# Index for Quantifying Geometric Point Disorder in Geospatial Applications

S. Jones[1*] and H.G. Momm[1]

[1]Department of Geosciences, Middle Tennessee State University, Murfreesboro, Tennessee

* corresponding author: rj3h@mtmail.mtsu.edu

## Abstract:

Many techniques have been developed to quantify different conceptualizations of self-interaction and patterns within spatial data. We propose a new metric and related algorithm that describes the geometric spatial disorder of geographic point sets, the "Index of Disorder" (IoD). The IoD algorithm was applied to synthetic and natural datasets and was shown to be able to differentiate between areas of high spatial disorder (randomly placed points) and low spatial disorder (e.g., curvilinear grids, wallpaper groups, and other repeating patterns). Because the IoD is a quantitative metric, it can be used on its own as an aid for identifying areas of unusually high or low spatial disorder or as enrichment for machine learning classification algorithms.

## Keywords:

Point Patterns; Classification; Homogeneity, Algorithms; Forestry, Planning

## One Sentence Summary:

The spatial disorder of any arbitrary point in a set of points can be quantified by comparing the relative positions of that point's neighbors to the relative positions of its neighbors' neighbors.

# 1 Introduction

The acquisition of large amounts of spatial data has generated the need for automated methods to convert this data into usable information. A significant number of established methods used to detect patterns are available, particularly those statistical or locational in nature, such as spatial autocorrelation, outlier detection, clustering hotspots, and regression (Shekhar, Evans, Kang, & Mohan, 2011). However, there is a relative scarcity of methods to analyze and identify patterns purely comprised of the relationships of the coordinate positioning of points of geospatial data. Such “geometric patterns” are evident in the locational structure of many types of geospatial data, such as the neatly ordered placement of trees in an orchard or grids of house centroids in a planned housing development. Such geometric order contrasts with the relative disorder of naturally occurring trees and unplanned developments, which display little to no geometric structure as result of the relatively random positioning of each element. Geometric structure, or lack of structure, is often immediately apparent to human investigators, yet is surprisingly difficult to define mathematically (Antuono, Bouscasse, Colagrossi, & Marrone, 2014), and for this reason there is a relative dearth of methods that can be used to analyze geometric patterns in geospatial contexts.

For example, land cover classification is one of the most widely used and accepted applications of geospatial technologies. Land cover classification studies have been conducted using a wide array of data sources, mostly raster grids derived from aerially- or satellite-collected imagery (Lou, et al., 2016; Phiri & Morgenroth, 2017). The majority of landcover studies treated individual pixels as isolated data points, ignoring the information contained in the spatial relationships between pixels. Different set of methods have been developed to discriminate between features based on image texture and the frequency of tonal changes based on pre-determined texture operators (Momm, Easson, & Kuszmaul, 2009). Haralick textures were among the first proposed metrics that quantified the spatial relationships between adjacent pixels (Haralick, Shanmugam, & Dinstein, 1973), and have since been used to increase the accuracy of land cover classification studies (Momm, Easson, & Kuszmaul, 2009). Subsequent developments, such as so-called contextual classification methods, further sought to quantify the information contained within spatial pixel-neighbor relationship (Swain & Stephen B. Vardman, 1981). Quantification of lacunarity (self-similarity) has also been used as a way to quantify scale-dependent heterogeneity in raster imagery as a way to aid differentiation of ordered and disordered land use (Dong, 2000). Additional work has shown that textural periodicity is suggestive of certain landcover types (Trias-Sanz, 2006). Over time, a number of additional quantitative metrics of landcover textures and patterns have been developed as well (Remmel & Csillag, 2003). Each of these methods extracts some meaning from the spatial relationships of data, but they are not suited for the detection of geometric patterns: each method is merely a statistical operator on a windowed subset of data, which inherently preserves some information about spatial relationships, but the information about the positioning of data points within the subset is ignored.

Furthermore, because imagery and elevation models are most easily represented as raster datasets, the bulk of methods developed to quantify spatial textures and patterns in geospatial contexts are applicable to raster grid data only. Yet not all geographic information can or should be represented as raster data, such as the coordinate locations of discrete features of interest. Extensive work has been done on the development of algorithms for detecting regularly repeating structures in point clouds, but these methods have been primarily applied in non-geographic contexts (Pauly, Mitra, Wallner, Pottmann, & Guibas, 2008). Additionally, these algorithms do not make an attempt to quantify the level of regularity of individual points in the point cloud even though the regularity of the point cloud is detected quantitatively. *Antuono et al.* suggest an algorithm to quantify disorder within simulated fluid-particle systems, but this algorithm makes the assumption that disorder is defined as the deviation from a grid-like structure, an assumption that is likely valid for fluid simulations but fails in the context of geographic systems where order and disorder might be more loosely defined (Antuono, Bouscasse, Colagrossi, & Marrone, 2014). Because of this, geometric patterns such as row crop spacing or patterns of building footprints in planned communities are an aspect of geospatial data that is often ignored, and there is a deficiency of methods that can quantify the disorder of geospatial point data without relying on *a priori* assumptions about what patterns constitute “order”.

To quantify the geometric order/disorder of geospatial point datasets, a new metric and related algorithm is described. The proposed metric, referred to as “Index of Disorder” (IoD), quantitatively describes the geometric spatial order/disorder of geographic point sets. Calculation and assignment of an IoD score to each point allows the identification of areas of relatively high or low point pattern disorder. The main objectives of this study are three-fold: (1) describe the IoD assumptions, parameters, and calculation, (2) discuss the interpretation of IoD scores, and (3) assess the IoD’s performance through its application to synthetic and natural geospatial datasets.

# 2 Methods

## 2.1 Algorithm Description

The IoD is designed to quantify similarities in point spatial pattern by comparing multiple “neighborhoods” of points; in which a neighborhood is defined by a set of all points within a user-specified distance of an individual point. In each pair of neighborhood analysis, an IoD sub-score is calculated for the point being investigated. The IoD for a point is obtained by calculating the IoD sub-score for that point and each of its neighbors and taking the mean of these scores.

If two neighborhoods are similar, then the relative positions of corresponding points will be close together; conversely, if two neighborhoods are dissimilar, then corresponding points will be far apart. A point set that is “ordered” will have a neighborhood that is, on average, similar to the neighborhoods of all of its neighbors, and thus the typical point-pair deviation will be small. Because this is a relative rather than absolute metric, an arbitrary point set could potentially have any IoD value since the IoD quantifies the disorder of a point relative to the other points in the set and in accordance with the algorithm's parameterization. Consequently, ordered points do not necessarily have a low IoD but rather have an IoD that is lower than the IoD of disordered points in the same dataset at the scale of interest.

The calculation of a the IoD sub-score between two points $p_1$ and $p_2$ which are part of the set of points $P$ is as follows (*Figure 1*). A parent point $p_1$ is defined (red point in *Figure 2*A) and its neighbors $N_1$ selected based on user-provided threshold distance (*Figure 2*B). Similarly, a child point $p_2$ is defined and its neighbors $N_2$ selected (blue points in *Figure 2*C). Absolute X and Y coordinates values for all points in neighborhoods $N_1$ and $N_2$ are transformed to a relative coordinate system in which both parent and child points are located at the origin (*Figure 2*D). Each point in $N_1$ is assigned to a unique point in $N_2$ such that the mean cost function for point assignment is minimized (*Figure 2*E). The mean value of the cost function is the IoD sub-score (*Figure 1*). The process continues by recording the sub-score and repeating the operation by selecting a new child point (*Figure 2*F). Optionally, prior to point pair assignment, it is possible to apply a second coordinate transformation to $N_2$ to further minimize the cost of point assignment (*Figure 1*). Details of the implementation of the point pair assignment and coordinate realignment are provided in subsequent sections.

The IoD sub-score is not the IoD itself, but rather an intermediate parameter used in the calculation of the IoD for a given point. A low IoD sub-score implies geometric similarity between the neighborhoods of two particular points, whereas a low IoD implies the relative presence of a geometric ordered pattern around a given point.

### 2.1.1 Point Pair Assignment

Calculation of the IoD sub-score for two sets of points involves point pair assignment, where every point in one set (neighborhood $N_1$) is assigned to a point in the other set (neighborhood $N_2$) with one-to-one correspondence (each point in each set has one and only one assignment). Additionally, a goodness-of-fit metric (score) is calculated for every point-pair assignment that describes how good the assignment is. The objective is to find an optimal set of point-pair assignments such that that average cost of assignment is minimized. The cost function for point-pair assignment is arbitrary, though the advantages of certain cost functions over others are discussed in 3.3.1. A readily apparent cost function for point-pair assignment is the Euclidean (ordinary straight line) distance between the two members of a point pair—two points that are close together are a more obvious match than those that are far apart—but other cost functions such as the square root of the distance or sigmoidal functions (Figure 3) that relate Euclidean distance to a score are viable as well. This assignment of correspondence is a specific example of the so-called "assignment problem".

The assignment problem is formulated here such that there exists a complete bipartite graph composed of two sets parent neighborhood vertices and child neighborhood vertices. Each candidate solution connects a parent to a child neighborhood (representing a possible point-pair assignment) and has an associated cost $c(i, j)$. If the sets are not of equal size, then the graph is necessarily not complete, but the graph can be made complete by adding some number of arbitrary vertices to the smaller set and connecting them to the larger set with an arbitrarily large assignment cost. When a solution is found that minimizes the assignment costs, the arbitrary vertices and edges are removed. Thus, the larger set will have some number of unpaired points.

Solving the assignment problem through brute force implies a time complexity of $O(n!)$, where $n$ is the number of points being matched, though many algorithms have been proposed that solve the problem with better efficiency. The point pair assignment algorithm used in this study is the Hungarian method as implemented in the scientific computing library *scipy* (Oliphant, 2007 and Millman and Aivazis, 2011). The Hungarian method solves the assignment problem in polynomial time ($O(n^3)$), a significant improvement on the factorial time complexity of the brute force solution (Kuhn, 1955). The algorithm works by reframing the problem as an *n x n* cost matrix *M* where the cost between $P_i$ and $P_j$ is $c(i, j)$. The solution is the set *S* of *n* entries that minimizes the sum of the set while respecting one-to-one correspondence. In other words, the solution set contains exactly one entry from each column and each row of *M*. The Hungarian algorithm takes advantage of the fact that subtracting a constant from a row or column does not change the set of optimal entries, and so by repeatedly subtracting choice values from rows and columns the assignment problem can be reduced to a trivial form where some of the matrix entries are 0 (Kuhn, 1955).

### 2.1.2 Point Set Realignment

In some cases it may be desirable to realign neighborhoods for better correspondence if it is suspected that spatial patterns in the data may become offset (such as in Figure 4) or rotate, scale or flip as a function of space. This process of realignment is sometimes referred to as point registration. Most point registration methods require assignment of correspondence; it is natural to use the same cost function laid out in the previous section as the cost function for correspondence assignment.

Although multiple methods are available and any of them could have been used herein, the optional step of point set realignment for the IoD algorithm implemented in this study was done using the Iterative Closest Point (ICP) method (Besl & McKay, 1992). The algorithm iteratively registers points by assigning all points in $N_1$ to points in $N_2$ by finding the optimal point-pair assignments, and then finds the least squares rigid transformation that minimizes the point-pair assignment cost. The process is repeated until the point-pair deviation falls below a threshold or the algorithm exceeds the pre-determined number of iterations.

Though ICP realignment is not used in every instance of the IoD demonstrated in this paper, when ICP realignment is used the same point-pair assignment cost function is used for both ICP realignment and assigning correspondence between neighborhoods outside of the ICP algorithm.

