The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-4/W1-2022
Free and Open Source Software for Geospatial (FOSS4G) 2022 – Academic Track, 22–28 August 2022, Florence, Italy

A METHOD FOR UNIVERSAL SUPERCELLS-BASED REGIONALIZATION (PRELIMINARY RESULTS)

Jakub Nowosad[1]*, Tomasz F. Stepinski[2], Mateusz Iwicki[1]

[1] Institute of Geoecology and Geoinformation, Adam Mickiewicz University,
Poznan, Poland - nowosad.jakub@gmail.com, matiwi1@st.amu.edu.pl
[2] Space Informatics Lab, Department of Geography and GIS, University of Cincinnati, Cincinnati, OH, USA - stepintz@ucmail.uc.edu

Commission IV, WG IV/4

KEY WORDS: Spatial Patterns, Knowledge Discovery, Regionalization, Supercells, Over-segmentation

ABSTRACT:

Geospatial data comes in various forms, including multi and hyperspectral images but also rasters of local composition, local time series, local patterns, etc. Thus, we generalize the SLIC algorithm to work with a library of different data distance measures that are pertinent to geospatial rasters. This contribution includes a description of the generalized SLIC algorithm and a demonstration of its application to the regionalization of the raster of local compositions (of land cover classes). Two workflows were tested, both starting with SLIC preprocessing. In the first, superpixels are subject to regionalization using the graph-partitioning algorithm. In the second, superpixels are first clustered using the K-means algorithm, followed by regions delineation using the connected components labeling. These two workflows are compared visually and quantitatively. Based on these comparisons, coupling of superpixels with a graph-partitioning algorithm is the preferred choice. Finally, we propose using the SLIC superpixel preprocessing algorithm for the task of regionalization of various geospatial data in the same way as it is used for the task of image segmentation in computer vision.

1. INTRODUCTION

Generalization is one of the fundamentals of scientific research. In the context of spatial information, generalization needs to allow for finding common properties but also for spatial contiguity. Therefore, such generalization is often made through regionalization - partitioning of space into spatial clusters or regions. This process is vital for environmental studies, such as geography, ecology, biology, and landscape analyses, where many patterns and processes are autocorrelated spatially. Examples of regionalizations include delineation of ecoregions, detection of homogeneous zones for precision agriculture, the definition of climate regions, and so on.

Traditionally spatial generalization was performed manually (Bailey et al., 1985), often based on a compilation of pre-existing, independently conducted studies (e.g., Omernik (1987); Olson and Dinerstein (2002)). This approach lacks a quantitative framework, and thus no systematic checks, modifications or objective updates are possible. Currently, the abundance of remote sensing spatial data, such as satellite imagery, gridded climate data, or land cover maps, allows for fast extraction of relevant spatial information on regional and global scales, making possible studies rooted in a clear quantitative framework.

Such data, however, still requires spatially-aware generalization to formulate general concepts or claims. Remote sensing data stores information as a set of raster cells, where a single cell is unaware of its spatial context. This is often not enough to understand underlying objects or processes. (Geographic) object-based image analysis (OBIA) (Blaschke, 2010) is frequently applied to resolve this issue. It is an approach to partition space consisting of raster cells into homogeneous objects and thus making spatial regionalization possible. Several generalization techniques were developed for OBIA, including a superpixels approach that proved to perform best for image processing and remote sensing data analysis (Csillik, 2017).

The main idea of superpixels is to create connected groupings of cells with similar values (Ren and Malik, 2003; Achanta et al., 2012). Each superpixel represents a desired level of homogeneity while at the same time maintaining spatial structures. Superpixels also carry more information than each cell alone, and thus they can speed up the subsequent processing efforts (Ren and Malik, 2003; Achanta et al., 2012).

The Simple Linear Iterative Clustering (SLIC) superpixels algorithm (Achanta et al., 2012) proved to perform well for image processing (Stutz et al., 2018) and remote sensing data analysis (Subudhi et al., 2021). However, recall that SLIC is only a preprocessing algorithm and has to be coupled to a clustering/segmentation algorithm for the ultimate goal of regionalization. Also, SLIC uses the Euclidean metric to calculate the data component of the distance between cells. This is not adequate for multi-dimensional data (Aggarwal et al., 2001), as well as for non-vector data. Non-vector data are common in geospatial applications; for example, histograms represent the local composition data (Buchhorn et al., 2020), time-series represent climate data (Netzel and Stepinski, 2016), and co-occurrence matrices represent pattern data (Jasiewicz et al., 2018).

The results presented during the GIScience 2021 conference (Nowosad and Stepinski, 2021) provide a basis for addressing the Euclidean metric issue. The proposed extension to the SLIC algorithm has a library of different metrics to calculate the data component of the distance between cells. A user selects a metric most appropriate for the type of data at hand. This extension is already available as open-source software in the form of an R package `supercells`.

