1, 2019.
- Esri. 2019h. Raster Calculator. <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/raster-calculator.htm>. Viewed: April 1, 2019.
- Esri. 2019i. RasterToNumPyArray. <https://pro.arcgis.com/en/pro-app/arcpy/functions/rastertonumpyarray-function.htm>. Viewed: April 1, 2019.
- Esri. 2019j. Raster To Polygon. <https://pro.arcgis.com/en/pro-app/tool-reference/conversion/raster-to-polygon.htm>. Viewed: April 1, 2019.
- Esri. 2019k. Rename. <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/rename.htm>. Viewed: April 1, 2019.
- Esri. 2019l. UpdateCursor. <https://pro.arcgis.com/en/pro-app/arcpy/functions/updatecursor.htm>. Viewed: April 1, 2019.
- Esri. 2019m. What is z-score? What is p-value? <https://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/what-is-a-z-score-what-is-a-p-value.htm>. Viewed: March 15, 2019.
- Foster, J., W. Reiners. 1986. Size distribution and expansion of canopy gaps in a northern Appalachian spruce-fir forest. *Vegetation*. 68(2):109-114.
- Gaulton, R., T. Malthus. 2010. LiDAR mapping of canopy gaps in continuous cover forests: a comparison of canopy height model and point cloud based techniques. *International Journal of Remote Sensing*. 31(5):1193-1211.
- Getzin, S. K. Wiegand, I. Schoning. 2012. Assessing biodiversity in forests using very high-resolution images and unmanned aerial vehicles. *Methods Ecol. Evol.* 3(2011):397-404.
- [GOO] Government of Ontario. 2010. Discovering Ontario Data. <https://www.javacoeapp.lrc.gov.on.ca/geonetwork/srv/en/main.home?uuid=2ecc3b65-34e5-4404-a718-b9b8e048a9d8>. Viewed: April 1, 2019.
- Gray. A., T. Spies, M. Easter. 2002. Microclimatic and soil moisture responses to gap formation in coastal Douglas-fir forests. *Can. J. For. Res.* 32(2002):332-343.
- Green, P. 1996. Canopy gaps in rain forest on Christmas Island, Indian Ocean: Size distribution and methods of measurement. *Journal of Tropical Ecology*. 12(3):426-434.
- Hu, L., J. Zhu. 2009. Determination of the tridimensional shape of canopy gaps using two hemispherical photographs. *Agricultural and Forest Meteorology*. 149(5):862-872.
- Hunter, M., M. Keller, D. Morton, B. Cook, M. Lefsky, M. Ducey, S. Saleska, R. Cosme de Oliveira, J. Schietti. 2015. Structural dynamics of tropical moist forest gaps. *PLOS One*. 10(8):1-19.
- Kathke, S., H. Bruelheide. 2010. Gap dynamics in a near-natural spruce forest at Mt. Brocken, Germany. *Forest Ecology and Management*. 259(3):624-632.

