Spatial Clustering in SOLAP Systems to Enhance Map Visualization

Ricardo Silva, Universidade Nova de Lisboa, Portugal

João Moura-Pires, Universidade Nova de Lisboa, Portugal

Maribel Yasmina Santos, Universidade do Minho, Portugal

ABSTRACT

The emergence of the SOLAP concept supports map visualization for improving data analysis, enhancing the decision making process. However, in this environment, maps can easily become cluttered losing the benefits that triggered the appearance of this concept. In order to overcome this problem, a post-processing model is proposed, which relies on Geovisual Analytics principles. Namely, it takes advantage from the user interaction and the spatial clustering approach in order to reduce the number of elements to be visualized when this number is inadequate to a proper map analysis. Moreover, a novel heuristic to identify the threshold value from which the clusters must be generated was developed. The proposed post-processing model takes into account the query performed, i.e., the number of spatial attributes, the number of spatial dimensions, and the type of spatial objects selected from dimensions. The results obtained so far show: (i) the novel approach to support queries with two spatial attributes from different dimensions allows useful analysis; (ii) the proposed post-processing model is very effective in maintaining a map suitable to the user's cognitive process; and, (iii) the heuristic proposed provide the user participation in the clustering process, in a user-friendly way.

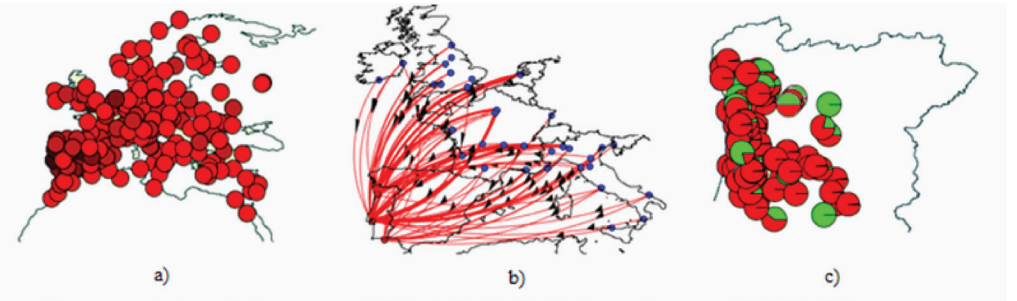
Keywords: Data Visualization, DBSCAN, Geovisual Analytics, SOLAP, Spatial Clustering

CONTEXT AND MOTIVATION

Most OLAP applications are focused on textual data and numerical measures even though available studies have concluded that 80% of data is associated with spatial information (Bédard, Rivest, & Proulx, 2006). Consequently, the integration of spatial data within the multidimensional model was envisaged. Rivest, Bédard, and March (2001) defined the Spatial OLAP (SOLAP) concept as "*a visual platform built especially to support rapid and easy spatio-temporal analysis and exploration of data following a multidimensional approach comprised of aggregation levels available in cartographic displays as well as in tabular and diagram displays*". SOLAP systems allow the integration of spatial data, included either in dimensions (spatial dimensions) or in fact tables (spatial measures), in OLAP ap-

DOI: 10.4018/jdwm.2012040102

Figure 1. Examples of cluttered maps



plications enabling cartographic displays on those applications (Rivest, Bédard, & March, 2001). This way, thematic maps are produced by using the members of spatial dimensions and the numerical measures, combining them with the visual variables (Rivest, Bédard, Proulx, Nadeau, Hubert, & Pastor, 2005).

In this new environment for the analysis of spatial data, several benefits have been mentioned from thematic maps visualization enabled by SOLAP systems. Among them we found better and faster global perception of query results, and the possibility to discover correlations between phenomena, as detailed in Bédard, Rivest, and Proulx (2006).

For the users, it is worth to mention that there is a difference in the results that can be analyzed by an OLAP user compared to a SOLAP user. A typical result of the former involves one to two dozen lines with the aggregated data. However, the latter may involve hundreds of lines if the user is interested in the analysis of data at a lower level of granularity (e.g., customer level). If this is the case, and depending on the geographical distribution and the spatial objects' representation, a thematic map can easily become cluttered and hard to analyze, as illustrated in the examples displayed in Figure 1.

Regarding the example shown in Figure 1a, the points are airports locations and the brightness associated to them is given by the value of a numerical measure. For this map, the user is unable to compare the numerical measure value of different airports as many markers are hidden due to an overlapping between them. The same happens in Figure 1c, where we have pie charts with information of several facilities in a geographical location. In the Figure 1b, the blue markers correspond to the departure airports; the red markers are the arrival airports; and, the value of the numerical measure is given by the arc's width. Again, the map comprehension/visualization becomes harder due to the high number of arcs and their superposition. Moreover, this last example is a result based on a novel approach to support queries which has involved two spatial attributes from different spatial dimensions (in the query select clause). This novel approach is proposed in this work and further detailed in subsequent sections of this paper.

The presented examples result from the interaction with SOLAP+ system (Jorge, 2009; Silva, 2010), which is afterwards described in this paper. In all of them, the map showed to be inadequate to the user visualization and corresponding analysis.

Thus, in a SOLAP system it is necessary to control the number of results returned to the user in order to maintain the usefulness of maps in the decision making process. To maintain the benefits that come from map visualization, this paper extends our previous work (Silva, Moura-Pires, & Santos, 2011) by proposing a post-processing stage that relies on Geovisual Analytics principles (Andrienko et al., 2007; Keim, Andrienko, Fekete, Görg, Kohlhammer, & Melançon, 2008; Andrienko, Keim, MacEachren, & Wrobel, 2011), combining the

human strengths with computational data processing instead of relying only in computational data processing. Namely, it takes advantage from the user interaction and the spatial clustering approach. The post-processing stage is applied before the results (maps, data tables and graphics) are presented to the user.

This approach takes into account the amount of spatial information to be displayed on the map and the possible overlapping between the different representations. Moreover, it gives to the user the ability to control the existence, or not, of the post-processing stage. When the post-processing stage is used, the user can also control the clustering level based on a new heuristic or if the clustering process is constrained by a spatial hierarchy present in the multidimensional model. The proposed heuristic is used to determine the threshold value, affecting the number of generated clusters. Also, this process is query-aware, taking into consideration: (*i*) the number of spatial attributes; (*ii*) the number of spatial dimensions; (*iii*) the type of spatial objects selected from dimensions (points or polygons); (*iv*) the numerical measures and the used aggregation operator; (*v*) the semantic attributes and their relation (of granularity) with the spatial attributes.

Setting our solution based on the geovisual principles, the user's preferences can be properly taken into account as well as the quality of the results can be improved.

The following sections include: (*i*) some related work associated with the area of SOLAP and spatial clustering algorithms; for the former area we present the main SOLAP concepts, characteristics needed in SOLAP applications, and discuss an important property present in the SOLAP+ system; for the latter area we discuss four methods of spatial clustering algorithms based on the purpose of this work; (*ii*) the post-processing model proposed in this work to deal with the huge complexity that can emerge in the analysis of spatial data in a SOLAP system. Also, it is presented a novel approach to handle queries which have two spatial attributes involved from different spatial dimensions, and a new heuristic to determine the threshold value used to generate the clusters (*iii*) the prototype that was developed to implement the proposals presented in this work; and, (*v*) some remarks about the presented work and some guidelines for future work.

