Digital Cartographic Generalization in Spatial Databases: application issues in Power Grids CAD tools

Javier Novo Rodríguez, Elena Hernández Pereira, Mariano Cabrero Canosa

Department of Computer Science, University of A Coruña, Spain

ABSTRACT

This work presents the results of applying several digital cartographic generalization techniques to improve the performance of an Electrical Power Grids Computer-Aided-Design application. The performance increase is attained by adjusting the level of detail of the grid topology being visualized in the CAD application. This adjustment takes place at the database level using a multi-scale architecture and the available spatial extensions of Geographic Information Systems databases. The data volume is minimized to fit the exact requirements of the scale being used in the visualization so that no processing time is wasted on representing irrelevant elements. Results show that up to a 90 % of the data can be skipped and thus, a 84 % of the time required to render the visualization can be saved on average, enabling the CAD application to render it in real-time. Furthermore, the optical legibility of the grid visualization is enhanced as a consequence of applying cartographic generalization.

Categories and Subject Descriptors (according to ACM CCS): J.6 [Computer Applications]: COMPUTER-AIDED ENGINEERING—Computer-aided design (CAD), H.2.8 [DATABASE MANAGEMENT]: Database applications—Spatial databases and GIS

1. Introduction

The electrical distribution grids are complex networks in charge of delivering the electric energy transported through long distances by the much simpler transmission networks to the consumers. They are large and very populated networks which cover entire states and their cities, having complex topologies comprising up to hundreds of thousands of branches. All those branches – formed by one or more power lines – are defined using geodetic coordinates and thus, must be processed before being visualized in a Computer-Aided-Design (CAD) application. Modern graphics cards contain very powerful Graphics Processing Units (GPUs) which easily outperform the Central Processing Units for parallelizable tasks. These GPUs are also programmable which has lead to the increasing popularity of General-Purpose Computing on GPUs in the last few years. Using the GPU to perform the coordinate translation from the geographic to the graphic domain has reduced the rendering time of the grids visualization from seconds to tenths of seconds [NRCCHP10]. However, in order to be able to offer a seamless real-time interaction with the visualization, further optimizations must be introduced so that the amount of data arriving to the GPU is minimized and the visualization can be rendered in less than a tenth of a second.

When interacting with the grid topology through the CAD application, the electrical engineers zoom in and out to the different areas of the network that are relevant to them. They may wish to visualize the topology of a whole country, go down to a state or even zoom into a city or to a couple of blocks. The visualization of each one of these areas requires a different map scale and not all the data from the grid topology is relevant to every scale. Different levels of detail can be applied since, for instance, there is no point in trying to render the details of the streets when a country is being visualized. Although the CAD application will usually visualize more advanced information such as overlays, charts or satellite imaging, this work is focused on improving the most basic visualization of the power grids topology. Such visualization consists on the rendering of the power grid as unitary-width lines. These kind of lines are the most easily processed and rendered and therefore any improvement attained in this scenario will be amplified in more complex ones – as it may be for instance using the width of the lines to represent the loads for power flow analysis.

In order to generate the different levels of detail, digital cartographic generalization is used. In cartography, generalization is the process of abstracting the representation of geographic information to match the scale requirements of a map. This work focuses on graphic cartography generalization, whose objective is to reduce the spatial resolution. The type of symbolization is not changed but the symbols themselves may be enhanced or exaggerated to keep its optical legibility. As a result of applying several graphic generalization techniques, the volume of the data supplied to the GPU has been reduced, yielding significantly faster visualization times for the CAD application. Moreover, the generalization techniques unclutter the visualization and thus, its optical legibility is improved.

This paper is organized as follows. Section 2 presents a characterization of the electrical distribution grids and their data. In Section 3, the cartographic generalization process and techniques are presented, emphasizing those used in this work. Section 4 describes the implementation made using a spatial database, before showing its results analysis in Section 5. Finally, Section 6 presents the conclusions.

2. Data Characterization

Power management comprises disparate areas that conform a very complex system known as the electric utility system, usually divided into three stages: generation, transmission and distribution. The electricity is transported by the transmission system over long distances, from the generation plants to the distribution system. The latter is in charge of retailing the energy and can be split into primary distribution systems – or medium-voltage networks – and secondary distribution systems – low-voltage networks. They both have complex topologies as opposed to the simpler transmission networks.

This work is focused on primary distribution systems. The networks used and an analysis of their topologies are presented in this section. The problem of visualizing the high volumes of data involved in a limited physical space is also considered.