- Koukoulas, S., G. Blackburn. 2004. Spatial relationships between tree species and gap characteristics in broad-leaved deciduous woodland. *Journal of vegetation science*. 16(5):587-596.
- Kuhlman, D. 2012. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*.
- Kuttner, B. 2018. Ontario forest resource inventory enhancement using ADS40 aerial imagery and LiDar data. Mitacs. <https://www.mitacs.ca/en/projects/ontario-forest-resource-inventory-enhancement-using-ads40-aerial-imagery-and-lidar-data>. Viewed: October 15, 2018.
- Lorimer, C., L. Frelich. 1989. A methodology for estimating canopy disturbance frequency and intensity in dense temperate forests. *Canadian Journal of Forest Research*. 19(5):651-663.
- Malahlela, C., M. Cho, O. Mutanga. 2014. Mapping canopy gaps in an indigenous subtropical coastal forest using high-resolution Worldview-2 data. *International Journal of Remote Sensing*. 25(17):6397-6417.
- Muscolo, A. S. Bagnato, M. Sidari, R. Mercurio. 2014. A review of the roles of forest canopy gaps. *Journal of Forest Research*. 25(4):725-736.
- Nakashizuka, T., T. Katsuki, H. Tanaka. 1995. Forest canopy structure analyzed by using aerial photographs. *Ecological Research*. 10(1):13-18.
- Ni, X., B. Bjorn, W. Yang, H. Li, S. Liao, B. Tan, K. Yue, Z. Xu, L. Z, F. Wu. 2018. Formation of forest gaps accelerates C, N and P release from foliar litter during 4 years of decomposition in an alpine forest. *Biogeochemistry*. 139(3):321-335.
- Numpy. 2018. Quickstart tutorial. <https://docs.scipy.org/doc/numpy/user/quickstart.html>. Viewed: January 27, 2019.
- Oldeman, R. 1978. Architecture and energy exchange of dicotyledonous trees in the forest. In Tomlinson, P., M. Zimmermann. 2010. *Tropical Trees as Living Systems*. Cambridge University Press, New York. 696 pp.
- [OMNR] Ontario Ministry of Natural Resources. 2009. *Forest Resource Inventory Technical Specifications*. Ontario Ministry of Natural Resources. 93 pp.
- [OMNRF] Ontario Ministry of Natural Resources and Forestry. 2018. *Northwestern Ontario Orthophotography Project (NWOOP) 2017 Digital Elevation Model User Guide*. Queen's Printer for Ontario. Toronto, ON. 21 pp.
- Rounds, R. 1979. Height and species as factors determining browsing of shrubs by Wapiti. *Journal of Applied Ecology*. 16(1979):227-241.
- Runkle, J. 1981. Patterns of disturbance in some old-growth forests of Eastern North-America. *Ecology*. 62(4):1041-1051
- Runkle, J. 1982. Patterns of disturbance in some old-growth mesic forests of Eastern North-America. *Ecology*. 63(5). 1533-1546.

- Runkle, J. 1992. Guidelines and sample protocol for sampling forest gaps. General technical report. USDA Forest Service, Pacific Northwest Research Station. Portland, Oregon. 44pp.
- Salvador-Van Eysenrode, D., J. Bogaert, P. Van Hecke, I. Impens. 1998. Influence of tree-fall orientation on canopy gap shape in an Ecuadorian rain forest. *Journal of Tropical Ecology*. 14(6):865-869.
- Schliemann, S., J. Bockheim. 2011. Methods for studying treefall gaps: a review. *For. Ecol. Manage.* 261(7):1143-1151.
- Schneider, E., A. Larson. 2017. Spatial aspects of structural complexity in Sitka spruce – western hemlock forests, including evaluation of a new canopy gap delineation method. *Can. J. For. Res.* 47(2017):1033-1044.
- SciPy Community. 2014. `Scipy.ndimage.filter.generic_filter`. https://docs.scipy.org/doc/scipy0.16.1/reference/generated/scipy.ndimage.filters.generic_filter.html. Viewed: January 27, 2019.
- SciPy Developers. 2019. <https://www.scipy.org/>. Viewed: January 27, 2019.
- Seidel, D., C. Ammer, K. Puettmann. 2015. Describing forest canopy gaps efficiently, accurately, and objectively: New prospects through the use of terrestrial laser scanning. *Agricultural and Forest Meteorology*. 213(2015):23-32.
- Spies, T., J. Franklin, M. Klopsch. 1990. Canopy gaps in Douglas-fir forests of the Cascade Mountains. *Canadian Journal of Forest Research*. 20(5):649-658.
- Stewart, G., A. Rose, T. Veblen. 1991. Forest development in canopy gaps in old-growth beech (*Nothofagus*) forests, New Zealand. *Journal of Vegetation Science*. 2(5):679-690.
- St-Onge, B. 2018. SEGMA – Tree crown delineation from airborne Lidar CHM's. User Guider. University du Quebec a Montreal. 21pp
- St-Onge, B., J. Jumelet, M. Cobello, C. Vega. 2004. Measuring individual tree height using a combination of stereophotogrammetry and LIDAR. *Canadian Journal of Forest Research*. 34(10):2122-2130.
- Torimaru, T., A. Itaya, S. Yamamoto. 2012. Quantification of repeated gap formation events and their spatial patterns in three types of old-growth forests: analysis of long-term canopy dynamics using aerial photographs digital surface models. *Forest Ecology and Management*. 284(2012):1-11.
- Tyrrell, L., T. Crow. 1994. Structural characteristics of old-growth Hemlock-hardwood forests in relation to age. *Ecology*. 75 (2):370-386.
- Vehmas, M., P. Packalen, M. Maltamo, K. Eerikainen. 2011. Using airborne laser scanning data for detecting canopy gaps and their understory type in mature boreal forest. *Ann. For. Sci.* 68(4):825-835.
- Vepakomma, U., B. St-Ong, D. Kneeshaw. 2007. Spatially explicit characterization of boreal forest gap dynamics using multi-temporal lidar data. *Remote Sensing of Environment*. 112(2008):2326-2340.