RELATED WORK

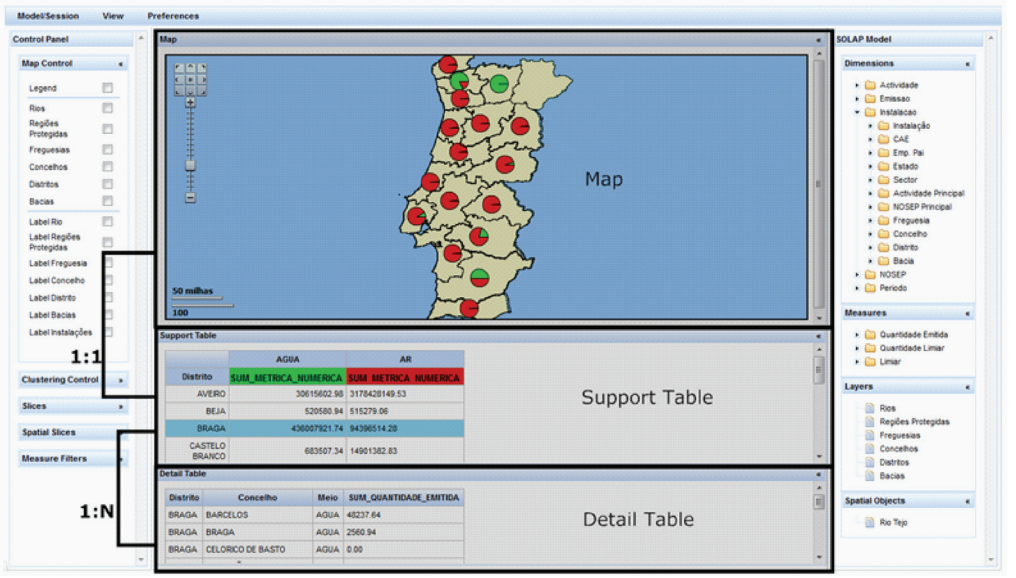
This section presents a set of research literature related to SOLAP, to the SOLAP+ system, to the spatial clustering algorithms, and to the Geovisual Analytics area.

SOLAP Concepts and Systems

The integration of spatial data into a multidimensional model adds it two new main concepts: spatial dimensions and spatial measures. The use of spatial measures is a widely discussed subject but it is far from having an agreement on it (Rivest, Bédard, Proulx, & Nadeau, 2003; Malinowski & Zimányi, 2007; Bimonte, Tchounikine, & Pinet, 2010). A spatial dimension is a dimension that includes one or more attributes related to spatial objects. Each dimension can index data at several detail/aggregation levels. Hierarchies can then be defined using the levels of a dimension. Malinowsky and Zimányi (2005) have defined different kinds of spatial hierarchies in spatial dimensions. In general, the most common hierarchies are those whose relationships between their members can be represented as a tree, named *simple spatial hierarchies*. For example, the time dimension can have the day, week, month and year attributes as a *simple spatial hierarchy*.

A SOLAP system should incorporate, according to Rivest, Bédard, and March (2001), three main areas: visualization, exploration and structure of data. For data visualization, the authors argue that cartographic displays should allow adequate exploration of the geometric component of the spatial data being analyzed (from spatial dimensions members), as well as the use of contextual information. Also, they refer the need to include statistical diagrams into cartographic displays, such as bar charts, pie charts, among others, in order to obtain sum-

Figure 2. The SOLAP+ interface



marized information of the data being analyzed. These combinations of elements, which can be displayed on a map, call our attention to the need of maintaining a legible and organized map. There are more guidelines related to the other two areas. They are not further explored since they are out of the scope of this paper.

Based on the concepts and guidelines already presented, the SOLAP+ system was developed (Jorge, 2009; Silva, 2010). This system will incorporate in a new version the proposals made in this paper. For a description of other SOLAP tools please refer to Gómez, Kuijpers, Moelans, and Vaisman (2009) and Bimonte (2010).

The SOLAP+ framework includes three main components: the map, the support table and the detail table, as shown in Figure 2. In the map, thematic maps, spatial objects and other spatial information are presented to the user. The support table is used to show the data in an alphanumeric format from which the map representation depends on. The detail table is a tool that provides to the user a more in-depth analysis.

There is one line in the support table for each spatial object displayed on the map (from the query result). This 1:1 relationship is maintained in any situation and should always be kept. Its purpose is to facilitate the user's cognitive process relating the alphanumeric data with their spatial location/representation.

If this relationship is not verified, the following scenarios could be verified: (*i*) multiple lines in the support table related to one spatial object on the map; (*ii*) one line in the support table related to multiple spatial objects on the map. Looking at the first case, the spatial object's information could be spread along the table, making harder (for the user) to relate the information in the table to the object on the map. Besides that, it can lead to confused thematic maps. For the second case, it will prevent the analyst from realizing the contribution of each spatial object to the global information present in one line of the support table. Plus, if we consider spatial objects far from each other, the user's cognitive process relating the data present in one line of the support table with the map will become very difficult.

The detail table is used to apply drill-down operations on a line of the support table. As a result, *n* lines are obtained in the detail table for each line in the support table. Further details about the SOLAP+ system can be found in Silva (2010).

Spatial Clustering Algorithms

Clustering is the process of grouping a set of objects into clusters in such a way that objects in the same cluster have high similarity with each other, but are as dissimilar as possible to objects located in other clusters (Miller & Han, 2009). Spatial clustering techniques have emerged to deal with the growing amount of spatial data that have been stored in spatial databases. Those techniques revealed great potentialities in the generalization of the spatial component present in spatial databases, reducing the number of elements to be observed and represented.

To uncluttering map visualization in a SOLAP context, using a clustering-based approach, the selected spatial clustering algorithm should verify a set of requirements. From the set of requirements mentioned in Kolatch (2001), the most meaningful to this work are: (*i*) no *a-priori* knowledge, like the number of clusters, should be asked because the request of several input parameters will turn the application very demanding, from a user point of view; (*ii*) the algorithm must be able to quickly process large amounts of data, avoiding the introduction of high delays in the data visualization process that results from the proposed post-processing model; (*iii*) the algorithm should identify groups with arbitrary shapes since it is expected that real datasets contain clusters with irregulars shapes.

Despite the high number of existing spatial clustering algorithms, they can be categorized in four methods (Miller & Han, 2009): (*i*) partitioning; (*ii*) hierarchical; (*iii*) density-based; (*iv*) grid-based.

In the partitioning method, *k* partitions are created forming *k* groups of data. The partitioning is created by attempting to optimize an objective partitioning criterion, such as the distance between the objects. The number of groups must be set in advance. The partitioning methods include algorithms like k-Means (Hartigan & Wong, 1979), k-Medoid (PAM) (Kaufman & Rousseeuw, 1990) and CLARANS (Ng & Han, 2002).

In the hierarchical method, a hierarchical decomposition is performed from an initial dataset. From a hierarchical decomposition results a dendrogram that shows the hierarchical structure of clusters. Usually, there is no need to specify the number groups, yet it is required to set the end condition for the decomposition process. In this process, a key decision is related to whether the objects must be, or not, unified. A wrong decision cannot be corrected later on. To overcome this issue other clustering techniques were integrated with hierarchical algorithms, emerging the multi-phase hierarchical algorithms, including Chameleon (Karypis, Han, & Kumar, 1999). Other examples of hierarchical algorithms are CURE (Guha, Rastogi, & Shim, 1998) and BIRCH (Zhang, Ramakrishnan, & Livny, 1996).

Concerning the density-based algorithms, they adopt the straightforward idea of identifying clusters as dense regions of objects that are separated from clusters with lower density of objects. The main advantage of this approach is the ability to discover clusters with irregular shapes. Some relevant examples of density-based clustering algorithms are DBSCAN (Ester, Kriegel, Sander, & Xu, 1996), P-DBSCAN (Joshi, Samal, & Soh, 2009) and SNN (Ertöz, Steinbach, & Kumar, 2003).

Regarding the grid-based method, its algorithms quantize the original space into a finite number of cells, creating a multi-level grid structure. The clustering operations are performed on the quantized space. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects as it only depends on the number of cells in each dimension in the quantized space. Examples of algorithms of this approach are CLIQUE (Agrawal, 1998) and WaveCluster (Sheikholeslami, Chatterjee, & Zhang, 2000).

In the context of this work, the partitioning algorithms are not considered since they require as input parameter the number of clusters. Looking at the grid-based algorithms, we may have efficient algorithms but they also need several input parameters, for which no heuristic approach is available to make them user independent. This is also true for the hierarchical algorithms. Excluding these three types, remain the density-based algorithms. These algorithms also require input parameters. However, some studies have been carried out to define heuristics that estimate the values of the input parameters (Ester, Kriegel, Sander, & Xu, 1996; Sander, Ester, Kriegel, & Xu, 1998). Although user interaction continues to be needed, this paper proposes a novel heuristic, which is presented later in this paper.

From the several density-based algorithms, this work adopted DBSCAN. No benchmarking was carried out to select this algorithm. This selection was only guided by the fact that there is a variant of DBSCAN to cluster regions (P-DBSCAN). This is a relevant issue in the context of this work, as we can have spatial objects represented as points and others represented as regions.

Geovisual Analytics

The post-processing model proposed in this work, which is conceived to enhance map visualization in SOLAP environment, incorporates some guidelines present in the Geovisual Analytics area (Andrienko, Andrienko, Keim, MacEachren, & Wrobel, 2011).