* Corresponding author

The issue of regionalization – how to best couple superpixels to a clustering/segmentation algorithm in a geospatial context – remains open. This contribution aims to present the work in progress related to developing a robust workflow for the regionalization of geospatial data utilizing supercells in the preprocessing step. Our focus is on the universality of the provided solution, i.e., to develop a workflow that works well with different types of geospatial data (multi and hyperspectral images, compositions, time series, patterns, etc.). To start, here we test two regionalization workflows of compositional data. Both workflows use SLIC preprocessing, but one clusters superpixels, whereas the other performs graph-based segmentation on superpixels. We tested the two workflows for feasibility, quality, and visual coherence.

2. METHODS

2.1 Extended SLIC

In the rest of this paper, we will use the terms cell and supercell instead of pixel and superpixel to underscore that we do not necessarily work with image data. The Simple Linear Iterative Clustering (SLIC, Achanta et al. (2012)) starts with regularly located cluster centers spaced by the interval of S. By construction, these initial cluster centers coincide with underlying cells and inherit from their positions and values. To avoid being on an abnormal cell, the initial centers' locations are perturbed in a 3×3 neighborhood to the lowest gradient position – a cell the least different from its neighbors. Next, the distance D between a cluster center and every cell in its $2S \times 2S$ region is calculated.

$$D = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \tag{1}$$

where d_c is the distance between cell objects, m is the compactness parameter, d_s is the spatial (Euclidean) distance between the cells, and S is the interval between the initial cluster centers. Since the original SLIC was designed for natural images, d_c is often referred to as the color (spectral) distance, but we are using a more general description of d_c as a data distance between objects carried by cells.

The spatial (Euclidean) distance between cells represents their spatial proximity:

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2}$$

A metric used to calculate a distance between two data objects, d_c, depends on the form of these objects. For example, if objects are B-dimensional vectors, the metric may be the B-dimensional Euclidean distance.

The distance between B-dimensional value vectors $I(x_i, y_i, s_{i,p}, p = 1 \ldots, B)$ and $I(x_j, y_j, s_{j,p}), p = 1 \ldots, B)$ is:

$$d_c = \sqrt{\sum_{p \in B} (I(x_i, y_i, s_{i,p}) - I(x_j, y_j, s_{j,p}))^2} \tag{3}$$

but different metrics must be used with different data in different forms. The distance between objects controls the homogeneity of supercells, while the spatial distance is related to spatial contiguity.

Supercells are created by assigning each cell to the cluster center with the smallest overall distance. Afterward, cluster centers (centroids) are updated to values equal to the average of all the cells belonging to their respective clusters. Note that cluster centers do not necessarily coincide with any particular cell after such an update.

The SLIC algorithm works iteratively, repeating the above process until it reaches the expected number of iterations. The last, optional, step enforces the 4-connectivity of the cell belonging to the same supercells by reassigning disjoint cells. SLIC has two main parameters - S and m. The first one controls the initial interval between cluster centers of each supercell, which is directly related to the number and the size of output supercells. The m parameter, compactness, controls the relative influence of d_s versus d_c on the results. Its large values result in more regularly shaped (squarer) supercells, while lower values create more spatially adapted, irregularly shaped supercells. In other words, the m parameter oversees the balance between the data distance and spatial distance.

Extension of SLIC proposed by Nowosad and Stepinski (2021) allows for the use of any distance measure (not just the Euclidean distance) to calculate d_c. It also allows for any function (not just the arithmetic mean) to be used for averaging values of cluster centers, and specify custom initial custom of cluster centers.

For example, for the compositional data, where the state of a cell is given by a normalized histogram, any of possible distances between histograms (Cha, 2007) could be used as d_c. When testing workflows for regionalization of compositional data (section 3) we calculate d_c using the Jenson-Shannon divergence (Lin, 1991) as a "distance" between histograms,

$$d_c = H(\frac{A+B}{2}) - \frac{1}{2}[H(A) + H(B)] \tag{4}$$

where A and B are normalized histograms of two cells, and $H(A)$ and $H(B)$ indicates values of Shannon's entropy (Shannon, 1948) of these histograms (recall that normalized histograms are discrete probability distribution functions):

$$H(A) = -\sum_{p \in A} A_p \log_2 A_p \tag{5}$$

A_p is the pth value of the first of the compared histograms.

2.2 Clustering and regionalization methods

In this contribution, we tested two distinctive regionalization workflows. Each will start from performing the SLIC preprocessing, but one utilizes k-means clustering on supercells to achieve regionalization while the other performs graph-based segmentation.