- Vepakomma, U., B. St-Onge, D. Kneeshaw. 2008. Spatially explicit characterization of boreal forest gap dynamics using multi-temporal LIDAR data. *Remote Sensing of Environment*. 112(5):2326-2340.
- Vepakomma, U., D. Kneeshaw, M. Fortin. 2012. Spatial contiguity and continuity of canopy gaps in mixed wood boreal forests: persistence, expansion, shrinkage and displacement. *J. Ecol.* 100(5):1257-1268.
- Whitmore, T. 1989. Canopy gaps and the two major groups of forest trees. *Ecology*. 70(3):536-538.
- Yamamoto, S. 1992. The gap theory in forest dynamics. *Botanical Magazine – Tokyo*. 105(1992):375-383.
- Zhang, Q., J. Zak. 1995. Effects of gap size on litter decomposition and microbial activity in a subtropical forest. *Ecology*. 76(7):2196-2204.
- Zhu, J., X. Wang, J. Chen, H. Huang, X. Yang. 2014. Estimating fractal dimensions of tree crowns in 3-D space based on structural relationships. *Forestry Chronicle*. 90(2):177-183.
- Zhu, J., G. Zhang, G. Wang, Q. Yan, D. Lu, X. Li, X. Zheng. 2015. On the size of forest gaps: Can their lower and upper limits be objectively defined? *Agric. For. Meteorol.* 213(2015):64-76.
- Zustovic, M. 2015. Canopy gap size affects regeneration potential of interior Douglas-fir. MSc. Thesis, University of British Columbia. 72pp.

APPENDIX I: THE SCRIPT

```

- #-----
- # Name:      Canopy gap identification and analysis
- # Purpose:
- #
- # Author:    rsweeney
- #
- # Created:   26/02/2019
- # Copyright: (c) rsweeney 2019
- # Licence:   <your Licence>
10 #-----

- def main():
-     pass

- if __name__ == '__main__':
-     main()

-     ###Importing Necessary Modules and Tools###
20
-     import numpy as np
-     import arcpy
-     import scipy
-     import scipy.stats
-     from scipy.ndimage import label
-     from scipy.ndimage import generic_filter
-     from arcpy import env
-     from arcpy.sa import *
-     arcpy.CheckOutExtension("Spatial")
30 import os
-     import math
-     arcpy.env.overwriteOutput = True
-     import Tkinter, tkFileDialog

-     ###Select Data to be used###
-     root = Tkinter.Tk()
-     Directory = tkFileDialog.askdirectory(parent=root, initialdir="/",
39 ..... title='Please select the geodatabase that contains your CHM and FRI file')
40
-     arcpy.env.workspace = Directory

-     inraster = raw_input("What is your CHM file called?")
-     standFC = raw_input("What is your FRI file called?")

-     arcpy.env.workspace = r"E:\Thesis\CHM1Again.gdb"
-     inraster = "CHM1"
-     standFC = "FRI1"

50 #datasets = arcpy.ListDatasets(feature_type="Feature")
-     #for dataset in datasets:
-         #print(dataset)

-     #Creation of Array1 from Raster CHM Image
-     inrasterP = arcpy.Raster(inraster)
-     Array1=arcpy.RasterToNumPyArray(inraster)

-     #Set negative values to 0, as CHM's should not contain negative values
60 Array1 = Array1.clip(min=0)