Like other Geovisual Analytics approaches (Sips, Schneidewind, & Keim, 2007; Yildizli, Pedersen, Saygin, Savas, & Levi, 2011; David De Chiara, 2011) we need to handle huge amounts of data. In our case, this problem may arise when a SOLAP user makes a query at lower level of granularity as we already discussed. Moreover, we also mentioned that spatial clustering techniques revealed great potentialities in reducing the number of elements to be displayed on the map.

However, automatic data analysis only works well for well-known problems (Keim, Andrienko, Fekete, Görg, Kohlhammer, & Melançon, 2008). In order to overcome this problem, the Geovisual Analytics area proposes a model that combines interactive visualization with computational data processing. Therefore, humans and machines cooperate using their respective distinct capabilities in order to get effective results. Consequently, this model has implicit two components: (*i*) data processing component; (*ii*) interactive visualization component.

The data processing component is used to synthetize the data dealing with the complexity and the huge amount of data problem. The interactive visualization component is used not only to overcome the limitations present in the data processing component (e.g., algorithms' parameters) but also to provide to the user the capability to guide the results that come from the data processing component, instead of being just left with the results. This way, it allows the user to take advantage of the results and to adjust/improve them whenever it is needed.

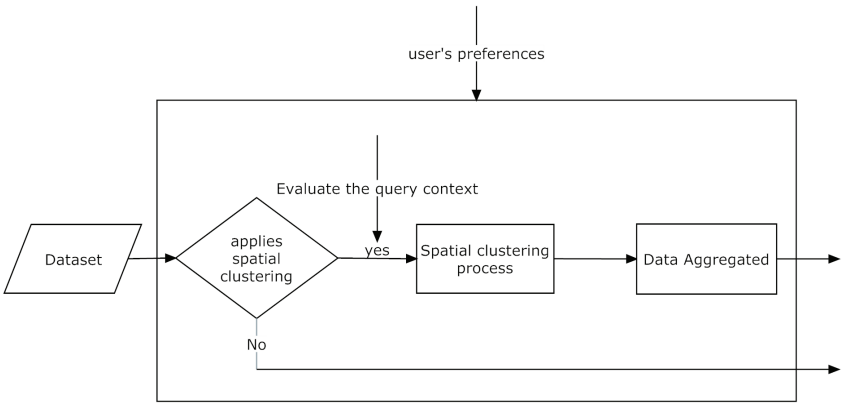
POST-PROCESSING MODEL

Given a query in a SOLAP context, the goal of the post-processing model is to display the data in an organized and understandable way for the user, maintaining the benefits that emerge from map visualization and its added-value to the user's analysis.

In order to provide an analyzable map, we extend our previous work (Silva, Moura-Pires, & Santos, 2011) proposing a post-processing model that combines the user interaction with the spatial clustering approach. This way, the user satisfaction can be enhanced, once he can adjust the results to his preferences. An overview of the defined post-processing model is presented in Figure 3.

In first step it is evaluated if the spatial clustering process should be applied or not. There are three options: (*i*) the user does not

Figure 3. The post-processing model



apply; (*ii*) the user does apply; (*iii*) the post-processing component decides automatically.

For the last option, a heuristic is needed to make that evaluation. Our main objective is to ensure proper objects visibility, but the performance should also be taken into account since the clustering process will introduce an extra processing time component. This component may introduce a significant time, or not, in the overall time needed to display the results. Therefore, a heuristic should compute this decision, based on indicators about the legibility and the performance. If one of them is verified, then the post-processing should be performed. The heuristic used to obtain these indicators should introduce a very small extra processing time, when compared with the time of the clustering process.

Regardless the used spatial clustering algorithm, the algorithm will be applied to the real coordinate space (geographical coordinates). The input parameters are influenced by the zoom level leading to a clustering with more or less clusters. Even if the results are not all displayed, due to the zoom level, the clustering process is applied to all the input data. In the visualizations, a panning doesn't change the clusters as a zooming (in or out) does.

The most common spatial objects associated to spatial attributes are points (e.g., customers locations) or polygons (e.g., country administrative divisions). At this stage, it is necessary to identify the type of spatial object we are dealing with, as the post-processing model needs to choose the proper clustering algorithm and the respective distance function. Despite the type of spatial object, the spatial clustering algorithm should include the requirements presented and discussed previously in the related work section.

In order to accomplish our goal, we propose two approaches in the post-processing model: an ad-hoc or a region-based clustering. The first approach creates clusters without any semantic meaning apart from the geographical proximity among objects. Alternatively, the region-based clustering creates groups that are constrained by a spatial hierarchy, besides the geographical proximity that is also considered. Therefore, the clustering approach to be applied depends of the spatial object and the selected approach (by the user). Nevertheless, a spatial clustering algorithm is always applied to the spatial objects.

After the clusters identification, a new representation for each cluster is generated, decreasing the number of spatial objects that need to be displayed on the map. If a new spatial representation is generated for each cluster, the non-spatial data should also be aggregated allowing the synchronization between the map and the tabular display, as it is recommended in Rivest, Bédard, and March (2001). As a result, this process has an important impact in the

tabular display, as the data can be presented at different levels of granularity.

As it was mentioned previously, the post-processing model is query-aware and it is defined following the next notation:

- **Semantic Dimension (*sD*)** is a dimension where all levels contain semantic attributes.
- **Spatial Dimension (*spD*)** is a dimension with one or more levels that contain spatial attributes.
- **Spatial Attribute (*spA*)** is a spatial attribute of a *spD* level.
- **Semantic Attribute (*sA*)** is a textual or numerical attribute from a *sD* or a *spD*
- **Numerical Measure (*nM*)** is a numerical value associated to a fact and stored in the fact table. The numerical measure associated to an aggregation operator is represented as *nM*(*aggO*).
- **Query (*Q*)** defines a representative query. It could have spatial attributes, semantic attributes and numerical measures. We assume that *Q* may have: (*i*) one or two *spA* and the corresponding *sA*(*spA*); (*ii*) zero or more *sA*; (*iii*) one or more *nM*, each one with an associated aggregation operator.
- **Spatial Hierarchy (*spH*)** is a simple spatial hierarchy composed by *n* levels.

In the following sections, it will be detailed the post-processing model based on representatives queries: (*i*) one spatial attribute: point; (*ii*) one spatial attribute: region; (*iii*) semantic attributes; and, (*iv*) two spatial attributes (from the same or from different spatial dimensions)

One Spatial Attribute: Point

Whenever the query result involves a spatial point attribute, the clustering point flow of the post-processing model will be performed. For now, we only consider queries that fit into the following representative query:

$$Q\{spA_1, sA_1(spA_1), nM_1(aggO_1), \ldots, nM_n(aggO_n)\}$$

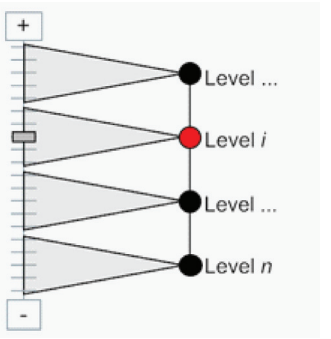
This representative query contains one spatial attribute with the associated semantic attribute and one or more numerical measures. An example of such query could be: *What is the total amount of carbon dioxide emissions by facilities that are within 5 kms radius from a city?* In this query, spA_1 represents the facilities locations, $sA_1(spA_1)$ is associated to the facilities names and $nM_1(SUM)$ is the corresponding total amount of the carbon dioxide emissions.

Initially, all spA_1 values resulting from Q are extracted and a spatial clustering algorithm is applied to them. The algorithm result will be the spA_1 values associated to one or no cluster. After that, two actions are performed for each group: the definition of a new representation and the aggregation of data with the appropriate aggregation operator (data must be aggregated using the aggregation operator associated to each nM).

There are several possibilities to create a new representation for each group, such as: centroid, convex or concave hull. Each one of them has its advantages in some specific application domains. If we use a polygon solution as a new representation then it would restrict the number of possible thematic maps that can be created, since it will be impossible to use the visual variable size to express some attribute or numerical measure. Therefore, our proposal for a new cluster representation is to combine a polygon with a point representation. This way, the possibility to apply the visual variable size is maintained and at the same time the user has information about the area covered by the cluster. Once clusters with arbitrary and irregular shapes are expected, the use of a concave hull algorithm (Moreira & Santos, 2007), when compared with a convex hull approach, seems appropriate.