### 2.1.3 Scoring Function

Once two neighborhoods have been assigned to one another, some function must be defined that relates the displacement between two assigned points with a score describing the quality of the match, which we call the “assignment score” for that point. It is convenient to define a scoring function that has a lower bound of 0 (to describe no disorder) and an upper bound of 1 (to describe maximal disorder), though such a range is arbitrary. It is also natural to use the same function used to calculate assignment cost as the scoring function; doing so ensures that the calculated point correspondence results in the minimum score possible.

### 2.1.4 Unpaired points

Neighborhoods being compared may not, and typically do not, have the same number of points, and this will result in one or more unpaired points during point pair assignment. Because the distance between a point and its assigned partner is used to calculate the point’s assignment score, a decision must be made regarding how to calculate assignment scores for unpaired points.

The simplest option is to ignore unpaired points when calculating the mean assignment cost. An obvious drawback is that this approach would be that, overall, a neighborhood with few points has a low assignment cost to any arbitrary neighborhood with many points because each point in the sparsely populated first neighborhood has a high likelihood of being matched very closely to a point in the densely populated second neighborhood, which would result in depression of the IoD despite the fact that no true point correspondence necessarily exists.

A second option is to penalize unmatched points by some arbitrary amount, which would prevent IoD depression when sparse neighborhoods are compared to dense neighborhoods. However, there are some circumstances where two neighborhoods may show the same pattern, but the pattern in one neighborhood is truncated because e.g., the neighborhood’s parent point lies at the edge of a pattern, such as is the case in Figure 4. In this case, penalizing unpaired points would raise the IoD even though true pattern correspondence exists.

A final and generally superior option is to punish unpaired points if and only if they are within the convex hull of the subset of points that have an assignment. In this paper, the IoD algorithm always uses this third option. This allows for patterns in compared neighborhoods to be of unequal spatial extent while still disallowing noisy points from depressing the IoD. The convex hull of a set is defined as the smallest convex set that contains it. In special case where the set consists of 2-dimensional points, then this can be visualized as the area enclosed by a rubber band stretched over the set. Illustration of this option is presented in Figure 4.

## 2.2 Theoretical Evaluation

The potential of the IoD to differentiate ordered from disordered points was assessed through the development of 6 synthetically generated datasets (Figure 5). Each of these synthetic datasets were generated by perturbing a pattern with noise of increasing strength outside an arbitrary distance from the origin. Thus, the order of the pattern is entirely preserved within the arbitrary radius, but the system becomes increasingly disordered beyond that radius. Multiple patterns were investigated: square grid with 10% of the points randomly removed (Figure 5a), rectangular grid modified with a sinusoidal function (Figure 5b), pattern formed by overlapping two offset rectangular grids (Figure 5c), pattern formed by overlapping three offset rectangular grids (Figure 5d), pattern formed by overlaying concentric circles with equal linear point densities (Figure 5e), and pattern formed by overlapping a square grid with a copy of the same grid rotated 45 degrees (Figure 5f).

All synthetic datasets were generated using evaluated with an implementation of the IoD that uses a sigmoidal function for point assignment and assignment scoring. Each synthetic point set was evaluated twice: with and without iterative closest point realignment, but with no other changes to algorithm parameters. When ICP realignment was used, the same sigmoidal function used for point assignment and assignment scoring was used to calculate correspondence within the ICP algorithm. An explanation of the parameters used can be found in Table 1. The values adopted for the theoretical evaluation were a neighborhood radius of 15 units, deviation value at which sigmoidal scoring funding assigns a score of 0.5 ($K_m$) of 3 units, degree of cooperativity of the sigmoidal scoring function of 0.5, assignment score given to unpaired points of 1, punishment of points outside of the convex hull set to false, Euclidian distance selected as cost function, and reorientation of points set to false.

The neighborhood radius is visualized as a blue circle for all synthetic datasets. The larger red circle visualizes the distance at which application of noise to the dataset begins, and the mean amplitude of the noise increases linearly as a function of the distance from this red circle (Figure 5).

## 2.3 Applied Evaluations

The IoD was applied to three geospatial datasets, two consisting of tree crown locations and one consisting of building centroids. Each dataset is known to contain at least one area of intrinsic order. The quality of this differentiation (quantitative comparison of manually determined and threshold applied to IoD) was quantified using Cohen's kappa coefficient (κ) of agreement, a measure of inter-rater reliability that considers the possibility of chance agreement.

### 2.3.1 Tree Crowns

LiDAR (Light Detection and Ranging) point cloud datasets were acquired from United States Geological Survey's (USGS) National Map repository for two study areas. The LiDAR-processing software suite LASTools (Isenburg, 2019) was used to generate multiple rasterized elevation models. Digital elevation models (DEMs) represent the elevation of the bare earth. Digital surface models (DSMs) represent the elevation of the earth including partially reflective elements and were created by using only the first returns from the point clouds. Digital height models (DHMs), sometimes called canopy height models, were created by subtracting the DEMs from the corresponding DSMs, which results in a raster that represents the height of reflective elements such as tree canopies above the ground.

Automated detection of tree crowns from raster height models is a topic that has been extensively researched, and so many methods to perform this function (Zhen, Quackenbush, & Zhang, 2016). Tree crown locations in this study were extracted using the Laplace of Gaussian blob detection algorithm implemented in the Python package scikit-learn (Pedregosa, et al., 2011) on the DHM raster grids of each study area.

Each tree crown centroid point set was then evaluated using the IoD algorithm. The IoD algorithm used a sigmoidal scoring function for both study areas, and this same function was used as the cost function for point assignment.

#### 2.3.1.1 Site 1 – Apple orchard in TN, USA

Orchards are collections of intentionally planted trees or shrubs that are maintained for food production. For ease of harvest and maintenance, orchards are typically planted in regular patterns that contrast with the disordered distribution of naturally occurring tree stands. The most common planting patterns are the square and rectangular planting patterns, and the predictability of these patterns has allowed for improved automated differentiation of orchards from natural tree stands on the basis the textural characteristics of orthoimagery (Aksoy, Yalniz, & Tasdemir, 2012). However, alternative patterns such has single and double hedged, hexagonal, quincunx and topographically contoured planting systems are also used depending on the needs of the agriculturist (Khan, Al-Yahyai, & Al-Said, 2017).

The IoD was applied to tree crown centroids extracted from a 7.5 km$^2$ area in small commercial apple orchard in Tennessee, USA. The study area is rural and consists of two distinct planted areas of rectangularly gridded apple trees trending northwest and northeast. Mature natural forest is also present to the west and north of the orchard as well as sparse, isolated trees that are also naturally occurring. The roughly rectangular grid patterns that comprise the orchard contrast with the disorder of the natural forest in the surrounding area.

LiDAR-based three-dimensional point clouds for the study area was collected by the U.S. Geological Survey (USGS) between January and April of 2019. Trees crown centroids were manually classified as being part of the orchard using both the LiDAR-derived elevation models and contemporaneous aerial imagery. The IoD was calculated for all tree centroids in the study area, and trees were classified as “disordered” if their IoD was above an arbitrary IoD threshold and “ordered” if below the threshold. “Ordered” trees were assumed to correspond to orchard trees and a sensitivity analysis was conducted to characterize the effect that varying the scoring function’s $K_m$ and the neighborhood radius has on the classification quality. The cooperativity of the scoring function was held constant at 5 during the sensitivity analysis.

#### 2.3.1.2 Site 2 – Reforested Area in Mooresville, NC

Trees may be intentionally planted in order to reduce wind exposure, reforest clear-cuts for ecological or timber-harvesting purposes, or create riparian buffers along streams and rivers. Though random or pseudorandom planting patterns are preferred when the planted trees are intended to provide ecological improvements, most reforested areas are planted with a square or rectangular pattern similar to the patterns seen in many orchards. Thus, most planted tree stands exhibit pattern regularity not found in natural tree stands.

Site 2 is a 0.6 km$^2$ area centered on a partially reforested zone located in Mooresville, NC, USA. The study area is rural and consists of linear rows of young planted trees abutted by older natural mature forest. The trees comprising the replanted areas form roughly rectangular grid patterns of varying orientation that contrast with the disorder of the natural forest.

LiDAR for the study area was collected by the USGS between February of 2016 and February of 2017. Trees were manually classified as being part of the replanted zone or not using both the LiDAR-derived elevation models and contemporaneous aerial imagery. The IoD was calculated for all trees in the study area, and trees were classified as “disordered” if their IoD was above an arbitrary IoD threshold and “ordered” if below the threshold. “Ordered” trees were assumed to correspond to replanted zones and a sensitivity analysis was conducted to characterize the effect that varying the scoring function’s $K_m$ and the neighborhood radius has on the classification quality. The cooperativity of the scoring function was held constant at 3 during the sensitivity analysis.

#### 2.3.1.3 Site 3 – Neighborhood in Nashville, TN

Urban planners have long noted the morphological peculiarities of unplanned or “organic” city growth as compared to those of planned growth. Planned portions of cities are marked by regularity of building placement and street network orientation, while organic growth does not (Nilsson & Gil, 2019). Furthermore, these differences have been exploited in order to quantify the degree to which cities are the result of planned or organic growth (Boeing, 2019). On a smaller scale, residential communities often exhibit similar spacing patterns between homes, particularly when the area is one that is relatively planned. Auxiliary structures, such as sheds and garages, are more irregularly spaced have “neighborhoods” (in the sense of the IoD algorithm) that are offset from surrounding buildings.

Lockeland Springs is a mixed-use neighborhood in Nashville, Tennessee approximately 2 km$^2$ in size. It is densely developed compared to Sites 1 and 2, though most buildings are still low-rise. Main buildings (residences and businesses) are developed in relatively regular rows within neighborhood blocks, while auxiliary structures such as sheds and garages are sporadically placed with no consistent spacing or lot positioning. Figure 6 shows a map of the neighborhood along with the classifications of each building footprint according to the City of Nashville, TN, USA.

Building footprints were obtained for the metropolitan Nashville area from the Tennessee Department of Finance and Administration via the geospatial data aggregator Koordinates (koordinates.com). In addition to spatial information, these data include additional information about buildings such as whether they are main or auxiliary structures. The IoD was then calculated for the extracted centroids, and buildings were classified as “disordered” if their IoD was above an arbitrary IoD threshold and “ordered” if below the threshold. “Ordered” buildings were assumed to correspond to “main” buildings, while disordered buildings were assumed to correspond to auxiliary structures. A sensitivity test was conducted to characterize the effect that varying the scoring function’s $K_m$ and the neighborhood radius has on the classification quality. The cooperativity of the scoring function was held constant at 5.

# 3 Results and Discussion

## 3.1 Theoretical Evaluation

In Figure 5, results from the synthetically generated patterns and their IoD scores are described. The inner blue circle visualizes the neighborhood size, while the larger red circle shows the radius outside of which the pattern becomes increasingly perturbed with noise. For point patterns generated based on regular grids, the IoD differentiates greatly between the unperturbed points (within red circle) and perturbed points (outside red circle in Figure 5a-d). In more complex patterns, the IoD yields milder to moderate differentiation between the unperturbed and noisy patterns than the patterns based on regular grids. Nonetheless, in all cases the lowest IoD values are observed in the center of the figures, and a rapid increase in the IoD is observed as the pattern deviation approaches and exceeds the $K_m$ of the scoring function, indicating that the IoD is capable of measuring relative levels of disorder and order within a dataset. Importantly, the IoD is agnostic to the general form of patterns, and thus is capable of detecting patterns with no knowledge of the exact form of the patterns.