K-means is a clustering method that partitions observations (values of raster cells or supercells, in our case) into k clusters. Each observation is assigned to a cluster with the nearest

cluster center (nearest mean) (Hartigan and Wong, 1979). The k-means method does not use information about spatial relationships between cells/supercells, and thus its result is a set of non-adjacent areas. Therefore, we needed to split each cluster into many regions consisting of linked cells using the connected component labeling method to create final regions. We refer to this regionalization workflow as KM- CCL (k-means followed by connected components labeling). Importantly, this approach does not allow for directly specifying the expected number of regions.

Graph-based segmentation starts from constructing a weighted graph with nodes located at supercells. The edges of the graph are weighted by the distance measured between linked supercells. We use the SKATER (Spatial 'K'luster Analysis by Tree Edge Removal) graph-based segmentation algorithm (AssunÇão et al., 2006). SKATER prunes the graph to its minimum spanning tree (MST) (Grygorash et al., 2006) and then iteratively partitions the graph by identifying edges whose removal increases the objective function (between-group dissimilarity) the most. The iterative process stops when a specified number of regions is obtained.

2.3 Quality assesment

The resulting regions, in theory, can be evaluated using internal and external validation techniques. External validation metrics are possible to calculate when the "ground truth" data is available that we can compare the results with. However, such data does not exist in this case, and thus internal validation can only be performed.

Two internal validation metrics used in this study are inhomogeneity and isolation. These measures are applied to both supercells and regions. Recall that supercells are small regions resulting from oversegmentation, they should be very homogeneous but not necessarily distinct from their neighbors. Regions (collections of supercells) should still be as homogeneous as possible and practical, but they will be less homogeneous than supercells. On the other hand, regions are expected to be distinct from their neighbors.

The inhomogeneity metric measures a degree of mutual dissimilarity between all cells in a supercell/region (Jasiewicz et al., 2018). It is derived by calculating an average distance between all cells in a supercell/region using a given distance metric (Jensen-Shannon distance, in our evaluation in section 3). The larger the value, the more inhomogeneous (worse) the supercells/regions are. Isolation is an average distance between the focus supercell/region and all of its neighbors (Haralick and Shapiro, 1985; Jasiewicz et al., 2018). For the regions, larger values of isolation are better; for supercells, values of isolation are less important. Both inhomogeneity and isolation have values between 0 and 1.

2.4 Software

Extended SLIC algorithm is implemented in the R programming language (R Core Team, 2021) as an open-source package called `supercells` (Nowosad and Stepinski, 2021). This package works on spatial data with one variable (e.g., continuous raster), many variables (e.g., RGB rasters or time-series), and spatial patterns (e.g., areas in categorical rasters). The calculations in the `supercells` package are customizable, providing about 50 built-in distances and similarity measures while

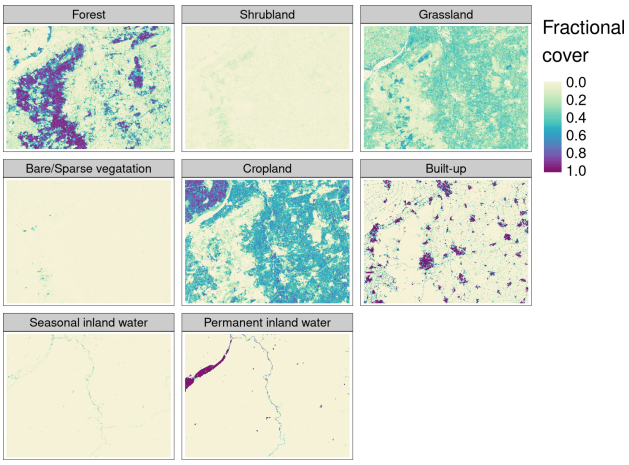

Figure 1. Fractional cover (0-1) of the 8 base land cover classes in the eastern Netherlands

also allowing any user-defined R function to be used as a distance measure. This extension also makes it possible to apply other averaging functions than the arithmetic mean when updating values of supercells' centers. It also has some experimental features, such as the possibility to provide user-defined initial cluster centers.

The k-means method was applied using the built-in R `kmeans()` function (R Core Team, 2021), while the `terra` and `sf` packages were used for connected-component labeling and general spatial data handling (Hijmans, 2021; Pebesma, 2018). For graph-based regionalization, the `skater()` function from the `rgeoda` package was used (Li and Anselin, 2022). Its current version allows providing a distance matrix, which enables regionalizations based on various dissimilarity measures.

Quality assesment was possible with the `regional` package (Nowosad, 2022). Its functions, `reg_inhomogeneity()` and `reg_isolation()`, take a spatial vector object containing regions and a raster object with the values of interest. Next, they compute values of inhomogeneity and isolation for each region based on a given distance measure and specified sample size.