The other approach for clustering point data is the region-based clustering. This method is very similar to the previous. However, the clusters that result from the spatial clustering algorithm share the spatial attribute value that comes from the computed level.

Figure 4. Level that constraints the clustering process (level i)



Consider that *spH* is the hierarchy chosen by the user, which must include spA_1. The considered *spH* levels have to be at a higher level of granularity compared to the spA_1 level and have to be represented by polygons. The level that constraint the clusters is computed in the following way: the map zoom levels are divided equally by the hierarchy levels and the resulting level is given by the zoom level as illustrated in Figure 4. Despite the proposed approach, this level can be manually selected by the user.

One Spatial Attribute: Region

In this section, we assume the same representative query but in this case spA_1 values are associated to polygons. A possible query could be: *What is the total amount of carbon dioxide emissions by counties that overlaps one or more protected regions?*

When the spatial objects are represented by polygons, which cannot typically overlap, a negative impact on map visualization can arise not only by the polygons visualization, but also by the visualization of charts or other elements associated with them.

The clustering process for polygons is similar to the clustering process for points. Initially, all the spA_1 values are obtained. Then, it is applied a spatial clustering algorithm suitable for polygons. Finally, for each group of polygons, a cluster, a new representation is computed and the data is aggregated in a proper way as we mentioned in the previous section. In this case, the new representation is the union of the polygons.

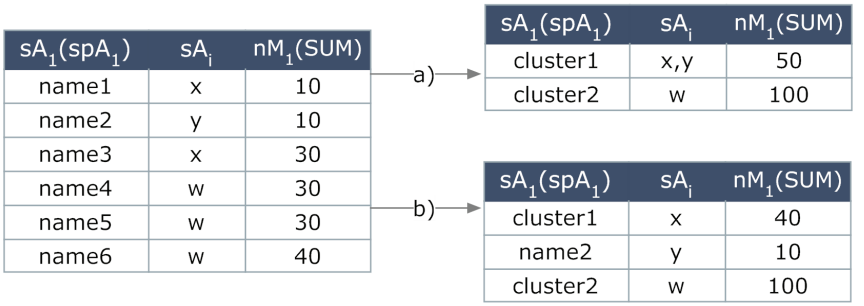
New issues arise in the process of clustering polygons. In the clustering of points, the Euclidean distance or other similar metric is used. Whatever the distance function is, it is expected the time complexity to be constant. The same is not true for polygons. A good way to measure the distance between two polygons is through the Hausdorff distance (Atallah, 1983). Unfortunately, the computation of the Hausdorff distance is expensive from the computational point of view, even though there are works that attempt to minimize it, such as Atallah (1983). Although those efforts, there is no work, to the best of our knowledge, that achieved a constant time complexity.

To overcome this issue we propose the precomputing of the distances among the polygons in the spatial dimensions. When a spatial clustering algorithm is applied to the query result, only the distances already computed are needed. Through this solution the time complexity for the distance function goes constant.

Including Semantic Attributes

So far, we have excluded the semantic attributes from the query. Including the semantic attributes into the query gives us the following representative query:

Figure 5. The tabular form after the post-processing stage is applied in first case: a) without constraint; b) with constraint



$$Q\{spA_1, sA_1(spA_1), sA_1, \ldots, sA_j, nM_1(aggO_1), \ldots, nM_n(aggO_n)\}$$

An example of a query in this context could be: *What is the total amount of air and water pollutant emissions by counties that overlaps one or more protected regions?*. In this query, spA_1 includes the counties polygons, $sA_1(spA_1)$ is associated to the counties names, sA_1 describe the type of pollutant emissions (air or water) and $nM_1(SUM)$ is the corresponding total amount of the pollutant emissions.

When we are dealing with semantic attributes we need to consider two distinct cases: (*i*) sA_i comes from the same spD that spA_1; (*ii*) sA_i comes from another dimension (where $1 \leq i \leq j$).

In the first case, it is important to look at the level at which both spA_1 and sA_i are. If sA_i is at the same or at a higher level than the spA_1 then there is only one value of sA_i for each spA_1. For these cases we introduce a new constraint to the clustering process: each cluster must share the sA_i value (regardless the selected clustering approach).

Consider the following representative query: $Q\{spA_1, sA_1(spA_1), sA_i, nM_1(SUM)\}$. In Figure 5 the sA_i verifies the first case and suppose that the *name1*, *name2* and *name3* form one cluster and the remaining values form another. Without the proposed constraint, the correspondence between the values of spA_1 and sA_i could change after the post-processing stage as illustrated in Figure 5a. This restriction was introduced to maintain the analysis with the same tabular representation. This way we maintain the 1:1 relationship between the tabular form and the map (Figure 5b).

In the second case, the post-processing stage applies a straightforward approach that maintains the relations between the attribute values as depicted in Figure 6, illustrating the previous Q.

Two Spatial Attributes

In the previous sections we have assumed queries with only one spatial attribute. However, we may have two spatial attributes. In these cases the representative query is:

$$Q\{spA_1, sA_1(spA_1), spA_2, sA_2(spA_2), sA_1, \ldots, sA_j, nM_1(aggO_1), \ldots, nM_n(aggO_n)\}$$

For this representative query, two distinct cases need to be considered: spA_1 and spA_2 are associated to the same spD or spA_1 and spA_2 came from different $spDs$. These two approaches are next described.

Same Spatial Dimension

In Jorge (2009) it is presented a generic case where there are two spatial attributes in the

Figure 6. The tabular form after the post-processing stage is applied in the second case

$sA_1(spA_1)$	sA_i	
	Value 1	Value 2
	$nM_1(SUM)$	$nM_1(SUM)$
name1	10	20
name2	5	8
name3	30	14

$sA_1(spA_1)$	sA_i	
	Value 1	Value 2
	$nM_1(SUM)$	$nM_1(SUM)$
cluster1	45	42

query and these attributes belong to different spatial hierarchies. For these cases, the author proposed the intersection between spA_1 and spA_2, under some conditions (see Jorge, 2009, for more details). A possible query in this context could be: *What is the total amount of air and water pollutant emissions by counties and watersheds?*. In this query, spA_1 is associated to the counties polygons, spA_2 is associated to the watersheds polygons, $sA_1(spA_1)$ and $sA_2(spA_2)$ are the respective counties and watersheds names, sA_1 describes the type of pollutant emissions (air or water) and $nM_n(SUM)$ is the corresponding total amount of the pollutant emissions.

The proposed approach works well since the original Q becames similar to a context presented previously:

$$Q\{spA_1 \bigcap spA_2,\ sA_1(spA_1),\ sA_2(spA_2),\ sA_1,\ nM_1(aggO_1),\ldots,\ nM_n(aggO_n)\}.$$

Different Spatial Dimensions

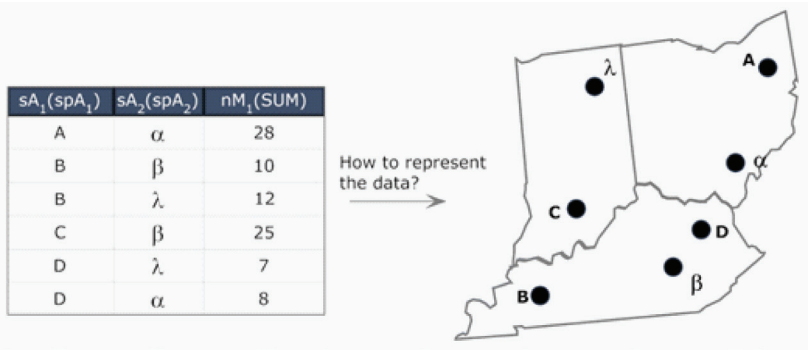
To the best of our knowledge, the existing SOLAP systems do not support queries with two spatial attributes coming from different spatial dimensions, where only one map and the respective tabular form is used. However, in many SOLAP application domains, there is the need and usefulness to perform them, such as travel and tourism, environmental, transportation and retail domains, among others. For instance, in the air transportation domain, a user may want to ask: *What is the total number of passengers who flight from one airport to another during the summer season?*. In this query, spA_1 is associated to the airport locations belonging to the departure dimension, spA_2 is associated to the airport locations belonging to the arrival dimension, $sA_1(spA_1)$ and $sA_2(spA_2)$ are the respective airports names and $nM_1(SUM)$ is the corresponding total number of passengers. As it can be seen in Figure 7, the main issue lies on the maps' data representation. As in any SOLAP system, the user should also take benefit from the map visualization. In order to support queries with two spatial attributes belonging to different spatial dimensions, we proceed with the proposal of a novel approach for such cases, before discussing the behavior of the post-processing model.

