Generation of Plateau-Approximated Fuzzy Zones

Hazaël Jones[1], Serge Guillaume[2], Patrice Loisel[3], Brigitte Charnomordic[3], Bruno Tisseyre[1]

[1]Montpellier SupAgro, UMR ITAP, Montpellier, France
[2]Irstea, UMR ITAP, Montpellier, France
[3]INRA, UMR MISTEA, Montpellier, France

*Corresponding author: hazael.jones@supagro.fr

Abstract

When modelling spatial real data in environmental context, the border between two zones is rarely a sharp edge. Often, the transition from one zone to another is a gradual process. Fuzzy zones intend to model this uncertainty in zones by allowing a spatial point to belong partially to different zones with different membership degrees. Generating fuzzy zones is not easy to perform and can be time-consuming and difficult to interpret. It is also difficult to ensure the convexity of the generated fuzzy zones.

Our idea is to generate plateau zones instead of fuzzy zones. Plateau zones are an approximation of fuzzy zones. The interest of using plateau zones lies in the following: easier spatial coherence and convexity of zones, preservation of uncertainties, level and number of plateaus can be chosen depending on the application, interpretation and easier operations between zones.

Our algorithm is based on quantiles of spatial data in order to produce some isocontours. According to the desired number of plateaus, it is possible to adjust the quantile values to find next plateaus for each resulting zones. The goal of this representation is to provide the user a simplification of the spatial representation and to preserve uncertainties in order to use them in the decision process.

Keywords

Fuzzy Zones, Plateau Zones, Spatial Data, Zoning Algorithm, Spatial Coherence

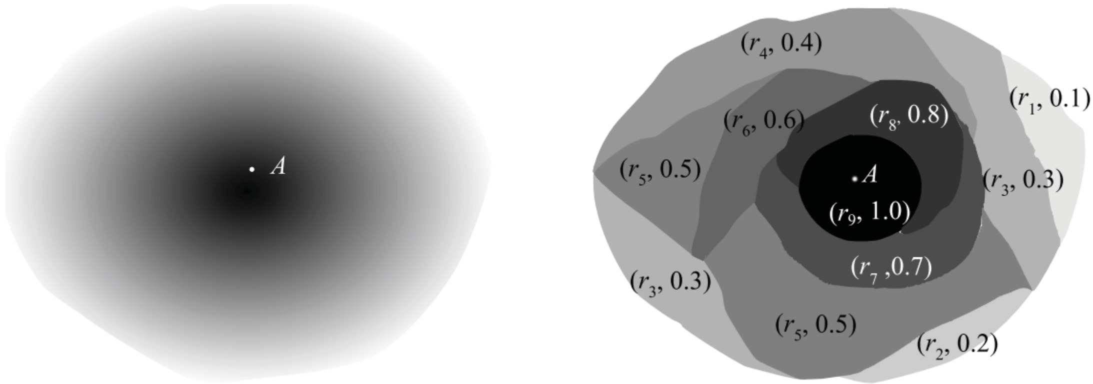
INTRODUCTION

When summarizing spatial data to consider zones (agronomy, geography, etc.), the border between two zones is rarely a sharp edge. In this context, there are no obstacles to detect but management zones to define. Often, the transition from one zone to another is a gradual process (e.g. between clay soil and sand soil, we can have a transition zone with more and more sand).

Fuzzy zones could be used to model this uncertainty/gradual change in zones by allowing a spatial point to belong to several zones with different membership degrees. Fuzzy zones keep the zone uncertainties until the final decision is taken, in order to be able to use this information to relax operational constraints.

Many methods exist to build zones based on segmentation (Pal and Pal (1993), Pedroso et al. (2010), Roudier et al. (2008)) but little work has been done on the generation of fuzzy zones (see Crane and Hall (1999), Philipp-Foliguet et al. (2009)). In Philipp-Foliguet et al., the membership degrees are assigned based on the distance to typical zones and attribute values, it is then difficult to ensure convexity of the generated fuzzy zones.

Plateau zones (see Kanjilal et al. (2010)) are an approximation of fuzzy zones as shown in Figures 1 and 2. Each plateau corresponds to a level with a membership degree. Kanjilal et al. (2010) have studied the interest of plateau zones for computation, but there are no computation methods to generate these plateau zones. This article is a first step to provide a method for such generation.