```



```

120      ###Converting discovered "Gap Data" to a form that can efficiently be used with FRI data (Polygon Feature Class)###
-
-
- #Convert RG raster to integer type, and run segmentation tool (Region Group)
- rgrwf_rst_INT = Int("rgrwf_rst")
- rgrwf_rst_INT.save("rgrwf_rst_INT")
-
- #Setting non-gap cells to "No Value" - Not allowing me to use Spatial Analyst Toolbox
- Raster_Gaps_Only = SetNull ('rgrwf_rst_INT' == 0, 'rgrwf_rst_INT')
- Raster_Gaps_Only.save("Raster_Gaps_Only")
130 Segmented = RegionGroup ("Raster_Gaps_Only", "EIGHT", "", "", 0)
- Segmented.save("SegmentedRaster")
- arcpy.RasterToPolygon_conversion("SegmentedRaster", "RasterToPoly1", "NO_SIMPLIFY")
-
- #Removing "Non-Gap" area from output feature class
- with arcpy.da.UpdateCursor("RasterToPoly1", "gridcode") as cursor:
-     for row in cursor:
-         if row[0] == 0:
-             cursor.deleteRow()
-
140 #This will delete any entries greater a given area value (1000m2) - remove "##" if desired
- with arcpy.da.UpdateCursor("RasterToPoly1", "Shape_Area") as cursor:
-     for row in cursor:
-         if row[0] > 1000: #Update or delete this to change maximum gap size Limit
-             cursor.deleteRow()
-
- #This will delete any entries less than a given area value (0m2)- remove "##" if desired
- with arcpy.da.UpdateCursor("RasterToPoly1", "Shape_Area") as cursor2:
-     for row in cursor2:
-         if row[0] < 0: #Update or delete this to change minimum gap size Limit
150         cursor.deleteRow()
-
- #Giving "RasterToPoly1" same spatial reference as the FRI feature Class
- dsc = arcpy.Describe(standFC)
- coord_sys = dsc.spatialReference
- arcpy.DefineProjection_management("RasterToPoly1", coord_sys)
-
-
-     ###Altering FRI data using Gap Data###
-
160 #Creation of fields to be populated based on coming calculations
- if len(arcpy.ListFields(standFC,"Gap_Clustering"))==0:
-     arcpy.AddField_management(standFC, "Gap_Clustering", "STRING")
- if len(arcpy.ListFields(standFC,"GapCoverage"))==0:
-     arcpy.AddField_management(standFC, "GapCoverage", "FLOAT")
- if len(arcpy.ListFields(standFC,"Gaps_Description"))==0:
-     arcpy.AddField_management(standFC, "Gaps_Description", "STRING")
- if len(arcpy.ListFields(standFC,"NumberOfGaps"))==0:
-     arcpy.AddField_management(standFC, "NumberOfGaps", "SHORT")
-
170 #These fields offer explanatory information, but are deleted if they are not required by the user. - remove "##" if desired
- ## if Len(arcpy.ListFields(standFC,"Gap_Area"))==0:
-     ## arcpy.AddField_management(standFC, "Gap_Area", "FLOAT")
- ## if Len(arcpy.ListFields(standFC,"Clustering_Z"))==0:
-     ## arcpy.AddField_management(standFC, "Clustering_Z", "FLOAT")
- ## if Len(arcpy.ListFields(standFC,"pValue"))==0:
-     ## arcpy.AddField_management(standFC, "pValue", "FLOAT")
-
- #List of fields to be accessed and/or written to in FRI data
- stfields = ["SHAPE@", "Gap_Area", "Clustering_Z", "SHAPE@XY", "POLYID", "SHAPE@AREA", "pValue", "GapCoverage", "Gaps_Description", "NumberOfGaps", "Gap_Clustering"]
180
-
- #Creation of Feature Layer from RasterToPoly1 feature class for use with "INTERSECT" method
- gapFC = "RasterToPoly1" #Gap Feature Class
- arcpy.MakeFeatureLayer_management(gapFC, "gapFC_lyr") #point shapefile's temporary Layer