To display just one map and the corresponding table, we propose to determine a visual representation for each distinct pair of spatial objects. Following this approach, when the user is looking at the map is able: (*i*) to relate the two spatial objects; (*ii*) to get an overview of the relationships between the spA_1 and spA_2 in a single map, allowing comparative analysis between the data associated to them.

In order to accomplish the previous requirements, each distinct pair of spatial objects is represented on the map as a line (in this case arc) that connects the corresponding spatial objects, as it is presented in Figure 8.

The 1:1 relationship between the table and the map is maintained, which is a relevant property discussed in the SOLAP Concepts and

Figure 7. The problem in SOLAP systems to support queries with two spAs from different spDs



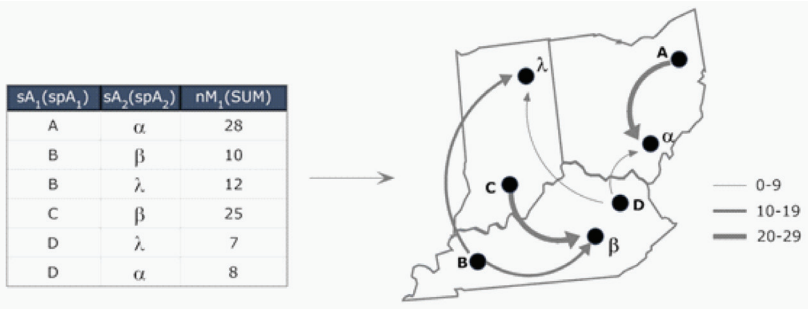
$sA_1(spA_1)$	$sA_2(spA_2)$	$nM_1(SUM)$
A	α	28
B	β	10
B	λ	12
C	β	25
D	λ	7
D	α	8

Systems section. To each line in the table corresponds an arc on the map. Besides this, we can produce thematic maps (Figure 8) or/and use charts (on top of the line) to display the data associated to the relations between the $spAs$, allowing a global and comparative analysis between the data related to them.

Furthermore, in a SOLAP context other types of spatial objects can be involved in the query: (*i*) spA_1 is a point and spA_2 is a polygon or vice-versa; (*ii*) both spA are polygons; (*iii*) spA_1 is a point and spA_2 is a line or vice-versa; (*iv*) spA_1 is a polygon and spA_2 is a line or vice-versa; (*v*) both spA are lines. For example, in the environmental domain, a user may want to know: *What is the total amount of waste released on rivers by facilities in Portugal?*. In this query, spA_1 is associated to the facilities points belonging to the facility dimension, spA_2 is associated to the rivers lines belonging to the river dimension, sA_1 describes the country considered and $nM_1(SUM)$ is the corresponding total amount of waste.

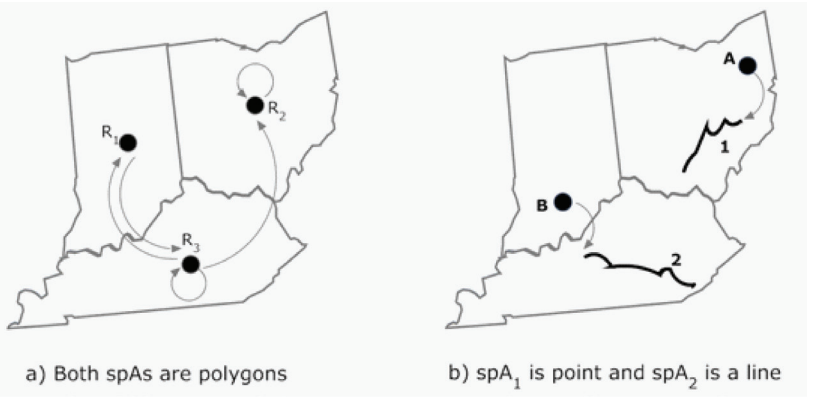
Independently of the spatial objects' type, it is necessary to define the location of the line end points. When a spA is a point, then the end point will be the point itself. For a polygon object, we propose a representative point like its centroid (Figure 9a). For a line object, it will depend on the other end point, i.e., the representative point will be the point where it is verified the minimum distance between the line and the point object (Figure 9b). In addition, if spA_1 and spA_2 are associated to the same spa-

Figure 8. Example of table and the corresponding map using the approach proposed



$sA_1(spA_1)$	$sA_2(spA_2)$	$nM_1(SUM)$
A	α	28
B	β	10
B	λ	12
C	β	25
D	λ	7
D	α	8

Figure 9. Examples of end points for other types of spatial objects



tial object then both endpoints are located at the same position (Figure 9a). This way, independently of the involved spatial objects' types, the user is able to relate the two spatial objects.

The algorithm responsible for the line creation has a huge impact on the usability of our approach. Since we use the arc to connect two endpoints, it is possible to create it with the concave upward or downward. When the rowset is iterated to compute the arcs, we propose the following approach: the arc is computed upward or downward, looking at the minimum number of intersections with the previous arcs already computed. In case of tie, it is a random procedure. Furthermore, if a symmetric relation between two endpoints exists, both arcs have the same concavity orientation as observable in Figure 9. However, the ideal algorithm would be the one that minimizes the overall intersections number. To accomplish this goal, an excessive processing time could be used by the algorithm, something that is undesired to the user. A balance between map readability and processing time need to be achieved.

As in similar examples previously presented, the map easily becomes cluttered. In fact, the map could become even more cluttered than in cases where only one spatial attribute is considered (Figure 10a). Again, the post-processing model plays a key role in the proposed work allowing the displaying of data in an understandable way.

Consider the next representative query:

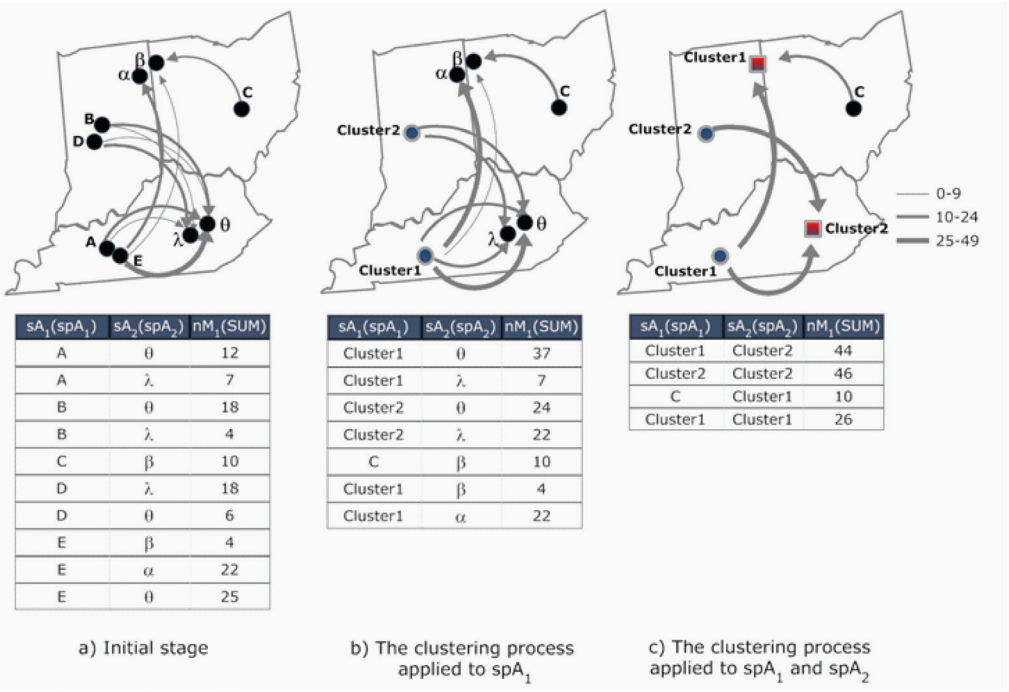
$$Q\{spA_1, sA_1(spA_1), spA_2, sA_2(spA_2), nM_1(SUM)\}$$

A possible result for this representative query is illustrated in Figure 10a. As can be seen, even with a few combinations between spA_1 and spA_2 the map becomes difficult to analyze.