(a) (b)

Figure 1: An example of a fuzzy zone modelling an air-polluted area (a) and its representation as a plateau zone (b) *source : Figures from article* Kanjilal et al. (2010)*.*

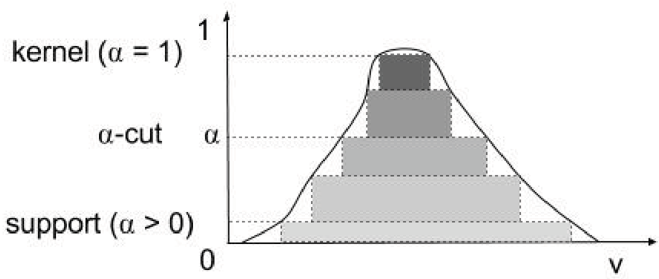


Figure 2: A 2-dimensional illustration of a fuzzy zone and its approximation by plateau zones.

Plateau zones present an interest for several reasons:

- Enhanced spatial coherence and convexity of membership degree for each zone,
- The level and the number of plateaus can be chosen depending on the application. Particularly in agriculture, it is not possible to apply a continuous treatment between one level and another.
- Interpretation: it is more efficient to use a map with plateau zones than continuous fuzzy zones in the decision making phase. Experts often reason using categories and not using a continuum of fuzzy values.
- Effective geometrical operations can be calculated between zones (difference, union, intersection, etc.) thanks to the simplification obtained by plateau (see Kanjilal et al. (2010)).

The goal of this representation is to provide the user a simplification of the spatial representation and to preserve uncertainties in order to use them within the decision process.

THE ALGORITHM FOR BUILDING PLATEAU ZONES.

The algorithm is based on quantiles of spatial data. The use of quantiles on spatial data intends to produce some isocontours. An isocontour (level set) is a line with the same value all along its length. According to the desired number of plateaus, it is possible to slightly adjust the quantile to find the next plateaus for each resulting zone. Quantiles used on spatial data are an efficient way to ensure the imbrication of plateaus with a spatial coherence. If expert knowledge is available on the data (thresholds to divide the data, knowledge about data distribution, etc.), it is possible to use it to set the quantile values. When no *a priori* information is available, quantiles can be chosen in order to divide data into equal proportions. The choice of the best quantile is not the focus of this paper.