Addtionally, visualizations in this paper were created using the `tmap` and `ggplot2` packages (Tennekes, 2018; Wickham, 2016).

2.5 Materials

The presented work is based on the Copernicus Global Land Service: Land Cover 100m data (Buchhorn et al., 2020) for the year 2019. Cover fractions (%) of the eight base classes (forest, shrubland, grassland, bare/sparse vegetation, cropland, built-up, seasonal inland water, and permanent inland water) were derived from `https://lcviewer.vito.be/` and cropped to an area of about 4200 km^2 located in the eastern Netherlands (Figure 1). The input raster file had 507 rows, 1105 columns, and eight layers.

3. WORKFLOWS COMPARISON

We tested two different regionalization workflows to delineate areas with similar land cover fractions. This is an example in which the input data is in the form of histograms (or discrete probability distributions). The composition of land cover types

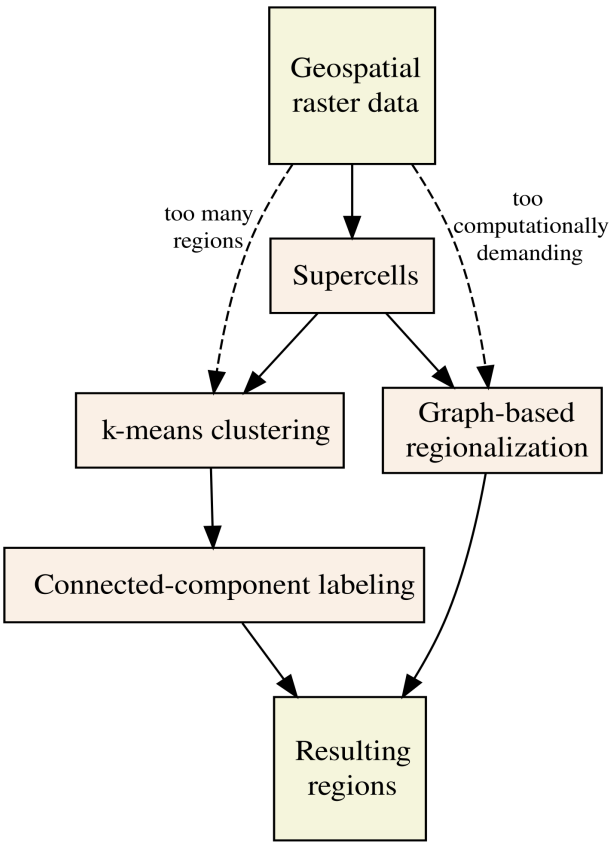

Figure 2. Possible approaches for regionalization of geospatial raster data

within a 100m×100m area is stored in each cell. Cells are similar if they have similar compositions of land cover types. The magnitude of this similarity is measured by $1 - d_c$, where d_c is given by the Jensen-Shannon dissimilarity (eq. 4).

In principle, regionalization of the compositional raster can be achieved with or without supercells preprocessing by using clustering or segmentation directly on cells. Thus, theoretically, we should consider four workflows.

1. The k-means clustering based on cells' values followed by the connected components labeling.
2. The SKATER regionalization of cells' values.
3. The k-means clustering based on supercells' values followed by the connected components labeling (KM-CCL).
4. The SKATER regionalization of supercells' values (SKATER).

Figure 2 summarizes the tested workflows and highlights some initial findings. The first workflow proved to be unsuitable for regionalization. Even small values of k produced a large number of regions, many of which were very small, resulting in a salt and paper map. For example, $k = 2$ returned 6,000 regions and $k = 3$ 10,000 regions. Thus, this workflow was insufficient for our purpose without any additional pre- or postprocessing of the results.

Similarly, the second workflow seemed to be infeasible for the stated problem. Most of the currently used regionalization methods, such as SKATER, and their software implementations are suitable for problems with hundreds or even thousands of observations. However, they do not work for data with hundreds of thousands of observations. In our case with about 500,000 cells, regionalization functions from the `rgeoda` R package were unable to return any results.

Therefore, the only viable workflows are the ones with supercells preprocessing. First, supercells based on the compositional raster input are created using the extended SLIC algorithm with the Jensen-Shannon distance as a dissimilarity metric, and parameters $S = 15$ and $m = 0, 1$. These calculations resulted in 3,505 supercells (an about 0.6% of the original cells' number). Area-weighted inhomogeneity of the obtained supercells was 0.09, while its average isolation equaled to 0.26.

The KM-CCL workflow requires specifying the number of clusters (k), which are then split into adjoining regions. For the values of k of 2, 3, 4, 15, and 100, we have obtained 468, 690, 1034, 1947, and 2986 regions, respectively. We then regionalized the same supercells using the SKATER to the same number of regions as was produced by the KM-CCL workflow.