To reduce the amount of data to be displayed, we need to reduce the number of combinations between the $spAs$. In order to reduce this number, we propose the application of the post-processing stage separately to each spA. To each spA it is evaluated if the spatial clustering process should be applied or not. In case there is the need to be applied, it is used the appropriate clustering process (point process or polygon process). Note that, the user interaction is also applied separately. For each spA, the user has the same interactivity that exists for the contexts with one spatial attribute (e.g., the user may choose explicitly to apply ad-hoc clustering to the spA_1 and to apply region-based clustering to the spA_2).

Recalling the context of Figure 10a, suppose that by applying the clustering process to the spA_1 the values *A*, *E* form one cluster, and *B*, *D* form another. Applying to the spA_2, the values λ, Θ form one cluster, and α, β form

Figure 10. The behavior of the post-processing model for queries with two spAs from different spDs



$sA_1(spA_1)$	$sA_2(spA_2)$	$nM_1(SUM)$
A	θ	12
A	λ	7
B	θ	18
B	λ	4
C	β	10
D	λ	18
D	θ	6
E	β	4
E	α	22
E	θ	25

a) Initial stage

$sA_1(spA_1)$	$sA_2(spA_2)$	$nM_1(SUM)$
Cluster1	θ	37
Cluster1	λ	7
Cluster2	θ	24
Cluster2	λ	22
C	β	10
Cluster1	β	4
Cluster1	α	22

b) The clustering process applied to spA_1

$sA_1(spA_1)$	$sA_2(spA_2)$	$nM_1(SUM)$
Cluster1	Cluster2	44
Cluster2	Cluster2	46
C	Cluster1	10
Cluster1	Cluster1	26

c) The clustering process applied to spA_1 and spA_2

another. As can be seen in Figure 10c, we obtain an uncluttered map by applying the clustering process separately to spA_1 (Figure 10b) and then to spA_2.

Let D_1 be the set of the spA_1 values that results from Q, and D_2 the set of the spA_2 values that results from Q. We distinguish the following two cases: (i) $D_1 \bigcap D_2 = 0$; (ii) $D_1 \bigcap D_2 \neq 0$. The example in Figure 10 shows only the scenarios that fit in the first case.

In general, the second case occurs when both spatial attributes represent the same entity. For instance, in the air transportation domain, let us assume that we have the airport departure and the airport arrival dimensions. Although we have two spatial dimensions, the spatial objects are the same (airport locations). For those scenarios, the clustering process is also applied separately. In addition, the result of the post-processing stage is independently of the order in which the clustering process to D_1 and D_2 is applied.

PROTOTYPE

The SOLAP+ is a generic system that relies on a three tier architecture composed by Oracle as the Database Management System (gives support to spatial data), a SOLAP server, and a client coupling the OLAP features with maps. The server was implemented from scratch, in Java, and it is responsible for listening to client requests, processing them and retrieving the appropriate results. The client handles all user interaction, data presentation and request generation. It was implemented in Java Server Faces (JSF) and the communication with the server is performed based on the XML protocol. Also, it uses Oracle Maps JavaScript API (to enhance the functionality of the Oracle MapViewer) in order to support map visualization and interaction.

Our post-processing model was included in the SOLAP server tier. Also, the DBSCAN and the P-DBSCAN algorithms were implemented and included in this tier. To the chosen

algorithms, the object neighborhood is based on some radius (Eps) and an object in the cluster has to contain at least $MinElements$ of elements. Thus, appropriate values for Eps and $MinElements$ need to be identified in order to implement the post-processing model in a user-friendly approach. For the latter parameter, we follow the formula: $MinElements = 2 * Dimensionality - 1$, proposed in Sander, Ester, Kriegel, and Xu (1998). For the former we propose a novel user independent heuristic presented in the next section.

User Independent Heuristic to Determine Eps

In Sander, Ester, Kriegel, and Xu (1998) it is proposed a $k.distance$ function mapping each object to the distance from its k-th nearest neighbor. Based on these $k.distance$ values it is created a plot with those values sorted in descending order, called a $sorted\,k.distance\,plot$ where the k value corresponds to the $MinElements$ value (in 2D space $k=3$). That plot gives some hints concerning the objects density distribution.

When choosing an object obj , it is assigned to the Eps parameter the value $k.dist\ (obj)$, and all objects at right will be considered as core objects, while the objects at left are labeled as noise. In this heuristic, the authors proposed that the user would choose the object at the first "valley", as illustrated in Figure 11 obtained from Sander, Ester, Kriegel, and Xu (1998).

However, on the one hand, this heuristic is not user independent. On the other hand, it does not allow the user participation in the clustering process in a user-friendly way. In order to give to the user a high level of abstraction from the algorithm parameters and to allow his participation in the clustering process, we developed a new heuristic.

The proposed one aims to find not only an appropriate value, but more than one value allowing the user to choose between a result with more or less clusters (for the same map zoom level).

Therefore, we propose a heuristic that searches for "breaks" in the $3.distance$ function. It is on the several breaks that the turning points are found, those that produce clusters with different densities.

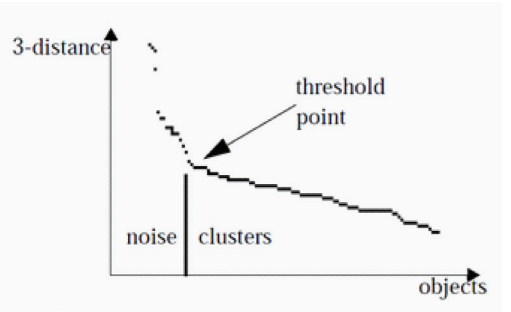
The breaks are obtained as follows: initially, the objects are sorted in an ascending order by the value of the $3.distance$ function. Then, in each iteration, it is calculated the following $\delta_i = 3.distance\ (obj_{i+1} - obj_i)$ and the average of this value is updated $\Delta_i = \sum_{0<j<i} \Delta_j / i$. When $\delta_{i+1} / \Delta_i > \alpha$ it is considered a break where α is an arbitrary value. In such cases, the $3.distance\ (obj_i)$ value is stored and the average is initialized to zero. Furthermore, the $3.distance\ (obj_i)$ value will be used as the Eps parameter. At the end of this process, a set of breaks are obtained, given the $break_i$ value more clusters than the $break_{i+1}$. This set must have at least three values to allow the user to choose between less or more clusters. Otherwise, the previous process is repeated with a decremented α until the resulting set of breaks verifies the previous condition. In our implementation, the α value is initialized to three and it is decremented from one value if the set of breaks has less than three values.

Once again, it were used the Geovisual Analytics principles to overcome the spatial clustering approach limitation and, at the same time, it is given to the user the possibility to participate in the clustering process. More precisely, we avoid the *Eps* parameter needed in the DBSCAN and the P-DBSCAN algorithms, and the user can adjust the results from the clustering process to his visual preferences.

Demonstration Case

In this section, we will present two demonstrations cases based on two real data sets. Through their analysis it will be possible to exemplify the usefulness of the several approaches presented in this paper.

Figure 11. Sorted 3-distance plot



Pollutant Emissions

The first demonstration case is based in a real data set about pollutant emissions in Portugal. The relevant information about the multidimensional model can be summarized as:

1. Contains the spatial dimension Facility. This dimension includes five spatial attributes: location, drainage basin, *Freguesia*, *Concelho*, *Distrito* and the semantic attribute: facility name;

The Facility dimension contains the Portuguese administrative divisions as a spH.

In this demonstration case the data is sliced with respect to three districts. From $Q\{facility\,location, facility\,name, nM_1(SUM)\}$ result, without performing the post-processing stage, the obtained results are displayed in Figure 12a.

Figure 12 also shows the result obtained after the application of the ad-hoc clustering approach. In Figure 12b, Figure 12c and Figure 12d are used the Eps values returned by the proposed heuristic allowing the user to choose between a result with more or less clusters. These results can be changed if the user is not satisfied with them. As the user is unaware of the Eps values only has to drag the slider to the left (more groups) or to the right (less groups) to change the detail associated to the results.