Because pattern detection is scale-dependent, alternative parameterization (particularly changes to the neighborhood radius and $K_m$) will result in varying levels of discrimination between ordered and disordered points. Optimal parameterization is generally simpler to achieve when patterns are also simple.

Realignment of neighborhoods during calculation of the IoD generally depresses both the unperturbed and noisy perturbed points by a similar amount (*Figure 7*). In these cases, there is no increase in differentiation of the unperturbed and noisy patterns by the IoD. However, in certain cases, realignment can cause an appreciable change in differentiation (*Figure 8*). The discrete repeating patterns in these examples, sometimes referred to as “wallpaper groups” (Liu, Collins, & Tsin, 2004), benefit greatly from realignment because the effect of pattern offset (illustrated in *Figure 8*) becomes significant; not realigning points will elevate the IoD even though pattern correspondence exists. Allowing neighborhood realignment can significantly increase computation time because point registration may be repeated multiple times per neighborhood rather than just once, so realignment should only be used when pattern offset is anticipated to occur and needs to be corrected for. In some cases, pattern offset may actually be of critical importance for feature identification, and so realignment is undesirable.

## 3.2 Natural Evaluations

### 3.2.1 Site 1

Sensitivity analysis was performed by varying IoD input parameters of neighborhood radius and sigmoidal assigned threshold value and comparing results with reference datasets (Table 3). The maximum kappa coefficient of agreement value of 0.81, interpreted as "almost perfect agreement" (**Error! Reference source not found.**) is achieved when the scoring function has a $K_m$ of 5 and the neighborhood radius is 80 meters. The corresponding overall accuracy for this classification is 96%.

The mild planting pattern heterogeneity of the orchard increased the calculated IoD somewhat, but overall the orchard trees are largely differentiable from the surrounding forest due to the gridded nature of the orchard (Figure 9). The apparent heterogenic pattern of the orchard is likely as much due to inaccuracies in crown extraction from the DHM than is it is due actual heterogeneity; crown extraction from LiDAR is itself highly parameter dependent. The high kappa value of this classification suggests that the IoD alone is sufficient to differentiate the orchard from surrounding trees without any other supporting data.

The classification quality of the IoD is highly sensitive to its parameterization (Table 3). Because patterns are a fundamentally scale-dependent phenomenon, it is not surprising that algorithms that quantify them must be parameterized appropriately. The neighborhood radius used to parameterize the IoD describes the scale of the anticipated patterns, while the $K_m$ describes the level of expected deviation within the patterns. Though the neighborhood radius is a parameter that will be present in any implementation of the IoD, Km is technically a parameter of the sigmoidal function used for scoring and point assignment. It is important to note that different functions can be used for these purposes, but their parameters will have different interpretations. If the classification quality is maximized and a strongly homogenous pattern is being differentiated from a highly disordered nonpattern, then the neighborhood radius and $K_m$ respectively characterize the actual pattern scale and deviation. Thus, the characteristic scale of the orchard in Site 1 is between 70 to 80 meters, and the threshold of the pattern deviation before the points become disordered is approximately 5 meters.

Because of this dependence on sensible, scale-aware parameterization, the IoD is not appropriate for quantification of patterns for which there is no prior knowledge of scale of pattern deviance, or if investigators wish to simultaneously quantify patterns with disparate scales simultaneously. Investigations utilizing the IoD may benefit from first using the IoD the characterize pattern scales on a subset of data before applying the IoD more broadly.

### 3.2.2 Site 2

Similarly, sensitivity analysis of was performed for IoD calculations in Site 2 (Table 4). The maximum kappa value of 0.74, interpreted as “substantial agreement” is achieved when the scoring function has a $K_m$ of 2 and the neighborhood radius is 25 meters. The corresponding raw accuracy for this classification is 87%.

Like Site 1, the planted trees in this study area display a gridded structure that explains the lower IoD in the planted zones relative to the surrounding mature forest (Figure 10). Though the kappa value for this site indicates that the classification agreement is not as strong as that of Site 1, the agreement is sufficiently strong to suggest that the planted trees are differentiable from the surrounding trees based on the IoD alone. As with Site 1, the sensitivity test (Table 4) suggests a characteristic scale and pattern deviance for the planted trees, which are 25 meters and 2 meters respectively.

### 3.2.3 Site 3

Performing sensitivity analysis, the maximum kappa value of 0.44, interpreted as “moderate agreement” (Cohen, 1960), is achieved when the scoring function has a $K_m$ of 6.5 and the neighborhood radius is 19 meters (Table 5). The corresponding raw accuracy for this classification is 76%.

The classification agreement is significantly lower at this site than for Sites 1 and 2. This is unsurprising due to the increased pattern complexity displayed by the building centroids (Figure 11). While the planted trees at the previous sites displayed relatively simple grids patterns that contrasted with the highly disordered positioning of the naturally occurring trees, the buildings in this site are arranged in gridded blocks that vary in scale, orientation, and deviance. Though the main buildings generally adhere to straight lines within these blocks, the exact position of their centroids can vary along this line and occasionally there is no discernible pattern to their placement at all. Conversely, auxiliary buildings overall do not display the level of pattern adherence that the main buildings do—not every main building has an auxiliary building, and when an auxiliary building is present its placement on the property is relatively varied—but their placement is not perfectly random. Occasionally a block will have enough auxiliary buildings that a pattern similar to that of the main buildings emerges.

These phenomena together elevate the IoD of the main buildings and depress that of the auxiliary buildings, reducing the ability of the IoD alone to differentiate these building types. Depending on the desired classification agreement, the IoD itself may not be sufficient on its own for classification in complex systems. However, the IoD has some level of classification power even in complex systems and so can existing datasets, improving the accuracy of more complicated classification schemes such as decision trees and neural nets. The IoD could be calculated and added to a dataset, increasing its dimensionality.

Though peak classification quality is achieved with a neighborhood radius of 19 meters and a $K_m$ of 6.5 meters, the respective interpretation of these values as the characteristic pattern scale and pattern deviation of the study area is not necessarily as clear is it is for Sites 1 and 2, which consist of strongly patterned and nonpatterned points. The placement of building centroids in Site 3, in contrast, are moderately patterned (main buildings) or weakly patterned (auxiliary buildings). Because of this, the ideal neighborhood radius and $K_m$ do not necessarily describe either pattern bur rather represent, respectively, a discriminatory scale and a discriminatory deviance.

## 3.3 Impact of Alternative Implementations of the IoD

### 3.3.1 Scoring and Point Assignment Functions

For the purposes of scoring the spatial deviation of a pair of assigned points, swapping any monotonic function for another for the purposes of scoring will not change the relative ranking of the disorder of the points, and thus the choice of scoring function is ultimately an aesthetic choice. However, it is often convenient to use the same function used to calculate assignment costs to calculate the IoD in order to simplify interpretation of the output, and the assignment cost function does have an impact on what points are assigned to one another, and thus may have an impact on the relative ranking of IoD scores for points in a set. Consider the comparison of assignment methods in *Figure 4*. Using the Euclidean distance between points as the assignment function results in many suboptimal pairings; many points are assigned to a point for which there is no obvious correspondence but results in an overall minimization of the assignment cost. Using a sigmoidal function, however, allows for more intuitive assignments for most points by reducing penalties for assignments with large Euclidean displacements.

Other functions, such as the square root of the Euclidean distance, similarly reduce the penalty for large assignment displacements and generally improve assignment. An advantage of the sigmoidal function over other options is that the midpoint ($K_m$) imparts additionally scale-awareness to the IoD; the $K_m$ describes the expected deviation, or “noise” within a pattern, and so displacements below $K_m$ are considered within the tolerance of the expected noise level and punished less. Conversely, deviations beyond the noise threshold are punished harshly. Without $K_m$, the IoD is aware of characteristic pattern scales (via the neighborhood radius) but will be unable to account for intra-pattern noise.

### 3.3.2 Realignment Function

Though the Iterative Closest Point method is used in this paper’s implementation when realignment of neighborhoods is called for, there exists a great deal of alternative methods for realignment, typically called point set registration or point matching algorithms in the field of computer vision. Discussion of their differences and similarities is beyond the scope of this paper, but it is worth noting that all registration methods must either calculate or be provided with the correspondence between the point sets being aligned. For the IoD, it is sensible to use the same function for this calculating assignment here as is used elsewhere in the IoD.

Additionally, realignment algorithms are not guaranteed to “correctly” align the neighborhoods, and realigning neighborhoods may also lead to spurious depressions in the IoD even when no pattern similarity exists. If this spurious depression exceeds the reduction in IoD when true pattern correspondence exists, then realignment will actually reduce the ability of the IoD to differentiate ordered and disordered point sets. Additionally, in some cases it may be preferable to punish offset patterns if the goal is to e.g., detect points out of alignment with a grid. In such a situation then realignment is not desirable. Thus, the use of realignment may or may not be appropriate depending on the intent of the study. Realignment is also computationally expensive it requires repeated recalculation of neighborhood point assignments.

### 3.3.3 Handling of Unpaired Points

As mentioned previously, there are several ways to account for the presence of points in a neighborhood that have no assignment to a point in its corresponding neighborhood, something that occurs when comparing neighborhoods with an unequal number of points.

The simplest option is to ignore unpaired points when calculating the mean assignment cost. Though simple, this option would find that a neighborhood with few points has a low assignment cost to any arbitrary neighborhood with many points because each point in the first neighborhood has a high likelihood of being matched with a very close point in the second neighborhood, leading to a potentially spurious depression of the IoD. In order to prevent such spurious depression of the IoD, a second option is to penalize unmatched points by some arbitrary amount. Under this scheme the IoD depression is reduced in cases where one neighborhood has many more points than a comparison neighborhood.

There are some circumstances where two neighborhoods may show the same pattern, but the pattern in one neighborhood is truncated because e.g., the neighborhood’s parent point lies at the edge of a pattern, such as is the case in *Figure 4*. In this case, punishing unpaired points would depress the IoD even though there is true pattern correspondence between the two neighborhoods. A third and generally superior option is to punish unpaired points if and only if they are within the convex hull defined by the set of points that are assigned. This allows for patterns in compared neighborhoods to be of unequal spatial extent while still preventing noisy points from inadvertently depressing the IoD.

## 4 Conclusions

Existing methods to quantify disorder have relied on either raster data or non-geospatial algorithms that quantify the disorder of point data based on the assumption that order is grid-like, an assumption that is often violated in geospatial contexts. Thus, the Index of Disorder algorithm provides a new way to quantify spatial disorder of individual points in a set, which is achieved by quantifying the similarity of a point’s “neighborhood” to the neighborhoods of its neighbors.

Datasets evaluated in this study indicate that the IoD alone is sufficient to differentiate planted stands of trees, which tend to be planted in curvilinear grids, from mature forest, which displays no pattern in the positioning of trees. On this principle the IoD can be used to estimate reforestation extent or identify orchards. The IoD can also be used to enrich datasets for classification in systems where spatial patterns alone may not be sufficient to make a classification, such as when classifying building types in complex urban systems.

Because spatial patterns are inherently scale-dependent phenomena, the IoD requires parameterization in order to satisfactorily quantify disorder. Thus, its utility may be limited in systems where the scale of the patterns being analyzed is poorly understood or if the pattern scale is variable across the study area. However, this limitation allows the IoD to be used backwards: if there is a priori knowledge of the classification labels of points in a system, then then an optimization algorithm can be applied to the IoD in order to estimate the scale of the pattern and magnitude of the pattern deviation.