Figure 3 shows a comparison between the two regionalizations of the data into 468 regions. In the two maps in the upper row, the regions are shown by their boundaries (in white) superimposed on the background that visually represents the similarity of cells' objects. The background maps are constructed as false-colors images where the green color was assigned to PC1 (the first principal component calculated from the entire compositional raster; it positively relates to the forest fraction), the color red was assigned to PC2 (it positively relates to the built-up areas), and the color blue was assigned to PC3 (it positively relates to inland water areas). Together, these three principal components account for 93% of variation in the data. The two maps in the bottom row of Figure 3 indicate different regions by random colors.

The qualitative difference between the two regionalizations is best observed in the random color maps. The KM-CCL map shows two large regions, a few medium-size regions, and a large number of small regions. The largest region covers 2,235 km^2 or 53% of the entire area. The area distribution is highly right-skewed. The SKATER map has the area distribution closer to the normal distribution, and its largest region has an area of only 501 km^2.

Yellow numbers in Figure 3 highlight some specific differences between the KM-CCL and the SKATER regionalizations. The first three numbers show areas undersegmented by the KM-CCL regionalization. Locations 1 and 2 are part of the large region (shown in light purple color on the random colors KM-CCL map), but the PCA-based background clearly shows that these locations should not be in this region. The region is predominately forested, but a portion of location 1 is water-dominated and another portion is build-up-dominated. Location 2 is build-up-dominated. Similarly, according to the PCA-based map, the single region in location 3 is actually inhomogeneous and should be broken up into several sub-regions. The SKATER regionalization was able to isolate these sub-regions.

In location 4 the PCA-map suggests that SKATER oversegments the raster, there is not much visual difference between this segment and large neighboring segments located to its west and south. However, a closer look at all of the eight input variables suggests that this region has a larger share of croplands