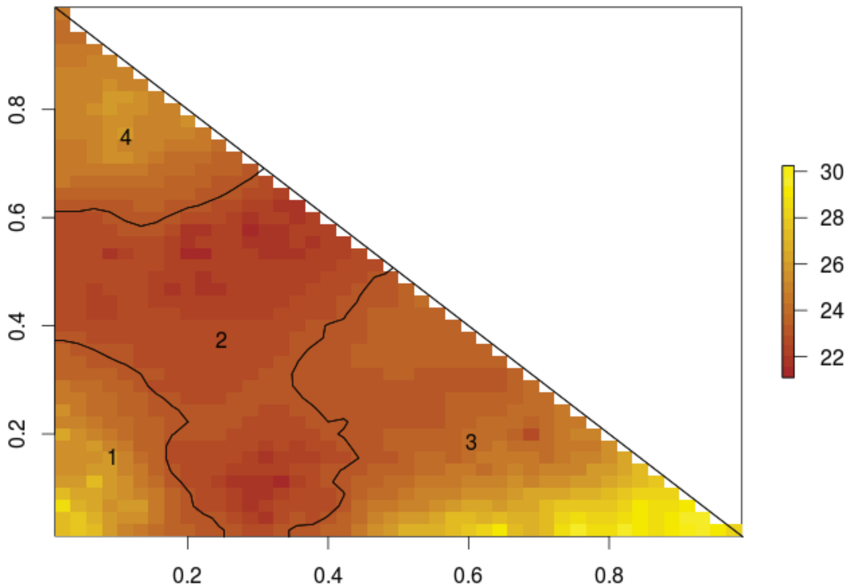
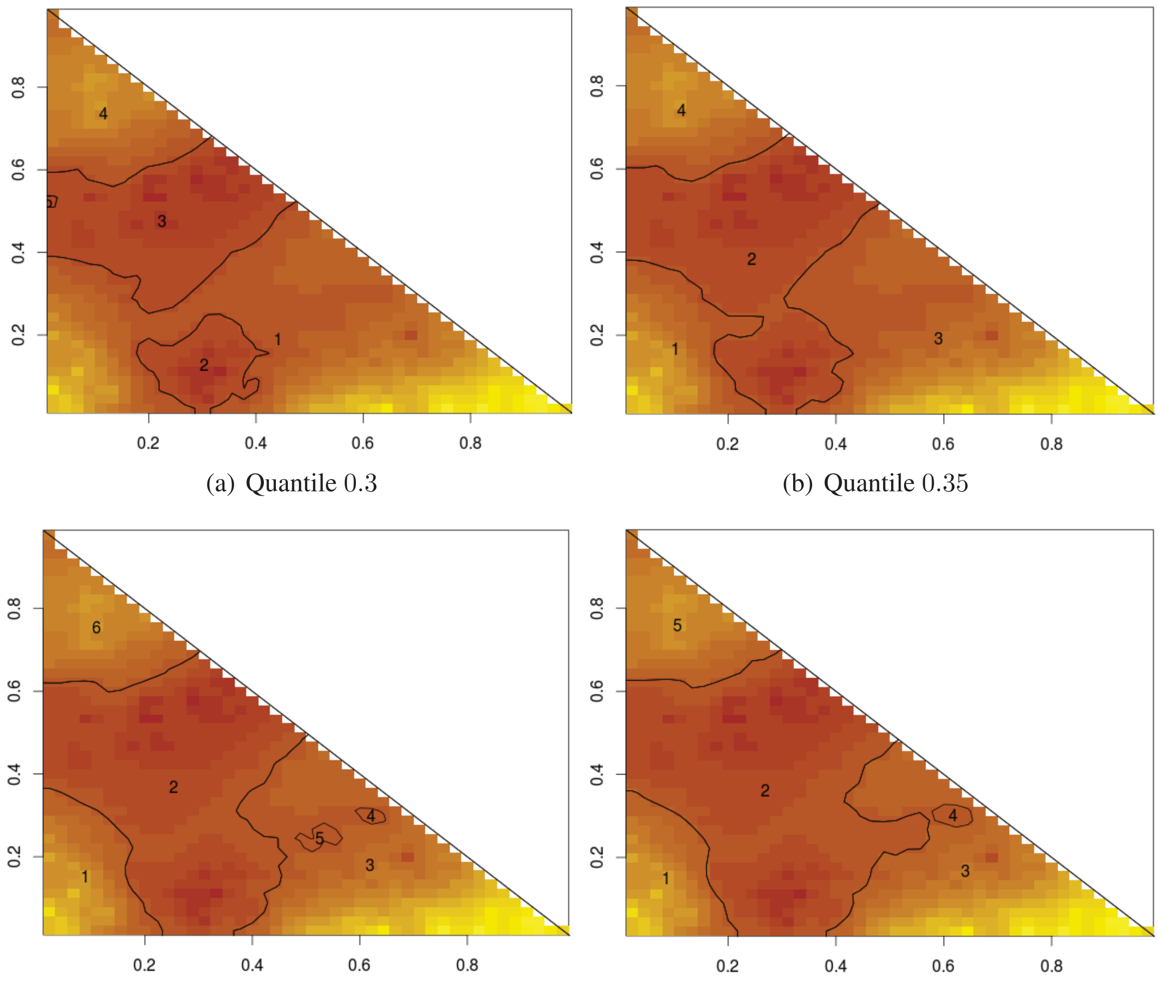


Figure 3: Initial zones based on a quantile.

Figure 3 shows the initial zoning given by one quantile (0.4) on simulated mono-dimensional geo-referenced data (generated using a Gaussian random field). Four distinct initial zones are obtained, and our goal is to first obtain plateaus from them, and to generate approximated fuzzy zones from these plateaus. We will focus on Zone 2 to illustrate the algorithm.

Figures 4(a), 4(b), 4(c) and 4(d) shows how the shape of Zone 2 evolves when the quantile is slightly adjusted. When applied to Zone 2, the algorithm can provide a computation of the kernel, α-cuts and support of this zone (see Figure 2 for an illustration of this concepts). Figure 4(a) shows the potential kernels of Zone 2.

Algorithm 1 is the proposition for building approximated fuzzy zones from plateau regions. For an initial zone and an initial quantile, our procedure BUILDPLATEAUZONES produces *nPlateau* zones associated to *nPlateau* quantile values. It is based on the following principle: as the initial zone (based on quantile 0.4) is selected to divide zones properly, it is granted a membership of 0.5. When we define zones thanks to quantiles inside this zone, the membership increases (see Algorithm 1, lines 6-7), and when we define them outside of this zone, it decreases (lines 8-9). The deeper (further) the zone is inside (outside) the initial zone, the more (less) it is characteristic of the zone. The algorithm allows a membership degree between 0 and 1 (line 7 between 0.5 and 1 - line 9 between 0 (0 excluded) and 0.5). A plateau with a membership degree of 0 is not relevant as it will not be part of the zone. Formula $0.5 - \frac{n}{nPlateau+1}$ (line 9) avoids this case.