In Figure 13 is displayed the result of the post-processing stage using the region-based approach (based on Figure 12d). The spH chosen is the hierarchy of the Portuguese administrative divisions. The level computed to restrict the cluster from the map zoom level is the District level. Also, the partial results of the support table and the detail table (detailing the *Group0*) are displayed.

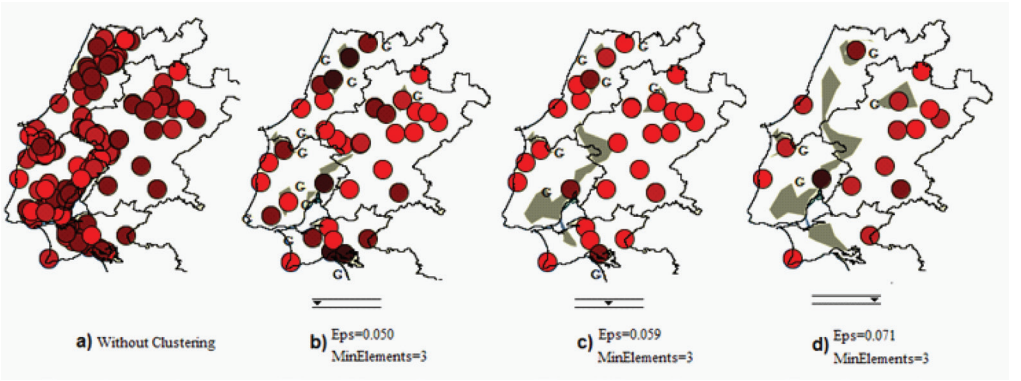
For each cluster it is displayed a delimited area that is covered by it. These areas are computed through the concave hull algorithm that uses as input the coordinates of the objects that are inside each cluster. The numerical measure value is displayed through a marker positioned at the cluster centroid (and over the covered area) that has the same representation of non-clusters markers. The facilities' names are not displayed as this information is confidential.

Booked Flights

The second demonstration case is based also in a real data set related with booked flights. The relevant information about the multidimensional model can be summarized as:

1. Contains two spatial dimensions: The departure dimension and the arrival dimension. Both dimensions contain two spatial attributes (airport location, country) and the corresponding attributes with the designations (the airport name and country name);
2. Both dimensions have one spatial hierarchy composed by the airport level and country level.

Figure 12. The usage of the clustering ad–hoc approach and the proposed heuristic



For this demonstration case the data is sliced to consider booked flights between Portugal, United Kingdom, France and Italy.

The *Q{country departure, country name departure, country arrival, country name arrival, num passengers* (*SUM*)}result is displayed in Figure 14.

Using the proposed approach the user can perform a comparative analysis of the number of passengers in these flights. In this particular case, we can compare the total number of passengers made between Portugal, United Kingdom, France and Italy. Effortlessly, it is visible that the booked flights made from Portugal to France and from Portugal to Italy outnumber the other ones.

For the next query, the data is sliced in order to consider the booked flights from Portugal to France.

The *Q{country departure, country name departure, airport arrival, airport name arrival, num_passengers* (*SUM*)}result is depicted in Figure 15a).

In queries with two spatial attributes the map is more susceptible to be cluttered as is shown in Figure 15a. However, by combining our novel approach with the post-processing stage (Figure 15b), the cluttering problem is overcome. Additionally, it offers the opportunity to compare the booked flights by the regions which resulted from the cluster process creation.

Figure 13. The result of the clustering region–based approach using the Figure 12d context as basis. The support table is above and the detail table below

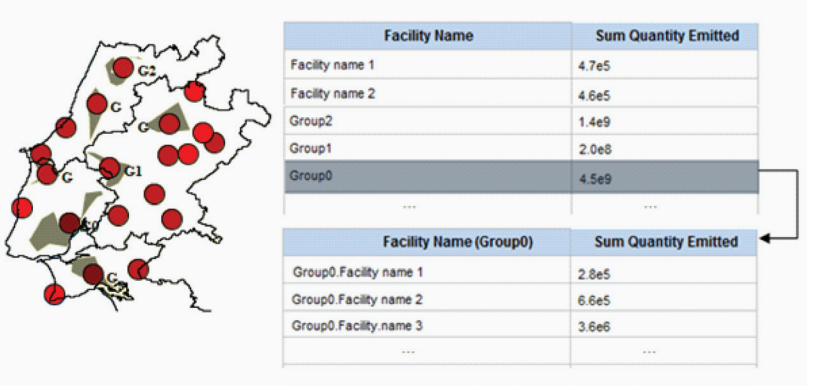
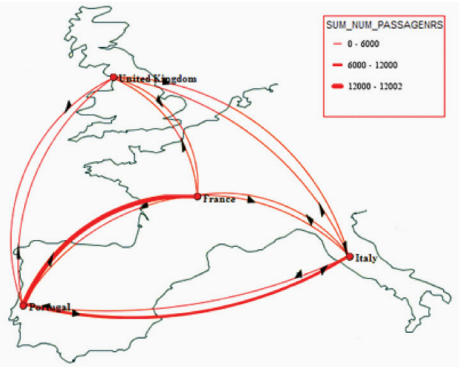


Figure 14. Example of the map result for a context of two spAs from different spDs



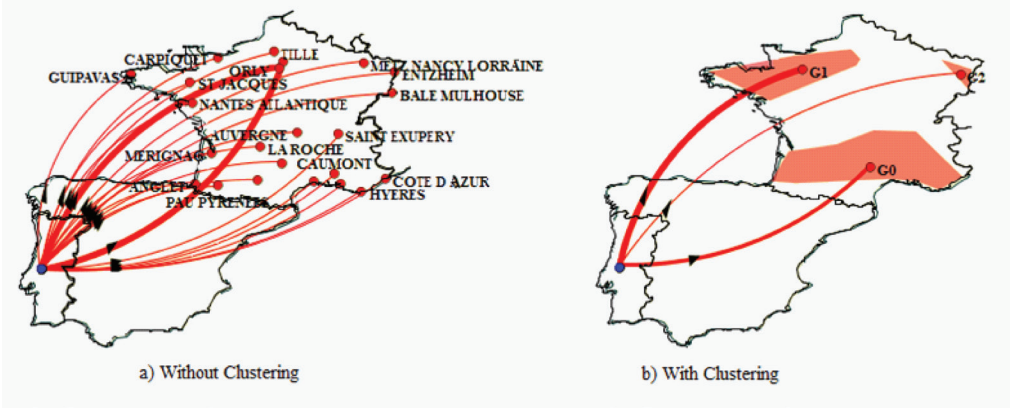
CONCLUSIONS AND FUTURE WORK

In this paper is presented an extension of the post-processing model from our previous work (Silva, Moura-Pires, & Santos, 2011) in order to maintain the benefits that emerge from map visualization in a SOLAP environment. The presented post-processing model has underlying the Geovisual Analytics principles, combining the user interaction with computational data processing.

From the computational point of view, it is used the spatial clustering technique to generalize the spatial component, attached to the SOLAP queries, in order to reduce the number of elements observable in the map. From the user interaction point of view, it is given the capability to control the existence, or not, of the post-processing stage, the level of the clustering process (based on our novel heuristic to estimate the *Eps* DBSCAN/P-DBSCAN parameter), and if the clusters are restricted by a spatial hierarchy. All the data analysis process is driven by a user specified query.

Moreover, the post-processing model is extended in order to support other representative queries. In this paper, we introduced a new representative query which has involved two spatial attributes (in the query select clause) from different spatial dimensions. For this

Figure 15. The usage of the clustering ad–hoc approach with two spAs from different spDs



representative query, we also proposed a novel approach to support it in a SOLAP environment.

The proposed post-processing model attends to balance the amount of spatial information to be displayed and the possible overlapping between representations. Additionally, to the best of our knowledge, there is no other SOLAP system with a mechanism to control the map visualization.

Future work can be directed to the proper evaluation of the post-processing model. We envisage two levels of evaluation. First, we should evaluate the spatial clustering approach to reduce the clutter in the map. Second, we should perform a comparative evaluation about the novel user independent heuristic face to the classic heuristic proposed by the DBSCAN authors. Regarding the first level, we should compare several spatial clustering algorithms in terms of the requirements discussed in the related work section and the quality of the results. Concerning the second indicator, it can be subdivided into a set of sub indicators that characterize the quality of the results, such as: number of clusters, average spacing between clusters, etc. This evaluation should be carried out both with synthetic and real data sets. Also, the level of users' satisfaction in the presence and absence of the post-processing model and the effectiveness of the novel heuristic regarding the number of generated clusters should be evaluated.