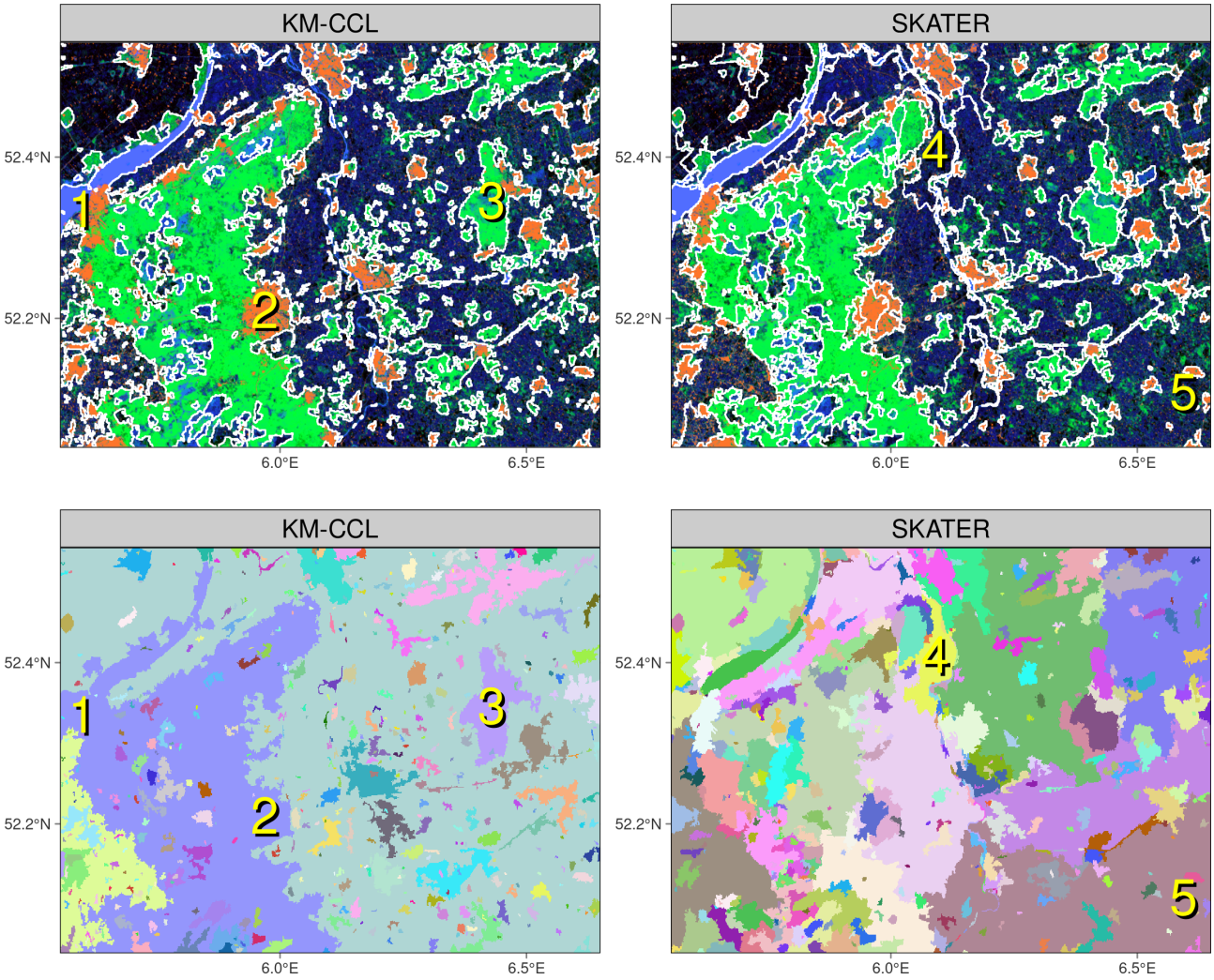

Figure 3. Comparison of the regionalization borders between the KM-CCL approach (left) and the SKATER approach (right) for 468 regions. Yellow numbers highlight areas with different regions between the two regionalizations. Top panels: the white borders of both regionalizations are superimposed on the RGB image created based on the principal component analysis on the input raster with the fractional cover of the eight land cover classes. Bottom panels: random colors represent all of the obtained regions. More explanation about the base map and the yellow numbers can be found in the text

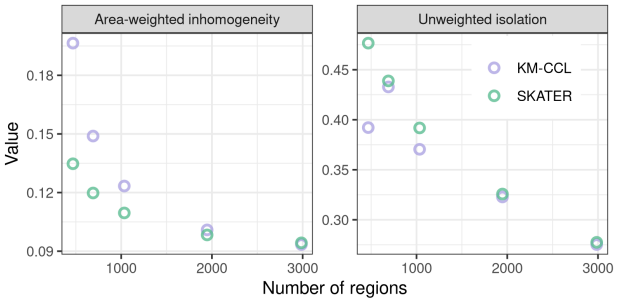

Figure 4. Comparison of area-weighted inhomogeneity (the lower the better) and unweighted isolation (the larger the better) beteen various number of regions created using the KM-CCL and the SKATER approach

and grasslands and a smaller share of forested areas compared to its neighbors. Location 5 highlights one of the few areas for which the KM-CCL regionalization created several small regions compared to a few larger ones created by the SKATER regionalization.

To quantitatively compare the two regionalizations we used inhomogeneity and isolation metrics. The resulting values allow determining the quality of each region independently, with inhomogeneity showing how internally inconsistent the region is while isolation expresses how much different the region is from its neighbors. To compare different regionalizations (sets of regions) an average value of each metric needs to be derived. However, regions have various sizes, and thus is necessary to calculate an average of inhomogeneity weighted by the region area. For isolation, an unweighted (arithmetic) average can be calculated. Values of both metrics for both regionalizations are shown in Figure 4.

The area-weighted inhomogeneity shows an expected decrease with the increased granularity (decreased size of regions due to their higher numbers); smaller regions are more homogeneous due to surface autocorrelation. At the same time, there is a visible difference between the KM-CCL and the SKATER with regard to how metrics change with granularity. For the three largest granularities (468, 690, and 1034), values of inhomogeneity for SKATER regionalizations are significantly smaller (better) and the values of isolation are larger (better) than the respective values for KM-CCL regionalizations. Values of metrics converge when granularity approaches the size of supercells.

Figure 5 provides additional insight into the differences in isolations between the KM-CCL and SKATER regionalizations. All of the KM-CCL regionalizations have visibly right-skewed distributions of the isolation values. This means that most of the regions have low isolation values while few regions have high (good) isolation values. On the other hand, values of isolation in SKATER regionalizations, especially those for the larger granularity, are more evenly distributed, meaning that a larger number of regions are standing apart from their neighbors (as should be the case).

4. CONCLUSIONS

The goal of this contribution was to test the regionalization of non-imagery geospatial data using supercells preprocessing. The ultimate goal is to divide the scene into contiguous internally homogeneous regions that stand apart from their immediate neighbors. It is important to differentiate between a regionalization map and a thematic map. The thematic map shows the spatial distribution of land types, whereas the regionalization map delineates all unique zones of similar land.

Of the two workflows tested, the SKATER regionalization is a natural choice, whereas KM-CCL is not. Thus, we expected that SKATER should outperform KM-CCL. We first checked that regionalization without supercells preprocessing is technically impractical for the raster with ∼500,000 cells, so supercells have to be used. Then we calculated regionalization maps using two competing workflows (Figure 3) and compared their quality metrics (Figures 4 and 5).