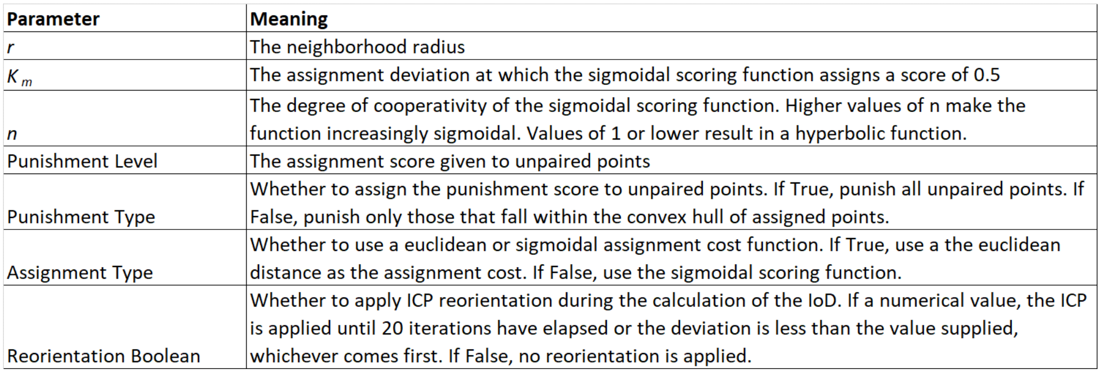
Because the measure is quantitative (though relative) it can also be used as an additional dimension of analysis for problems that benefit from data enrichment, such as machine learning classification. Further work is planned to explore in more detail the effects of alternative implementations of the IoD, as well as its utility in classification problems beyond the scope of what is presented here. In particular, the IoD may be of use in quantifying patterns present in 3-dimensional point sets.

## 5 Data and Materials Availability

All code used in the analysis for this paper is publicly available at https://github.com/rsjones94/point-disorder. The digital height models used to generate the tree crown datasets are available on request.

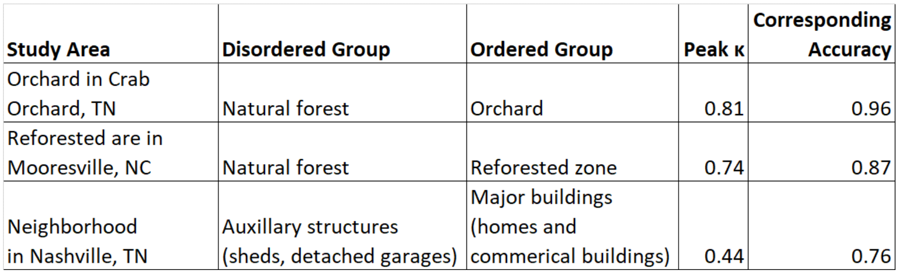
## 6 Tables and Figures

Table 1. Explanation of input parameters used in the implementation of the IoD demonstrated in this study.



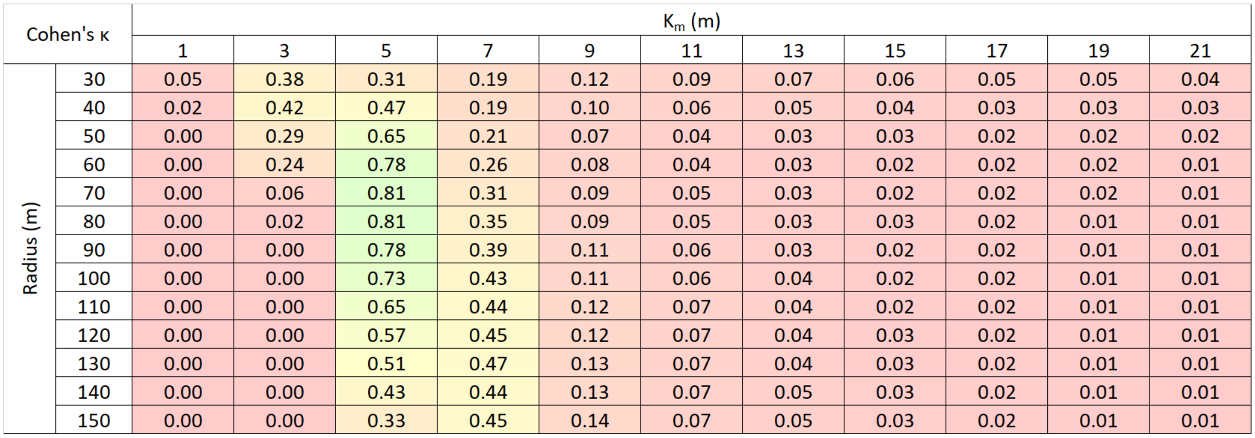
| **Parameter** | **Meaning** |
|---|---|
| *r* | The neighborhood radius |
| $K_m$ | The assignment deviation at which the sigmoidal scoring function assigns a score of 0.5 |
| *n* | The degree of cooperativity of the sigmoidal scoring function. Higher values of n make the function increasingly sigmoidal. Values of 1 or lower result in a hyperbolic function. |
| Punishment Level | The assignment score given to unpaired points |
| Punishment Type | Whether to assign the punishment score to unpaired points. If True, punish all unpaired points. If False, punish only those that fall within the convex hull of assigned points. |
| Assignment Type | Whether to use a euclidean or sigmoidal assignment cost function. If True, use a the euclidean distance as the assignment cost. If False, use the sigmoidal scoring function. |
| Reorientation Boolean | Whether to apply ICP reorientation during the calculation of the IoD. If a numerical value, the ICP is applied until 20 iterations have elapsed or the deviation is less than the value supplied, whichever comes first. If False, no reorientation is applied. |

Table 2. Summary of classification results for each study area. Classification quality was assessed using Cohen's kappa coefficient (κ) and accuracy.  Differentiation of natural and planted trees using only the IoD displays moderate to high classification agreement, indicating that the IoD alone is sufficient to differentiate planted and naturally occurring tree stands. Differentiation of building types shows weaker agreement. For this purpose, the IoD alone may not be sufficient for acceptable classification quality but could be used to enrich more comprehensive classification methods.



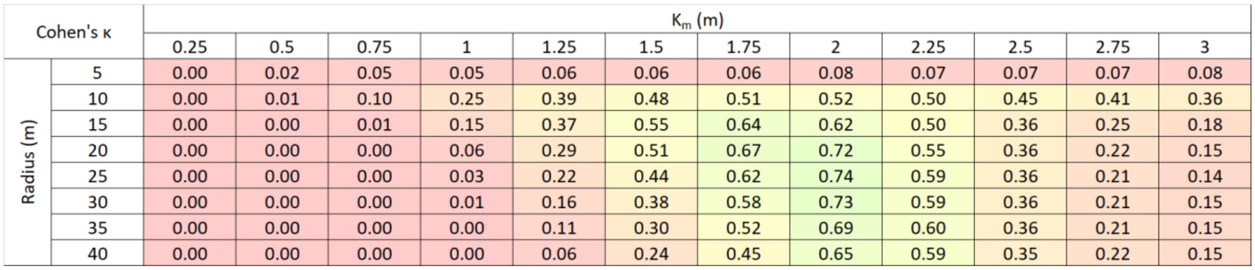
| Study Area | Disordered Group | Ordered Group | Peak κ | Corresponding Accuracy |
|---|---|---|---|---|
| Orchard in Crab Orchard, TN | Natural forest | Orchard | 0.81 | 0.96 |
| Reforested are in Mooresville, NC | Natural forest | Reforested zone | 0.74 | 0.87 |
| Neighborhood in Nashville, TN | Auxillary structures (sheds, detached garages) | Major buildings (homes and commerical buildings) | 0.44 | 0.76 |

Table 3. Sensitivity evaluation of the IoD for Site 1 using a threshold of 0.8 and varying neighborhood radius (r) and threshold of sigmoidal function (Km).



| Cohen's κ | | $K_m$ (m) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 |
| Radius (m) | 30 | 0.05 | 0.38 | 0.31 | 0.19 | 0.12 | 0.09 | 0.07 | 0.06 | 0.05 | 0.05 | 0.04 |
| | 40 | 0.02 | 0.42 | 0.47 | 0.19 | 0.10 | 0.06 | 0.05 | 0.04 | 0.03 | 0.03 | 0.03 |
| | 50 | 0.00 | 0.29 | 0.65 | 0.21 | 0.07 | 0.04 | 0.03 | 0.03 | 0.02 | 0.02 | 0.02 |
| | 60 | 0.00 | 0.24 | 0.78 | 0.26 | 0.08 | 0.04 | 0.03 | 0.02 | 0.02 | 0.02 | 0.01 |
| | 70 | 0.00 | 0.06 | 0.81 | 0.31 | 0.09 | 0.05 | 0.03 | 0.02 | 0.02 | 0.02 | 0.01 |
| | 80 | 0.00 | 0.02 | 0.81 | 0.35 | 0.09 | 0.05 | 0.03 | 0.03 | 0.02 | 0.01 | 0.01 |
| | 90 | 0.00 | 0.00 | 0.78 | 0.39 | 0.11 | 0.06 | 0.03 | 0.02 | 0.02 | 0.01 | 0.01 |
| | 100 | 0.00 | 0.00 | 0.73 | 0.43 | 0.11 | 0.06 | 0.04 | 0.02 | 0.02 | 0.01 | 0.01 |
| | 110 | 0.00 | 0.00 | 0.65 | 0.44 | 0.12 | 0.07 | 0.04 | 0.02 | 0.02 | 0.01 | 0.01 |
| | 120 | 0.00 | 0.00 | 0.57 | 0.45 | 0.12 | 0.07 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 |
| | 130 | 0.00 | 0.00 | 0.51 | 0.47 | 0.13 | 0.07 | 0.04 | 0.03 | 0.02 | 0.01 | 0.01 |
| | 140 | 0.00 | 0.00 | 0.43 | 0.44 | 0.13 | 0.07 | 0.05 | 0.03 | 0.02 | 0.01 | 0.01 |
| | 150 | 0.00 | 0.00 | 0.33 | 0.45 | 0.14 | 0.07 | 0.05 | 0.03 | 0.02 | 0.01 | 0.01 |

Table 4. Sensitivity test for Site 2 with an IoD threshold of 0.75.



| Cohen's κ | | $K_m$ (m) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0.25 | 0.5 | 0.75 | 1 | 1.25 | 1.5 | 1.75 | 2 | 2.25 | 2.5 | 2.75 | 3 |
| Radius (m) | 5 | 0.00 | 0.02 | 0.05 | 0.05 | 0.06 | 0.06 | 0.06 | 0.08 | 0.07 | 0.07 | 0.07 | 0.08 |
| | 10 | 0.00 | 0.01 | 0.10 | 0.25 | 0.39 | 0.48 | 0.51 | 0.52 | 0.50 | 0.45 | 0.41 | 0.36 |
| | 15 | 0.00 | 0.00 | 0.01 | 0.15 | 0.37 | 0.55 | 0.64 | 0.62 | 0.50 | 0.36 | 0.25 | 0.18 |
| | 20 | 0.00 | 0.00 | 0.00 | 0.06 | 0.29 | 0.51 | 0.67 | 0.72 | 0.55 | 0.36 | 0.22 | 0.15 |
| | 25 | 0.00 | 0.00 | 0.00 | 0.03 | 0.22 | 0.44 | 0.62 | 0.74 | 0.59 | 0.36 | 0.21 | 0.14 |
| | 30 | 0.00 | 0.00 | 0.00 | 0.01 | 0.16 | 0.38 | 0.58 | 0.73 | 0.59 | 0.36 | 0.21 | 0.15 |
| | 35 | 0.00 | 0.00 | 0.00 | 0.00 | 0.11 | 0.30 | 0.52 | 0.69 | 0.60 | 0.36 | 0.21 | 0.15 |
| | 40 | 0.00 | 0.00 | 0.00 | 0.00 | 0.06 | 0.24 | 0.45 | 0.65 | 0.59 | 0.35 | 0.22 | 0.15 |