Additionally, future work also includes: the automatic identification of the legibility and performance indicators to be used by the post-processing component to automatically decide if the spatial clustering process should be applied or not; the application of the spatial clustering process based on map representations (instead real coordinate space); and, finally, the extrapolation of this approach to other contexts beyond SOLAP, or SOLAP contexts but with spatial measures.

The source of our SOLAP prototype can be downloaded from: http://dl.dropbox.com/u/736321/SOLAPPlus.rar

REFERENCES

Agrawal, R. G. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 94-105).

Andrienko, G. L., Andrienko, N. V., Jankowski, P., Keim, D. A., Kraak, M. J., MacEachren, A. M., et al. (2007). Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of Geographical Information Science*, 839–857. doi:10.1080/13658810701349011

Andrienko, G. L., Andrienko, N. V., Keim, D., MacEachren, A., & Wrobel, S. (2011). Challenging problems of geospatial visual analytics. *Journal of Visual Languages and Computing*, 251–256. doi:10.1016/j.jvlc.2011.04.001

Atallah, M. J. (1983). A linear time algorithm for the hausdorff distance between convex polygons. *Information Processing Letters*, 207–209. doi:10.1016/0020-0190(83)90042-X

Bédard, Y., Rivest, S., & Proulx, M. J. (2006). Spatial on-line analytical processing (SOLAP): Concepts, architectures, and solutions from a geomatics engineering perspective. In R. Wrembel & C. Koncilia (Eds.), *Data warehouses and OLAP: Concepts, architecture* (pp. 298–319). Hershey, PA: IGI Global. doi:10.4018/987-1-59904-364-7.ch013

Bimonte, S. (2010). On modeling and analysis of multidimensional geographic databases. In L. Bellatreche (Ed.), *Data warehousing design and advanced engineering applications: Methods for complex construction, 6*(4) (pp. 96–112). Hershey, PA: IGI Global. doi:10.4018/978-1-60566-756-0.ch006

Bimonte, S., Tchounikine, A., & Pinet, F. (2010). When spatial analysis meets OLAP: Multidimensional model and operators. *International Journal of Data Warehousing and Mining*, 33–60. doi:10.4018/jdwm.2010100103

Davide De Chiara, V. D. (2011). A Chorem-based approach for visually analyzing spatial data. *Journal of Visual Languages and Computing*, 173–193. doi:10.1016/j.jvlc.2011.02.001

Ertöz, L., Steinbach, M., & Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the Third SIAM International Conference on Data Mining* (Vol. 112, pp. 47-59).

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery* (pp. 226-231).

Gómez, L., Kuijpers, B., Moelans, B., & Vaisman, A. (2009). A survey of spatio-temporal data warehousing. *International Journal of Data Warehousing and Mining*, *5*(3), 28–55. doi:10.4018/jdwm.2009070102

Guha, S., Rastogi, R., & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 73-84).

Hartigan, J. A., & Wong, M. A. (1979). A K-means clustering algorithm. *Applied Statistics*, *28*, 100–108. doi:10.2307/2346830

Jorge, R. (2009). *SOLAP+: Extending the interaction model* (Unpublished master's thesis). Universidade Nova de Lisboa, Lisbon, Portugal.

Joshi, D., Samal, A., & Soh, L. K. (2009). Density-based clustering of polygons. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining* (pp. 171-178).

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75. doi:10.1109/2.781637

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. New York, NY: Wiley Interscience.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In A. Kerren, J. T. Stasko, J.-D. Fekete, & C. North (Eds.), *Human-Centered Issues and Perspectives* (LNCS 4950, pp. 154-175).

Kolatch, E. (2001). Clustering algorithms for spatial databases: A survey *(Tech. Rep.)*. Baltimore, MD: University of Maryland.

Malinowski, E., & Zimányi, E. (2005). Spatial hierarchies and topological relationships in the spatial MultiDimER model. In *Proceedings of the British National Conference on Databases* (pp. 17-28).

Malinowski, E., & Zimányi, E. (2007). Logical representation of a conceptual model for spatial data warehouses. *GeoInformatica*, *11*(4), 431–457. doi:10.1007/s10707-007-0022-3

Miller, H. J., & Han, J. (2009). *Geographic data mining and knowledge discovery* (2nd ed.). Boca Raton, FL: CRC Press.

Moreira, A., & Santos, M. Y. (2007). Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In *Proceedings of the Second International Conference on Computer Graphics Theory and Applications* (pp. 61-68).

Ng, R. T., & Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE Transactions on Knowledge and Data Engineering*, 1003–1016. doi:10.1109/TKDE.2002.1033770

Rivest, S., Bédard, Y., & March, P. (2001). Towards better support for spatial decision-making: defining the characteristics. *Geomatica: the Journal of the Canadian Institute of Geomatics*, 539-555.

Rivest, S., Bédard, Y., Proulx, M. J., & Nadeau, M. (2003). Solap: a new type of user interface to support spatio-temporal multidimensional data exploration and analysis. In *Proceedings of the ISPRS Joint Workshop on Spatial, Temporal and Multi Dimensional Data Modelling and Analysis.*

Rivest, S., Bédard, Y., Proulx, M. J., Nadeau, M., Hubert, F., & Pastor, J. (2005). SOLAP technology: Merging business intelligence with geospatial technology for interactive spatio-temporal exploration and analysis of data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 17–33. doi:10.1016/j.isprsjprs.2005.10.002

Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, *2*(2), 169–194. doi:10.1023/A:1009745219419

Sheikholeslami, G., Chatterjee, S., & Zhang, A. (2000). WaveCluster: a wavelet-based clustering approach for spatial data in very large databases. *Very Large Data Base Journal*, 289-304.Silva, R. (2010). *SOLAP+* (Unpublished master's thesis). Universidade Nova de Lisboa, Lisbon, Portugal.

Silva, R., Moura-Pires, J., & Santos, M. Y. (2011). Spatial clustering to uncluttering map visualization in SOLAP. In *Proceedings of the International Conference on Computational Ccience and its Applications - Volume Part I*, Santander, Espanha.

Sips, M., Schneidewind, J., & Keim, D. (2007). Highlighting space-time patterns: Effective visual encodings for interactive decision making. *International Journal of Geographical Information Science*, 879–893. doi:10.1080/13658810701362147

Yildizli, C. B., Pedersen, T., Saygin, Y., Savas, E., & Levi, A. (2011). Distributed privacy preserving clustering via homomorphic secret sharing and its application to (vertically) partitioned spatio-temporal data. *International Journal of Data Warehousing and Mining*, *7*(1), 46–66. doi:10.4018/jdwm.2011010103

<conf>Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 103-114)

Ricardo Silva is a PhD Student in the Computer Science Department at the Faculty of Science and Technology of University Nova de Lisboa, in Portugal. He finished his degree and MSc in 2008 and 2010, respectively, in computer science from Faculty of Science and Technology. Presently, he is taking the Ph.D. in computer science at Faculty of Science and Technology, which motivation is the spatial decision support system (sDSS) area. His research focus on geovisual analytics area, a multidisciplinary field including knowledge discovery, spatio-temporal models, spatio-temporal knowledge representation, geovisualization, information visualization, among others.

João Moura-Pires is an Assistant Professor in the Computer Science Department at Faculty of Science and Technology of University Nova de Lisboa, in Portugal. He has a degree in Electrical Engineer from University of Angola (1984), a Ph.D. in Computer Science from University Nova de Lisboa (2000). He is also a researcher at center of Artificial Intelligence (CENTRIA). His research interests include business intelligence, spatial data warehousing, spatial data mining, spatio-temporal data models, spatial reasoning, fuzzy logic and knowledge representation and reasoning.

Maribel Yasmina Santos is an Assistant Professor in the Information Systems Department at the University of Minho in Portugal. She has a degree in Informatics and Systems Engineering from the University of Minho (1991), a MSc in Informatics and a Ph.D. in Information Systems and Technologies, both from the University of Minho (1996 and 2001, respectively). Currently, she is sub-director of the Information System Department and council member of the Association of Geographic Information Laboratories for Europe (AGILE). She is also member of the ALGORITMI research center. Her research interests include business intelligence, spatial data warehousing, spatial data mining, spatio-temporal data models, and spatial reasoning.