Visual comparison of regionalization maps produced by two workflows reveals qualitatively different area distributions of regions. SKATER workflow results in an approximately normal distribution of region areas, whereas KM-CCL workflow has power-law-like distribution with more than half of the area concentrated in a single, largest region, and a large number of very small regions. These maps also reveal undersegmentation errors of the KM-CCL workflow.

Figure 4. provides quantitative support for the higher quality of the SKATER regionalization. For relatively coarse granularities (but still rather fine by geographical standards), the regions produced by SKATER are of higher quality, with both inhomogeneity and isolation metrics having more desirable values than in the KM-CCl case. For granularity equal to 486, the inhomogeneity of SKATER regionalization is 0.135 compared to 0.09 for supercells despite regions having, on average, an order of magnitude larger areas than supercells. The average isolation of regions has improved to 0.5 from 0.26 for supercells. For the same value of granulation, Figure 5 shows a rather flat distribution of isolation values between 0.2 and 0.7. This is in contrast to the distribution of isolation values for regions in the KM-CCL regionalization where most of the regions have isolation values between 0.2 and 0.4. Overall, the SKATER regionalization is successful.

Surprisingly, the KM-CCL regionalization, although worse than the SKATER one, is better than we initially expected. Recall, that it was obtained by first clustering all supercells into just two clusters (for the coarsest granularity). Thus, based on the non-spatial component of the data, only two groups have been identified. Disaggregation of these two groups based on spatial contiguity (by the CCL) results in a map that shows regions of more than two types. We attribute this result to the following. The original two data groups are inhomogeneous, but part of this inhomogeneity is removed by spatial separation. That is, a non-spatial group included data from different, internally more homogeneous, spatial locations. However, we don't expect this kind of improvement to happen with all scenes and all data types. Therefore, we do not recommend using KM-CCL workflow for regionalization purposes.

The complete analysis presented in this paper was possible by using free and open-source software, mainly R packages `supercells` and `rgeoda`. The code allowing to reproduce this work is available at `https://github.com/Nowosad/foss4g2022_reg`.

Future work on the regionalization of geospatial data (both, more conventional, like multi and hyperspectral images, as

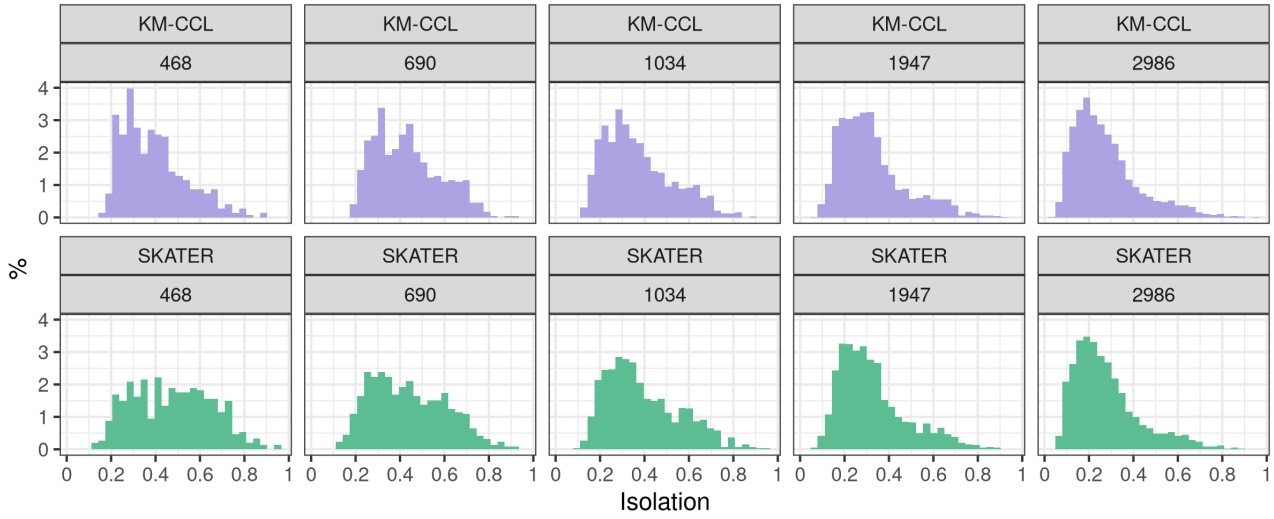

Figure 5. Distributions of the isolation metric values (the larger the better) for various number of regions created using the KM-CCL (top row), and the SKATER approach (bottom row)

well as less conventional, like the data presented in this contribution), will examine different regionalization algorithms and other data types. There are numerous algorithms for graph-based regionalization (Aydin et al., 2021). For example, our preliminary tests suggested that the REDCAP algorithm (Wang et al., 2018) provides results with values of average inhomogeneity and isolation very similar to those presented here for the SKATER algorithm, but with different delineation of regions. This is not surprising since regionalization is an optimization task, and thus it is NP-hard. All available algorithms employ heuristics to obtain a useful but suboptimal solution, since the differences in regions delineation. We also plan to test the presented workflow on a variety of geospatial datasets, including categorical rasters, spatial time-series, spatial patterns, etc. These datasets will require the use of different distance functions.