```

```

#Accessing fields for reading and writing within FRI data
with arcpy.da.UpdateCursor(standFC, stfields) as SCursor:

190 for SRow in SCursor:
    poly = SRow[0]
    gapDesc = "NO"
    GapClustering = "NA"

    arcpy.SelectLayerByLocation_management("gapFC_lyr", "INTERSECT", poly)

    cnt = int(arcpy.GetCount_management("gapFC_lyr").getOutput(0)) #Gets number of gap polygons contained in each stand

200
    SRow[10] = GapClustering
    SRow[9] = cnt
    if cnt == 1 or cnt == 2:
        gapDesc = "OU"
    elif cnt >= 3:
        gapDesc = "OC"
    SRow[8] = gapDesc
    SCursor.updateRow(SRow)

210
    if cnt == 0:
        continue

    #Setting Lists and variables to empty before Loops
    GArea = []
    MinDistList = []
    GapXMin = []
    GapYMin = []
    GapXMax = []
    GapYMax = []
    TotalArea = 0.0

220
    #This Section can be used if a rectangular area encapsulating all gap polygons is to be used instead of stand area in the clustering calculations
    with arcpy.da.SearchCursor("gapFC_lyr", "SHAPE@") as GCursor:
        for GAProw in GCursor:
            INTGap = SRow[0].intersect(GAProw[0], 4)
            GArea.append(INTGap)
            #print("stand", SRow[4], INTGap.getArea())
            TotalArea = TotalArea + INTGap.getArea()
            SRow[1] = TotalArea
            SRow[7] = (TotalArea / SRow[5])*100
            SCursor.updateRow(SRow)

230
            #Calculating the rectangular extent area for optional use in clustering calculations
            GapXMin.append(GAProw[0].extent.XMin)
            GapXMax.append(GAProw[0].extent.XMax)
            GapYMin.append(GAProw[0].extent.YMin)
            GapYMax.append(GAProw[0].extent.YMax)
            XMinExtent = min(GapXMin)
            XMaxExtent = max(GapXMax)
            YMinExtent = min(GapYMin)
            YMaxExtent = max(GapYMax)
            Rectangle = arcpy.Extent(XMinExtent, YMinExtent, XMaxExtent, YMaxExtent)
            RectanglePoly = Rectangle.polygon
            RectangleArea = RectanglePoly.area #This is the area of a rectangle that encloses all gap polygons

240
    #Lists emptied before Loop
    DISTLIST = []
    MinDistList = []

```

```

2540 #for gap polygons within each stand
for shapes in GArea:
    for shapes2 in GArea:
        if shapes2 != shapes:
            DistTo = shapes.distanceTo(shapes2)
            DISTLIST.append(DistTo) #DISTLIST becomes a List of ALL distances from a single gap intersection to the others in a single stand
        if len(DISTLIST) > 0:
            MinDistList.append(min(DISTLIST)) #The Lowest distance value is taken from DISTLIST and moved to a new List, storing ALL minimum
            SRow[8] = gapDesc
            SCursor.updateRow(SRow)

2560 #If more than 1 gap is present in a stand, feature-clustering calculations are performed
if len(MinDistList) > 1:
    print("stand min dist", SRow[4], sum(MinDistList),len(MinDistList))

    Do = (sum(MinDistList)/float(len(MinDistList)))
    De = ((0.50)/(math.sqrt(len(MinDistList)/SRow[5])))
    SE = (0.26136)/(math.sqrt(float(len(MinDistList))**2/SRow[5]))
    ZScore = ((Do-De)/SE)
    pval = scipy.stats.norm.sf(abs(ZScore))*2 #calculates the p val for a two-sided test

2570 #If there are 1 or fewer gaps in a stand, no calculations are performed
else:
    Do = None
    De = None
    SE = None
    pval = None
    ZScore = None

2580 #Qualitatively describing canopy gap distributions within each stand
if ZScore >= -1.96 and ZScore <= 1.96:
    GapClustering = "r"
elif ZScore < -1.96:
    GapClustering = "c"
elif ZScore > 1.96:
    GapClustering = "d"

if SRow[9] == 1:
    GapClustering = "NA"

2590 SRow[10] = GapClustering
SCursor.updateRow(SRow)

#Printing relevant outputs
print(SRow[4], SRow[5], ZScore, pval, len(MinDistList),Do, De, SE, RectangleArea)
print("-----")

#Populating ZScore and pValue fields in FRI
SRow[2] = ZScore
SRow[6] = pval
SCursor.updateRow(SRow)

```

```

3000 ###Deleting Unnecessary Files###
- arcpy.Delete_management(Raster_Gaps_Only)
- arcpy.Delete_management("rgrwf_rst")
- arcpy.Delete_management("SegmentedRaster")
- #arcpy.Delete_management("rgrwf_rst_INT") #Cannot be deleted due to schema lock
- arcpy.DeleteField_management(standFC, ["Clustering_Z", "pValue", "Gap_Area"]) #These fields are deleted as they are not integral, though may be useful

3100 ###Renaming Gap Feature Class Output relative to input FRI FC name###
- FRINameD = arcpy.Describe(standFC)
- FRIName = FRINameD.name
- Rename = FRIName + "Gaps"
- in_data = arcpy.env.workspace + "\RasterToPoly1"
- data_type = "FeatureClass"
- arcpy.Rename_management(in_data, Rename, data_type)

```