Table 5. Sensitivity test for Site 3 (building centroids in the Lockeland Springs neighborhood in Nashville, TN) with an IoD threshold of 0.70.

| Cohen's κ | | $K_m$ (m) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 4.5 | 5 | 5.5 | 6 | 6.5 | 7 | 7.5 | 8 | 8.5 | 9 | 9.5 | 10 |
| Radius (m) | 15 | 0.10 | 0.09 | 0.08 | 0.08 | 0.09 | 0.09 | 0.07 | 0.05 | 0.06 | 0.04 | 0.03 | 0.02 |
| | 17 | 0.31 | 0.33 | 0.34 | 0.37 | 0.37 | 0.37 | 0.37 | 0.36 | 0.35 | 0.34 | 0.33 | 0.31 |
| | 19 | 0.34 | 0.37 | 0.40 | 0.43 | 0.44 | 0.42 | 0.40 | 0.39 | 0.38 | 0.38 | 0.37 | 0.37 |
| | 21 | 0.33 | 0.36 | 0.39 | 0.41 | 0.43 | 0.41 | 0.39 | 0.39 | 0.38 | 0.37 | 0.37 | 0.36 |
| | 23 | 0.32 | 0.33 | 0.35 | 0.37 | 0.37 | 0.37 | 0.35 | 0.33 | 0.32 | 0.30 | 0.29 | 0.28 |
| | 25 | 0.28 | 0.31 | 0.33 | 0.34 | 0.34 | 0.33 | 0.32 | 0.30 | 0.27 | 0.25 | 0.23 | 0.20 |
| | 27 | 0.25 | 0.28 | 0.30 | 0.31 | 0.31 | 0.29 | 0.28 | 0.24 | 0.20 | 0.17 | 0.15 | 0.13 |
| | 29 | 0.24 | 0.26 | 0.28 | 0.28 | 0.27 | 0.25 | 0.24 | 0.21 | 0.19 | 0.17 | 0.14 | 0.12 |
| | 31 | 0.27 | 0.28 | 0.31 | 0.31 | 0.31 | 0.30 | 0.25 | 0.23 | 0.20 | 0.19 | 0.15 | 0.12 |

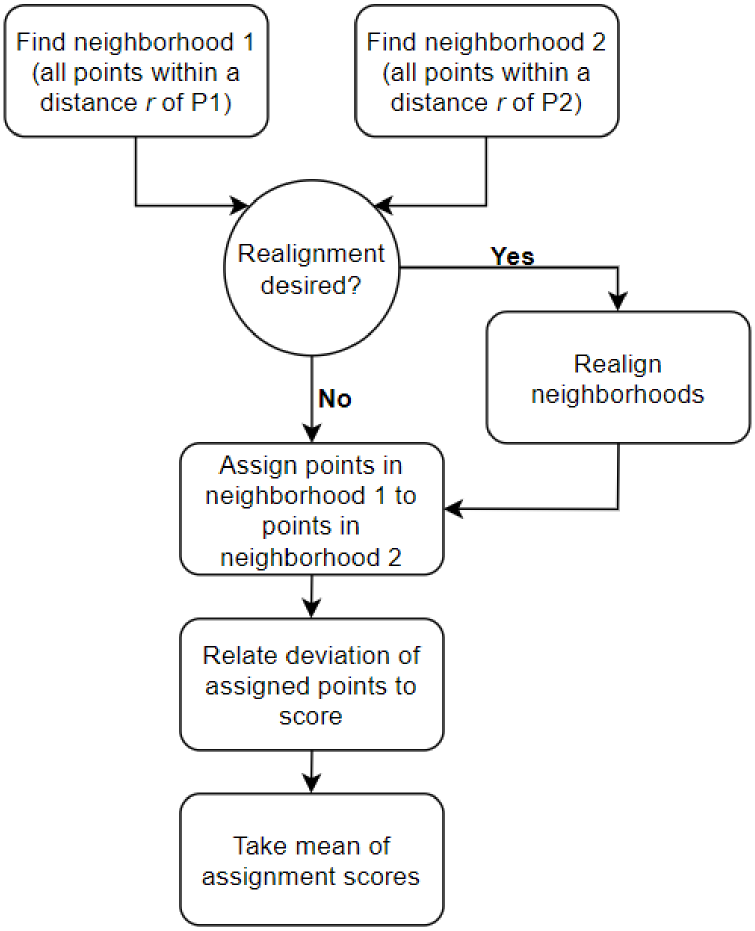
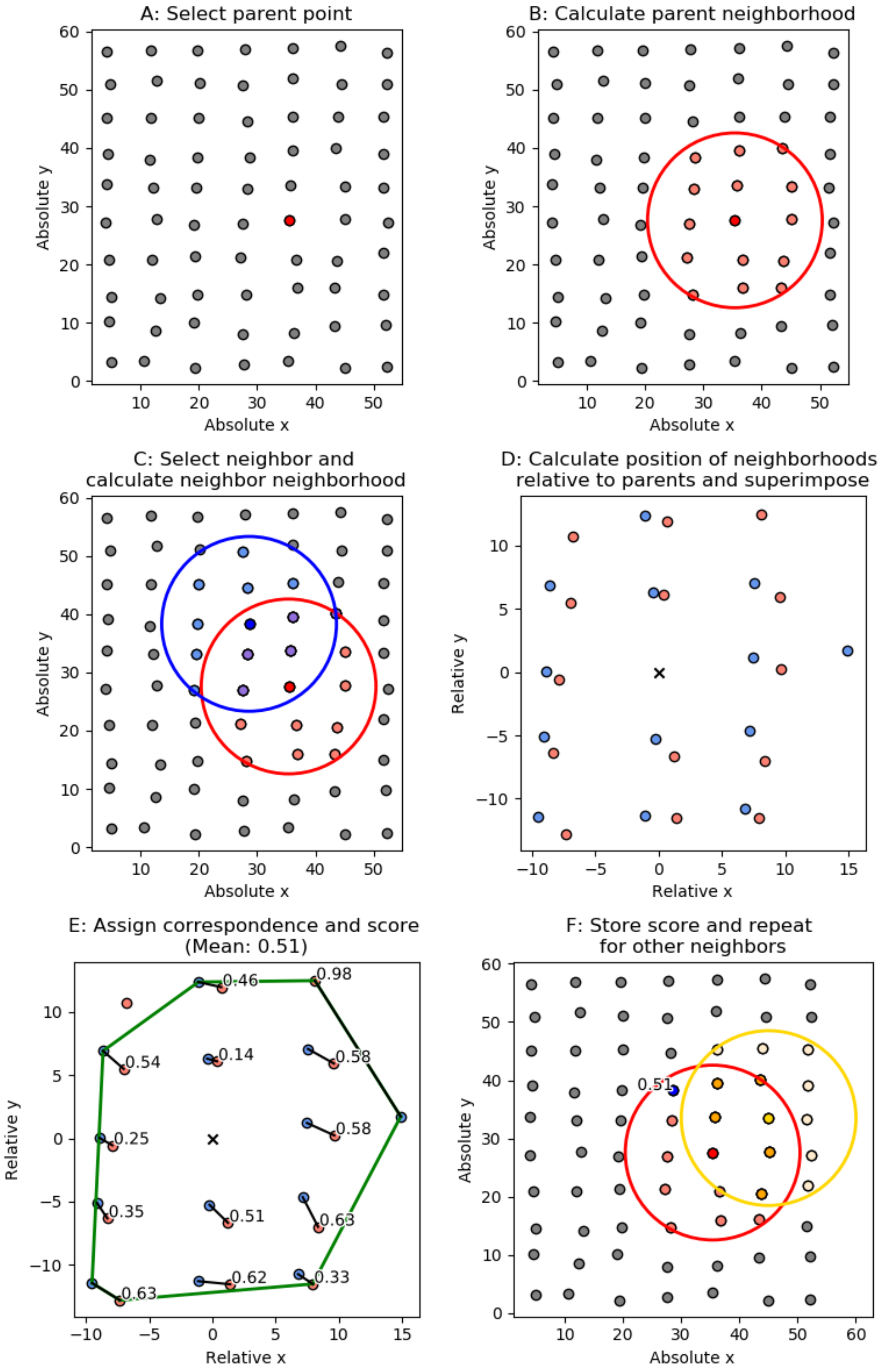

Find neighborhood 1 (all points within a distance *r* of P1)
Find neighborhood 2 (all points within a distance *r* of P2)
Realignment desired?
Yes
Realign neighborhoods
No
Assign points in neighborhood 1 to points in neighborhood 2
Relate deviation of assigned points to score
Take mean of assignment scores

Figure 1. Generalized process for calculating the IoD sub-score of two points $P_1$, $P_2$. Due to the heuristic nature of the algorithm, point set realignment, point pair assignment, and deviation scoring techniques can vary between implementations of the IoD.



A: Select parent point

B: Calculate parent neighborhood

C: Select neighbor and calculate neighbor neighborhood

D: Calculate position of neighborhoods relative to parents and superimpose

E: Assign correspondence and score (Mean: 0.51)

F: Store score and repeat for other neighbors

Figure 2. Illustration of the calculation of the IoD sub-score between two points. Given a parent point (A), its neighbors are selected (B). Similarly, given a child point its neighbors are selected (C). Both parent and child neighborhoods are converted from absolute to relative coordinate system (D). Each point in the parent set is assigned to a point in the child set (E). The process then repeated by selecting another child point set (F). The IoD sub-score between two neighborhoods is the mean of the assignment scores between a parent point and every neighbor.