ACKNOWLEDGEMENTS

This work was supported by the National Science Centre (Poland) under grant number 2019/03/X/ST10/00776, and the grant 038/04/NP/0020 funded by the Initiative of Excellence - Research University project at Adam Mickiewicz University, Poznan.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274–2282.

Aggarwal, C. C., Hinneburg, A., Keim, D. A., 2001. On the Surprising Behavior of Distance Metrics in High Dimensional Space. G. Goos, J. Hartmanis, J. van Leeuwen, J. Van den Bussche, V. Vianu (eds), *Database Theory — ICDT 2001*, 1973, Springer Berlin Heidelberg, Berlin, Heidelberg, 420–434.

AssunÇão, R. M., Neves, M. C., Câmara, G., Da Costa Freitas, C., 2006. Efficient Regionalization Techniques for Socio-economic Geographical Units Using Minimum Spanning Trees. *International Journal of Geographical Information Science*, 20(7), 797–811.

Aydin, O., Janikas, M. V., Assunção, R. M., Lee, T.-H., 2021. A Quantitative Comparison of Regionalization Methods. *International Journal of Geographical Information Science*, 35(11), 2287–2315.

Bailey, R. G., Zoltai, S. C., Wiken, E. B., 1985. Ecological Regionalization in Canada and the United States. *Geoforum*, 16(3), 265–275.

Blaschke, T., 2010. Object Based Image Analysis for Remote Sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1), 2–16.

Buchhorn, M., Lesiv, M., Tsendbazar, N.-E., Herold, M., Bertels, L., Smets, B., 2020. Copernicus global land cover layers – collection 2. *Remote Sensing*, 12(6), 1044.

Cha, S.-H., 2007. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4), 300–307.

Csillik, O., 2017. Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels. *Remote Sensing*, 9(3), 243.

Grygorash, O., Zhou, Y., Jorgensen, Z., 2006. Minimum spanning tree based clustering algorithms. *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, IEEE, 73–81.

Haralick, R. M., Shapiro, L. G., 1985. Image Segmentation Techniques. *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING*, 100–132.

Hartigan, J. A., Wong, M. A., 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28(1), 100.

Hijmans, R. J., 2021. Terra: Spatial Data Analysis.

Jasiewicz, J., Stepinski, T., Niesterowicz, J., 2018. Multi-Scale Segmentation Algorithm for Pattern-Based Partitioning of Large Categorical Rasters. *Computers & Geosciences*, 118, 122–130.

Li, X., Anselin, L., 2022. Rgeoda: R Library for Spatial Data Analysis.

Lin, J., 1991. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1), 145–151.

Netzel, P., Stepinski, T., 2016. On using a clustering approach for global climate classification. *Journal of Climate*, 29(9), 3387–3401.

Nowosad, J., 2022. Regional: Intra- and Inter-Regional Similarity.

Nowosad, J., Stepinski, T., 2021. Generalizing the Simple Linear Iterative Clustering (SLIC) Superpixels. *GIScience 2021 Short Paper Proceedings. 11th International Conference on Geographic Information Science. September 27-30*, 2021. Poznań, Poland (Online).

Olson, D. M., Dinerstein, E., 2002. The Global 200: Priority Ecoregions for Global Conservation. *Annals of the Missouri Botanical Garden*, 89(2), 199.

Omernik, J. M., 1987. Ecoregions of the Conterminous United States. *Annals of the Association of American Geographers*, 77(1), 118–125.

Pebesma, E., 2018. Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, 10(1), 439–446.

R Core Team, 2021. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.

Ren, Malik, 2003. Learning a classification model for segmentation. *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE, Nice, France, 10–17 vol.1.

Shannon, C. E., 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27, 55.

Stutz, D., Hermans, A., Leibe, B., 2018. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166, 1–27.

Subudhi, S., Patro, R. N., Biswal, P. K., Dell'Acqua, F., 2021. A survey on superpixel segmentation as a preprocessing step in hyperspectral image analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 5015–5035.

Tennekes, M., 2018. tmap: Thematic Maps in R. *Journal of Statistical Software*, 84(6), 1–39.

Wang, M., Dong, Z., Cheng, Y., Li, D., 2018. Optimal Segmentation of High-Resolution Remote Sensing Image by Combining Superpixels With the Minimum Spanning Tree. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1), 228–238.

Wickham, H., 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.