(a) Quantile 0.3 (b) Quantile 0.35

(c) Quantile 0.45 (d) Quantile 0.5

Figure 4: Zoning for different quantile values

Algorithm 1 Algorithm for building plateau zones

1: **procedure** BUILDPLATEAUZONES(*initialZone, quantile, step, nPlateau, data*)
▷ *listPlateaus* is a list of zones with associated memberships degrees
2: *listPlateaus* $\leftarrow \emptyset$
3: ADDZONEDEGREE(*listPlateaus*, *initialZone*, 0.5)
▷ compute new zones
4: $n \leftarrow 1$
5: **while** $n \leq nPlateau/2$ **do**
▷ compute new internal zones (memberships > 0.5)
6: *newZoneInt* $\leftarrow$ COMPUTEZONEINT(*initialZone*, *quantile* $- n *$ *step*, $data$)
7: ADDZONEDEGREE(*listPlateaus*, *newZoneInt*, $0.5 + \frac{n}{nPlateau}$)
▷ compute new external zones (memberships < 0.5)
8: *newZoneExt* $\leftarrow$ COMPUTEZONEEXT(*initialZone*, *quantile* $+ n *$ *step*, $data$)
9: ADDZONEDEGREE(*listPlateaus*, *newZoneExt*, $0.5 - \frac{n}{nPlateau+1}$)
10: $n \leftarrow n + 1$
11:
12: **function** COMPUTEZONEINT(*initialZone, quantile, data*)
13: *relevantNewZones* $\leftarrow \emptyset$
▷ return a list of zones based on *quantile*
14: *newZones* = COMPUTEZONES(*quantile, data*)
15: **for all** $Z \in$ *newZones* **do**
16: **if** $Z \subseteq$ *initialZone* **then** ADDZONE(*relevantNewZones*, Z)
▷ return the largest zone that belongs to eligible zones
17: **return** (BIGGERZONE(relevantNewZones), *quantile*)
18: **function** COMPUTEZONEEXT(*initialZone, quantile*, data)
19: *relevantNewZones* $\leftarrow \emptyset$
▷ return a list of zones based on *quantile*
20: *newZones* = COMPUTEZONES(*quantile, data*)
21: **for all** $Z \in$ *newZones* **do**
22: **if** $Z \supseteq$ *initialZone* **then** ADDZONE(*relevantNewZones*, Z)
▷ return the smallest zone that belongs to eligible zones
23: **return** (SMALLESTZONE(relevantNewZones), *quantile*)
24: **function** COMPUTEZONES(*quantile, data*)
25: *zonesFromQuantile* $\leftarrow \emptyset$
26: *isoContours* $\leftarrow$ GENERATEISOCONTOURS$(quantile, data)$
27: *zonesFromQuantile* $\leftarrow$ CLOSEZONES$(isoContours, data)$
28: **return** (*zonesFromQuantile*)

When zones are generated thanks to a new quantile (functions COMPUTEZONEINT and COMPUTEZONEEXT), relevant new zones must be included in the initial zone for COMPUTEZONEINT (and must include the initial zone for COMPUTEZONEEXT). As a single quantile can generate multiple contour lines, if several zones are relevant, we choose the one with the largest area for COMPUTEZONEINT (smallest for COMPUTEZONEEXT). An illustration of this choice can be seen on Figure 4(a) where zone 3 will be chosen as it is bigger than zone 2. COMPUTEZONES returns the list of zones generated on a quantile isocontour basis, it builds isocontours based on data (line 26), and closes them respecting the field border (line 27). The initial zone is built following the same principle but initial zoning is edited by removing/expanding zones which are too small.

Some functions are not specified in the algorithm to be more concise. ADDZONEDEGREE adds a zone and the associated membership degree to the final list of plateaus *listPlateaus*. ADDZONE adds a zone in the relevant zones. BIGGERZONE and SMALLESTZONE select the largest zone (respectively, the smallest) between a set of relevant zones. GENERATEISOCONTOURS produces isocontours from data values corresponding to a quantile and close them respecting the field border (CLOSEZONES).