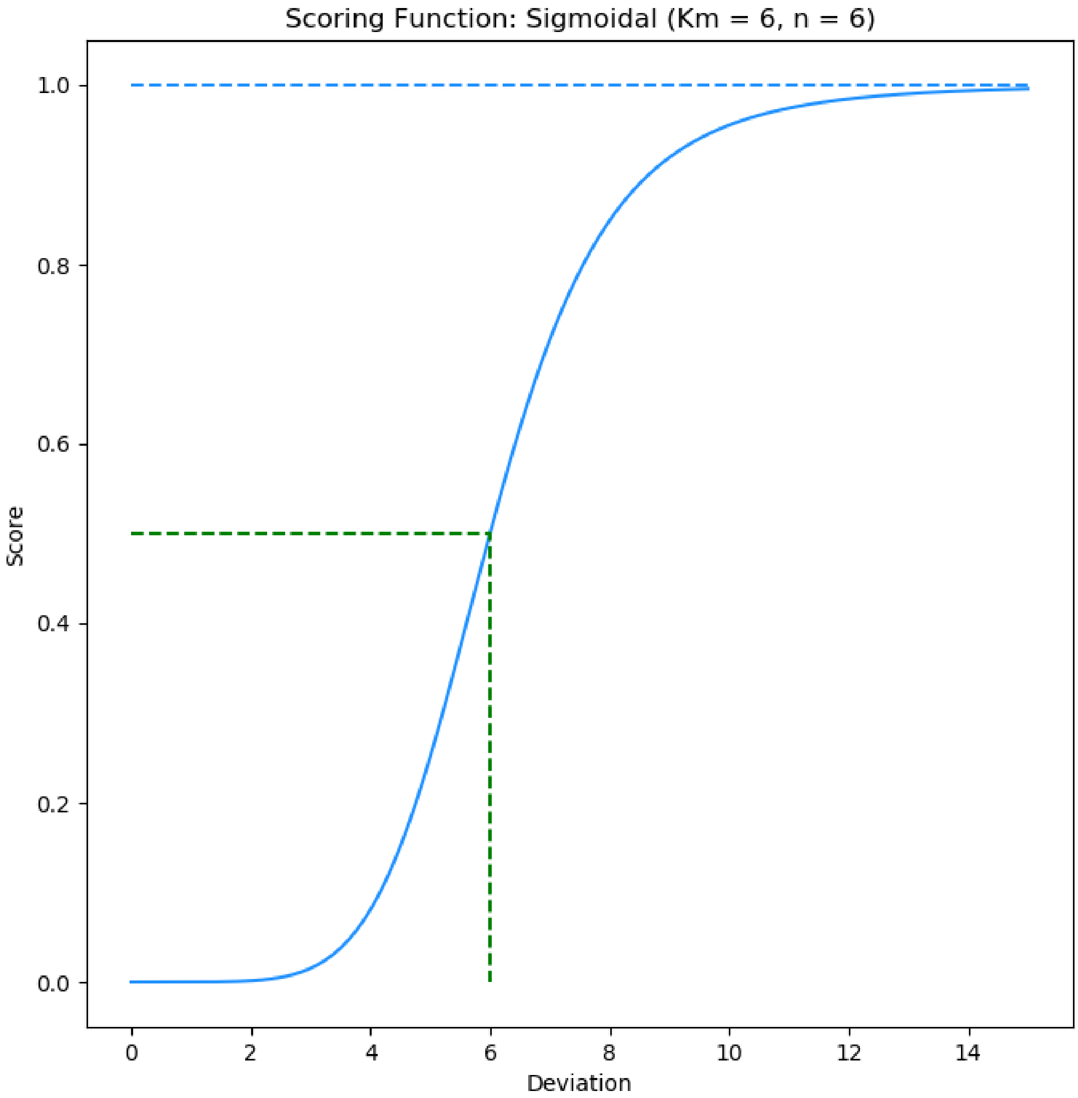


Figure 3. An example of a sigmoidal scoring function, $s = \frac{d^n}{{K_m}^n + d^n}$, where d is the point-pair deviation, n is the function cooperativity and K$_m$ is the deviation midpoint. An n of 1 will create a hyperbolic function, while values greater than 1 will make the function increasingly sigmoidal. The value of $K_m$ is the value of d at which the score is 0.5; this is represented by the dashed green lines.

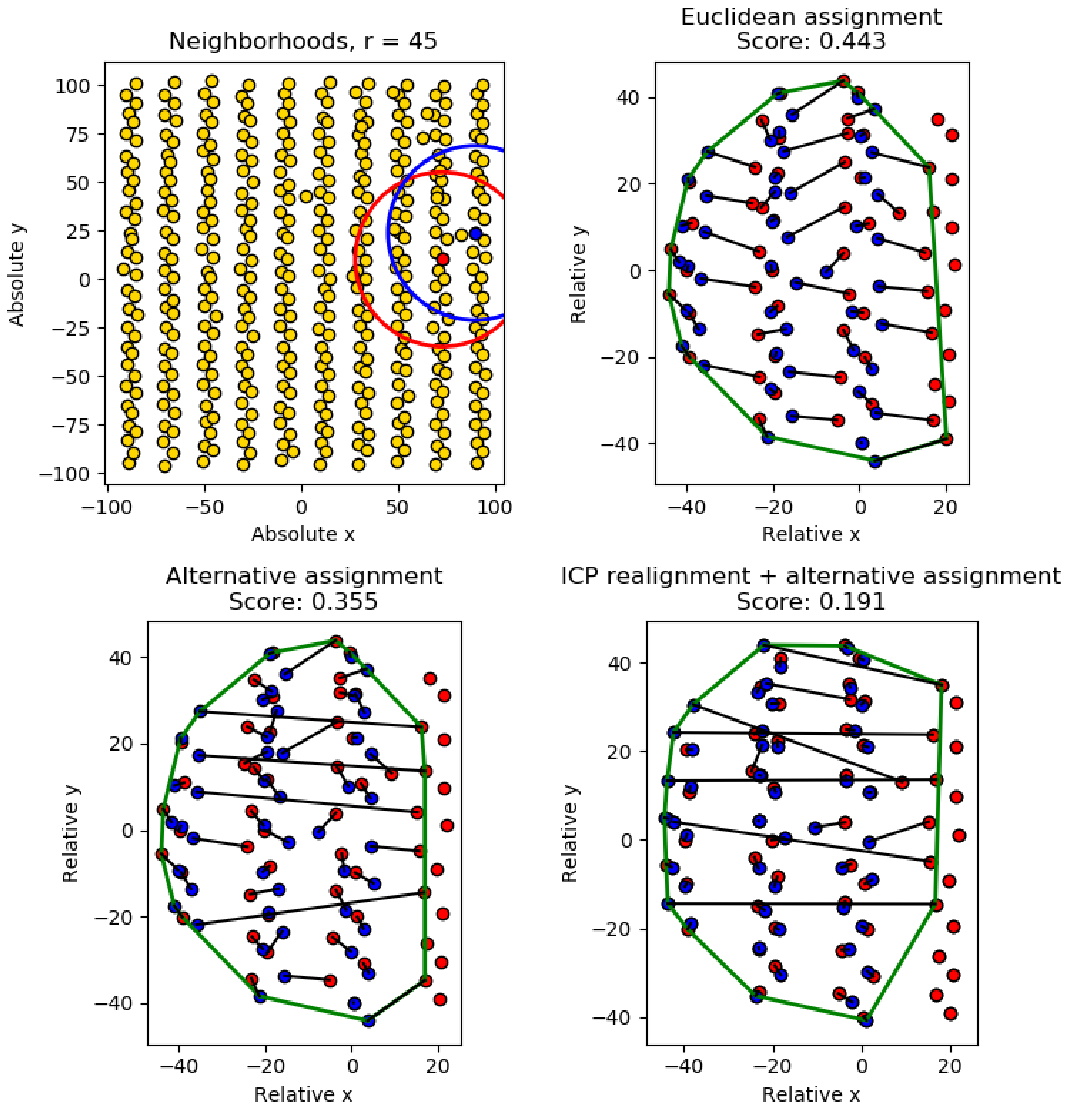


Figure 4. Visualization of the effects of using different cost functions on point assignment and IoD sub-score. The neighborhood radius is shown for both the red (parent) and blue (child) points in the upper left figure. The remaining three figures explore various point assignment methods and the calculated IoD.

5

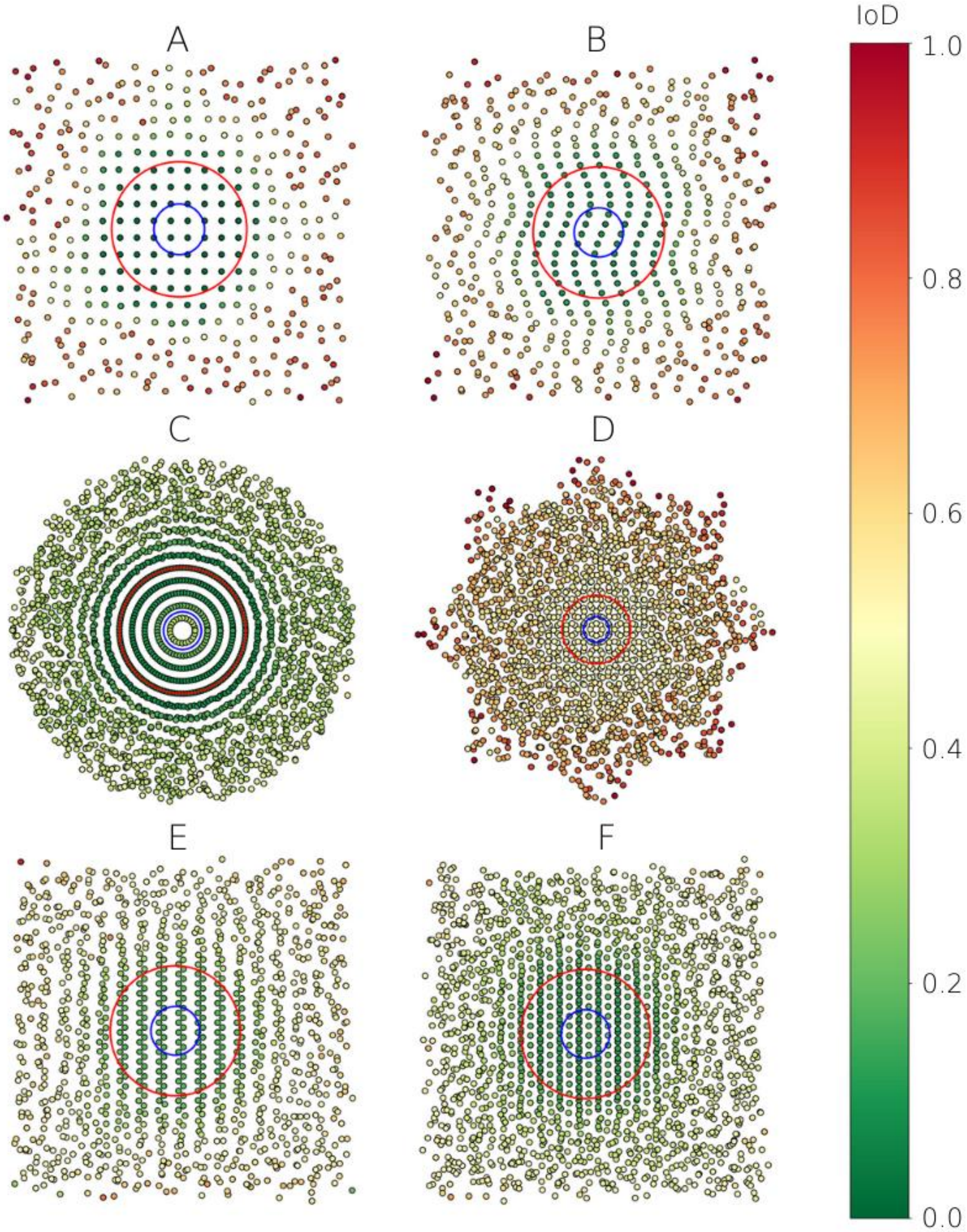


Figure 5. Results of theoretical evaluation of the IoD. The pattern is perturbed with random noise that increases in strength as a function of distance beyond the red circle, and the blue circle represents the neighborhood size. Multiple patterns were investigated: square grid (A), rectangular grid modified with a sinusoidal function (B), pattern 5 formed by overlaying concentric circles with equal linear point densities (C), pattern formed by overlapping a square grid with a copy of the same grid rotated 45 degrees (D), pattern formed by overlapping two offset rectangular grids, pattern formed by overlapping three offset rectangular grids (F). No reorientation was applied.