The result of the computation on Zone 2 is given in Figure 5. The dark green line represents the border of the kernel (plateau of level 1) of Zone 2. 0.75 plateau is in green, 0.5 in black, 0.3 in blue, 0.17 in light blue. The same procedure is followed separately for each initial zone.

Thanks to the use of quantiles, the shape of Zone 2 plateaus is not just a dilatation-erosion of the initial zone because plateaus are directly based on data.

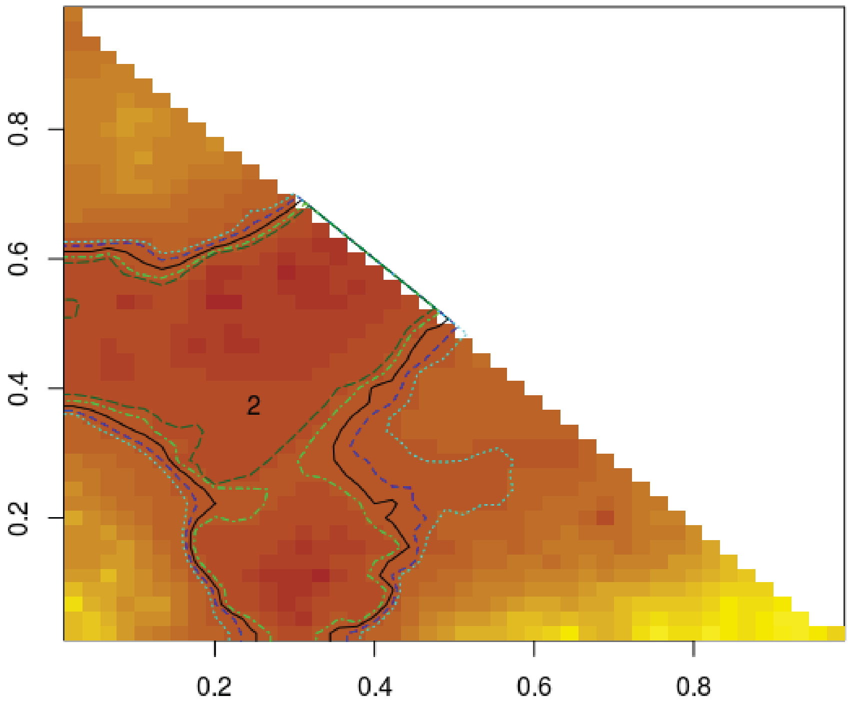


Figure 5: Final result with 5 plateaus for zone 2 (dark green (kernel), light blue (support)).

The algorithm allows the computation of fuzzy regions thanks to a plateau approximation. The computation is efficient (2.15 seconds for 5 plateaus on Intel core i7, 8 cores 2.7GHz, data size = 968) and provides plateaus on spatial data. Plateau zones are relevant because they provide both a simplification of the spatial representation and also the preservation of uncertainties.

This preliminary work can be extended in the following directions: First, it will be interesting to analyse the sensitivity of parameters. The initial quantile choice and the quantile step will influence the results. The bigger the value of the step, the wider the plateaus, the larger the uncertainty of each zone. Furthermore, a study considering the number of plateaus should be done in order to decide how many plateaus are needed to give the most relevant approximation. Finally, we want to show the interest of this representation using real agronomical data.

References

Crane S. E., Hall L. O. (1999). Learning to identify fuzzy regions in magnetic resonance images. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pp. 352–356. IEEE.

Kanjilal V., Liu H., Schneider M. (2010). Plateau Regions : An Implementation Concept for Fuzzy Regions in Spatial Databases and GIS. In *Proceedings of the 13th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 624–633.

Pal N. R., Pal S. K. (1993). A review on image segmentation techniques. *Pattern recognition 26*(9), 1277–1294.

Pedroso M., Taylor J., Tisseyre B., Charnomordic B., Guillaume S. (2010). A segmentation algorithm for the delineation of agricultural management zones. *Computers and Electronics in Agriculture 70*(1), 199–208.

Philipp-Foliguet S., Gony J., Gosselin P. H. (2009). FReBIR: An image retrieval system based on fuzzy region matching. *Computer Vision and Image Understanding 113*, 693–707.

Roudier P., Tisseyre B., Poilvé H., Roger J.-M. (2008). Management zone delineation using a modified watershed algorithm. *Precision Agriculture 9*(5), 233–250.