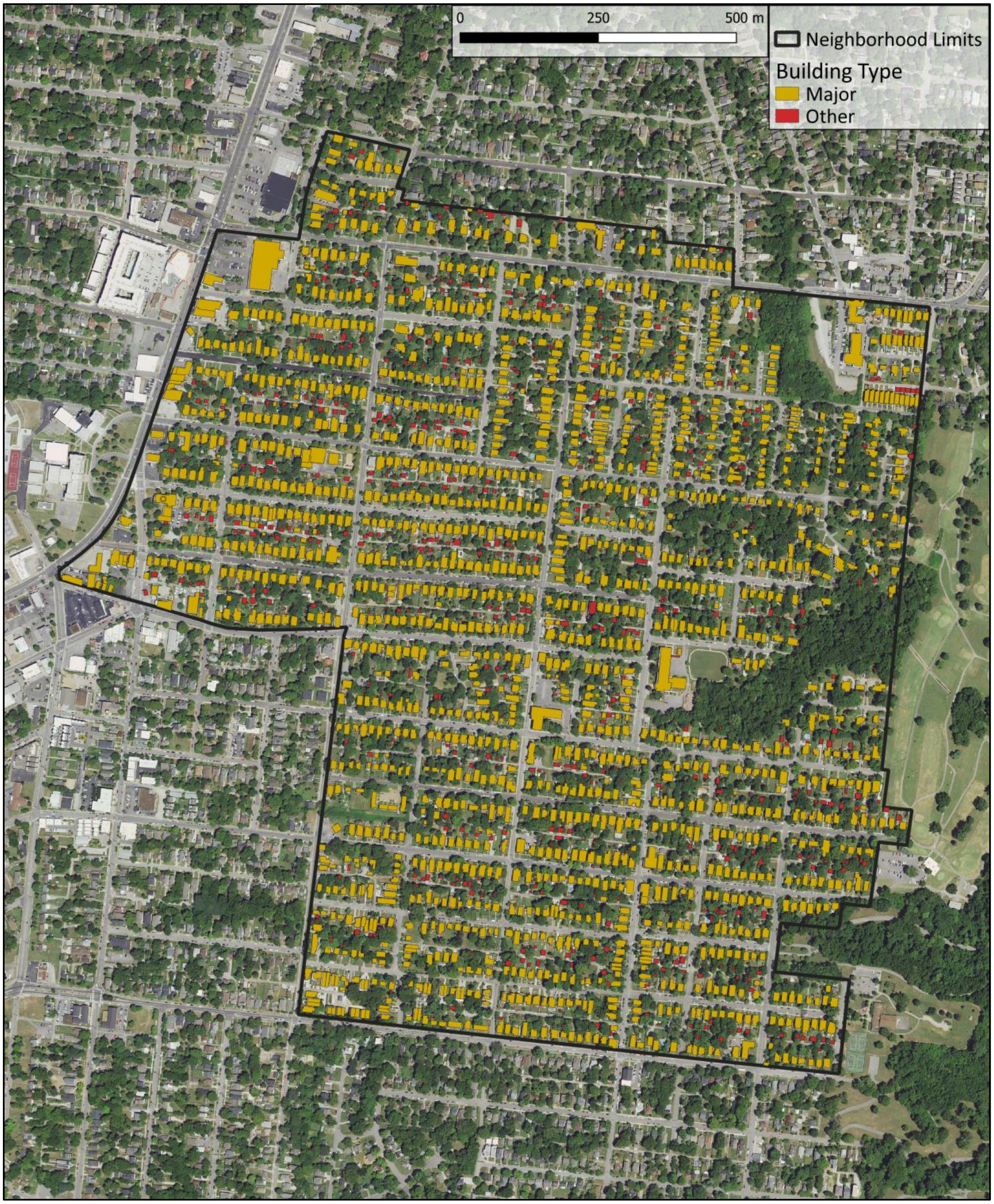
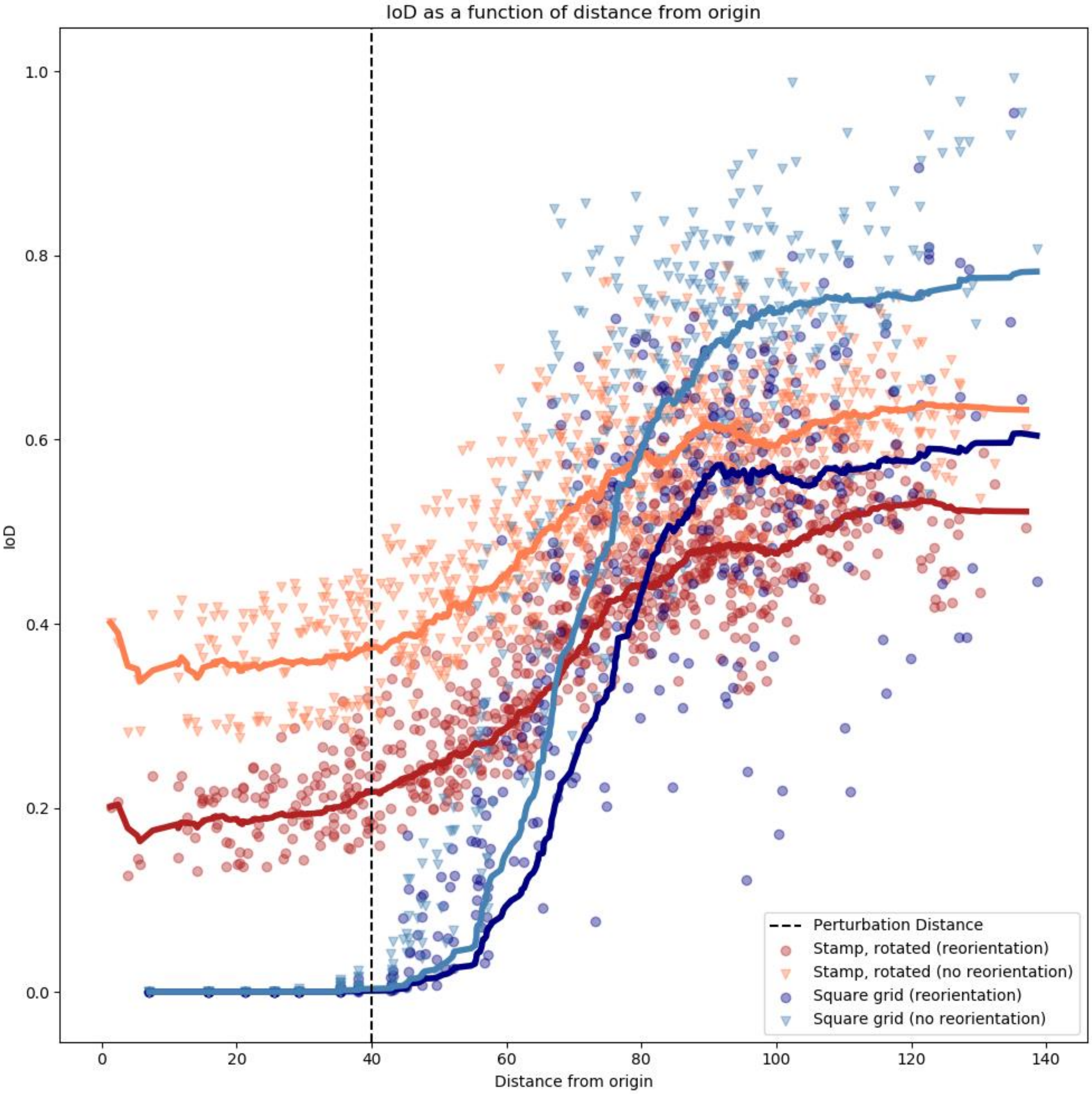


Figure 6. A map of building types in a neighborhood in Nashville, TN.



5 Figure 7. Illustration of index of disorder in theoretical evaluation with increasing noise beyond the perturbation distance. Some types of patterns, such as repeated stamps, show slightly increased discrimination between ordered and disordered points when reorientation is applied. More homogenous patterns, such as a square grid, show reduced discrimination when reorientation is applied.

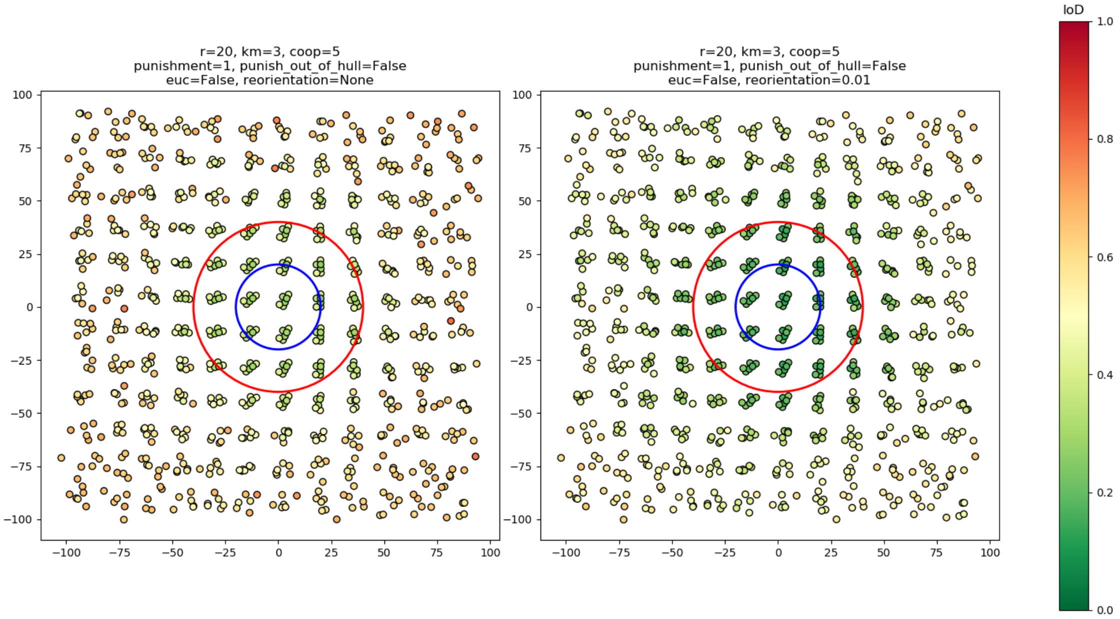


Figure 8. Results from applying two versions of the IoD algorithm without and with realignment (left and right respectively) to a pattern that consists of a repeating wallpaper group (a “stamp”) that is continuously rotated. The pattern is perturbed with random noise that increases in strength as a function of distance beyond the red circle, and the blue circle represents the neighborhood size.

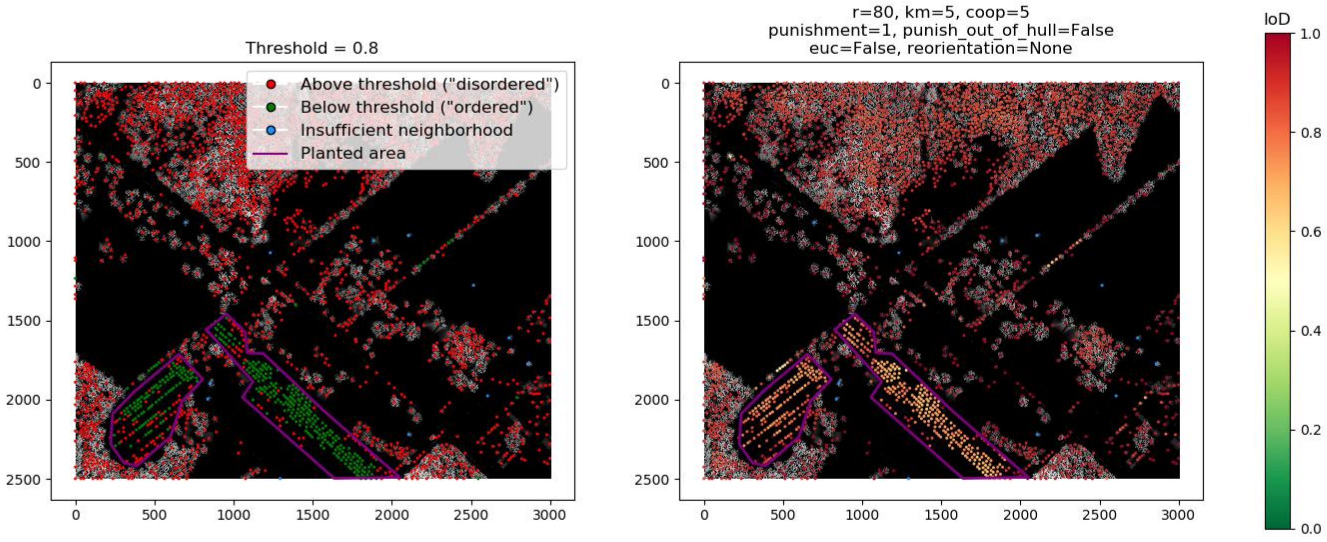

Threshold = 0.8
Above threshold ("disordered")
Below threshold ("ordered")
Insufficient neighborhood
Planted area
r=80, km=5, coop=5
punishment=1, punish_out_of_hull=False
euc=False, reorientation=None
0
500
1000
1500
2000
2500
3000
IoD
1.0
0.8
0.6
0.4
0.2
0.0

Figure 9. Results from applying the IoD to trees extracted from a DHM of an orchard near Crab Orchard, TN. Each point represents a tree crown. Axis units are meters. Trees with an IoD value above the threshold are classified as “disordered”, interpreted to be non-orchard trees. Trees with an IoD value below the threshold are classified as “ordered”, interpreted to be orchard trees.

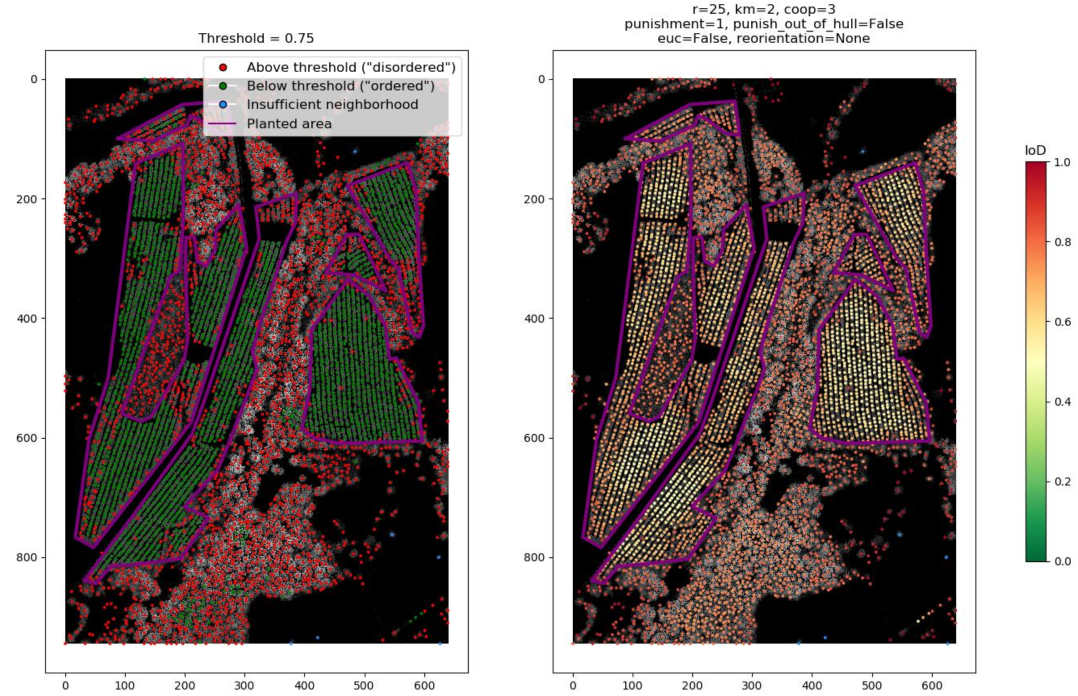


Figure 10. Results from applying the IoD to trees extracted from a DHM of a mixed planted and natural forest stand near Mooresville, NC. Each point represents a tree crown. Axis units are in meters. Trees with an IoD value above the threshold are classified as “disordered”, interpreted to be naturally occurring trees. Trees with an IoD value below the threshold are classified as “ordered”, interpreted to be planted trees.

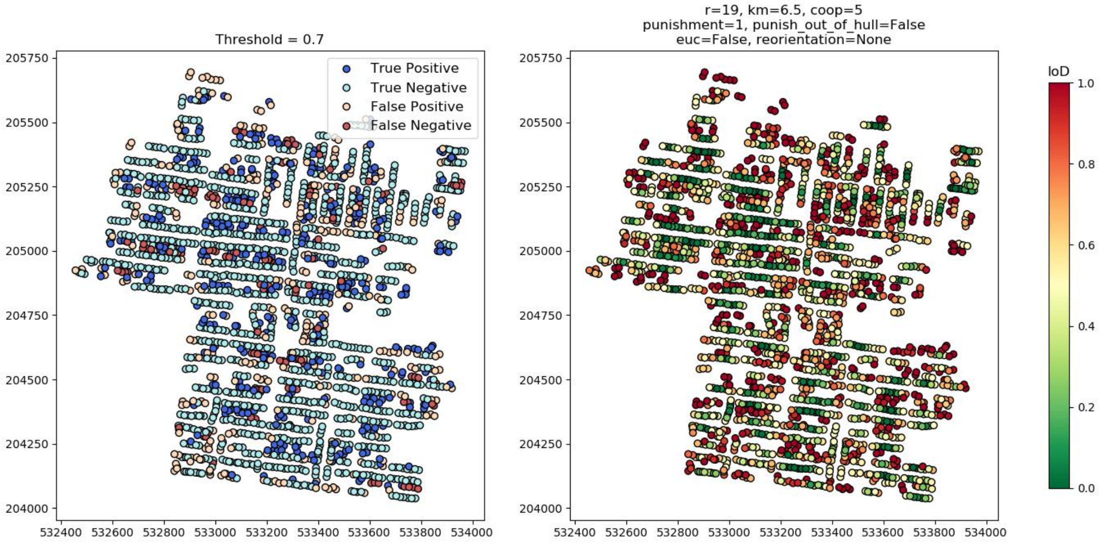


Figure 11. Building centroids in Nashville, TN. On the right, the IoD is shown. On the left, classification results are shown for a threshold of 0.7. Buildings with an IoD below the threshold are classified as “major buildings”, such as houses (a negative response) and those above the threshold are classified as “other”, which is comprised of auxiliary

buildings such as sheds and detached garages (a positive response). Axis units are in meters and correspond to the TN State Plane coordinate system.

## 7 Acknowledgements

We thank Dr. Racha El Kadiri for her input and guidance while writing this manuscript.

## 8 Funding

This work was supported in part by the Natural Resources Conservation Service [grant numbers xxxx, yyyy] and a URECA grant through Middle Tennessee State University.

## 9 References

Aksoy, S., Yalniz, I. Z., & Tasdemir, K. (2012). Automatic Detection and Segmentation of Orchards Using Very High Resolution Imagery. *IEEE Transactions on Geoscience and Remote Sensing, 50*(8), 3117-3131.

Antuono, M., Bouscasse, B., Colagrossi, A., & Marrone, S. (2014). A measure of spatial disorder in particle methods. *Computer Physics Communications, 185*(10).

Baena, S., Moat, J., Whaley, O., & Boyd, D. S. (2017). Identifying species from the air: UAVs and the very high resolution challenge for plant conservation. *PLoS One, 12*(11).

Besl, P., & McKay, N. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(2), 239–256.

Boeing, G. (2019). Urban Spatial Order: Street Network Orientation, Configuration, and Entropy. *Applied Network Science*.

Buttenfield, B. P. (1989). Scale-Dependence And Self-Similarity In Cartographic Lines. *Cartographica: The International Journal for Geographic Information and Geovisualization, 26*(1), 79-100.

Chang, T., & Kuo, C.-C. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing, 2*(4), 429-441.

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement. *Educational and Psychological Measurement, 20*(1), 37-46.

Dong, P. (2000). Lacunarity for Spatial Heterogeneity Measurement in GIS. *Annals of GIS, 6*(1), 20-26.

Getis, A. (n.d.). A History of the Concept of Spatial Autocorrelation: A Geographers Perspective. *Geographical Analysis, 40*(3), 297-309.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3*(6), 610 - 621.

Herold, M., Couclelis, H., & Clarke, K. C. (2005). The role of spatial metrics in the analysis and modeling of urban land use change. *Computer, Environment and Urban Systems, 29*(4), 369-399.

Huang, X., Liu, H., & Zhang, L. (2015). Spatiotemporal Detection and Analysis of Urban Villages in Mega City Regions of China Using High-Resolution Remotely Sensed Imagery. *IEEE Transactions on Geoscience and Remote Sensing, 53*(7), 3639-3657.

Isenburg, M. (2019). LAStools - Efficient Tools for LiDAR Processing. rapidlasso GmbH.

Jiang, B., & Yin, J. (2013). Ht-Index for Quantifying the Fractal or Scaling Structure of Geographic Features. *Annals of the Association of American Geographers, 104*(3), 530-504.

Khan, M. M., Al-Yahyai, R., & Al-Said, F. (2017). *The Lime: Botany, Production and Uses.* CABI.

Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 83-97.

Liu, Y., Collins, R., & Tsin, Y. (2004). A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(3), 354-371.

Lou, S., Wang, C., Li, X., Zeng, H., Li, D., Xia, S., & Wang, P. (2016). Fusion of Airborne Discrete-Return LiDAR and Hyperspectral Data for Land Cover Classification. *Remote Sensing, 8*(1).

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing, 1*(4), 823-870.

Momm, H., Easson, G., & Kuszmaul, J. (2009). Evaluation of the use of spectral and textural information by an evolutionary algorithm for multi-spectral imagery classification. *Computers, Environment and Urban Systems, 33*(6), 463-471.

Nilsson, L., & Gil, J. (2019). *The Mathematics of Urban Morphology.* Springer.

Oliphant, T. E. (2006). *A guide to NumPy.* Trelgol Publishing.

Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H., & Guibas, L. (2008). Discovering structural regularity in 3D geometry. *ACM Transactions on Graphics, 27*(3).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Phiri, D., & Morgenroth, J. (2017). Developments in Landsat Land Cover Classification Methods: A Review. *Remote Sensing, 9*(9).

Remmel, T. K., & Csillag, F. (2003). When are two landscape pattern indices significantly different? *Journal of Geographical Systems, 5*, 331-351.

Rossum, G. v. (1995). Technical Report CS-R9526, Python tutorial. Amsterdam: Centrum voor Wiskunde en Informatica (CWI).

Shekhar, S., Evans, M., Kang, J., & Mohan, P. (2011). Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1*(3), 193-214.

Swain, P. H., & Stephen B. Vardman, J. C. (1981). Contextual classification of multispectral image data. *Pattern Recognition, 13*(6), 429-441.

Trias-Sanz, R. (2006). Texture Orientation and Period Estimator for Discriminating Between Forests, Orchards, Vineyards, and Tilled Fields. *IEEE Transactions on Geoscience and Remote Sensing, 44*(1020).

Virtanen, P., Gommers, R., Oliphant, T. E., & Haberland, M. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*.

Williams, E., & Wentz, E. (2008). Pattern Analysis Based on Type, Orientation, Size, and Shape. *Geographical Analysis, 40*(2), 97-122.

Zhen, Z., Quackenbush, L. J., & Zhang, L. (2016). Trends in Automatic Individual Tree Crown Detection and Delineation—Evolution of LiDAR Data. *Remote Sensing, 8*(4), 333.