2.1. Primary Distribution Network Datasets

Five real primary distribution networks from different regions over the world, selected to cover different populations and densities, have been analyzed:

- Two Central American countries: Nicaragua and Panama with low population densities.
- One Eastern European country, Moldova, which has a medium population density.
- Two different regions of Spain: Galicia, the northwesternmost area of Spain, formed by 4 provinces with a scattered, medium-density population; and Central Spain which comprises 9 provinces in the center of Spain, including Madrid – the province with the highest population density in the country – and most of the less densely-populated provinces.

Table 1: *Population density by region.*

Region	Population	Area (km^2)	Density
Galicia	2,783,100	29,574	94.11
Central Spain	8,856,615	115,777	76.50
Moldova	3,633,369	33,846	107.35
Nicaragua	5,677,771	130,373	43.55
Panama	3,394,528	75,517	44.95

Table 1 summarizes some population characteristics of the mentioned regions. The largest covered area has over 130,000 squared kilometers which should already be too large and too highly detailed to interact with through visualization in a CAD application.

Table 2: *Distribution networks datasets.*

Region	Knots	Branches	Nodes	Lines
Galicia	23,812	91,959	301,118	209,159
Central Spain	35,522	147,651	493,121	345,470
Moldova	59,726	43,769	160,717	116,948
Nicaragua	34,912	95,770	245,135	149,365
Panama	85,990	85,319	262,319	177,000

For each region, a dataset including the geographic and topological data of the network has been created. Networks are composed by knots – which are commonly substations – and branches linking them – typically power lines. Table 2 shows the different datasets, detailing the number of knots and branches for each network. Furthermore, the number of lines and nodes that compose the branches are shown for each dataset. Since knots are connected using branches, the geographic coordinates of the knots are the same as the coordinates of some of the nodes forming the branches. They provide no extra information from a topological point of view and thus, only branches are used in this work.

2.2. Topological Analysis

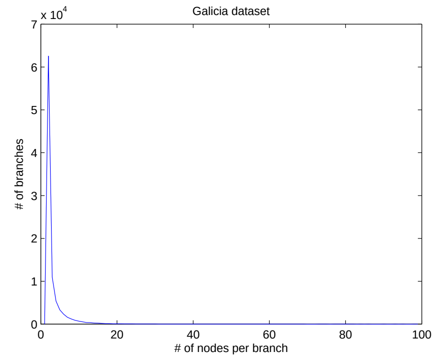
In this section, a topological analysis of the branches from Galicia and Central Spain datasets is presented. The analysis of the other datasets have yielded analogous results and have not been included for convenience.

Figure 1(a) shows the absolute frequency distribution of branches for the different amounts of nodes that compose them for the Galicia dataset. As it can be seen, most of the branches are composed by a small number of nodes. Figure 1(b), showing the detail of the frequency distribution for branches having more than 25 nodes, exhibits that the amount of branches with a high number of nodes decreases rapidly. Figure 2 shows how the Central Spain dataset exhibits analogous distributions. The average number of nodes per branch is 3.27 for Galicia and 3.34 for Central Spain.

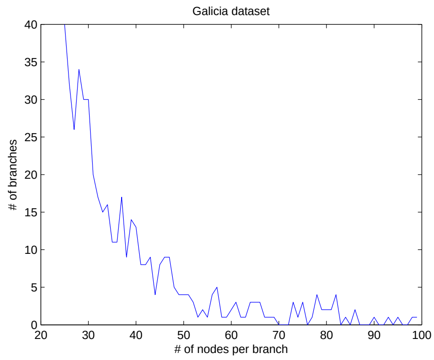
Table 3: *Percentage of branches formed by 2, 3 and 4 nodes.*

Dataset	2 nodes	3 nodes	4 nodes
Galicia	68.00 %	11.92 %	5.84 %
Central Spain	71.48 %	9.92 %	5.38 %

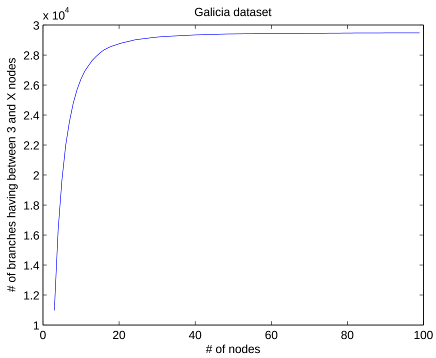
Percentages of the total number of branches having 2, 3 or 4 nodes are given in Table 3. For both datasets more than two thirds of the branches are composed by just 2 nodes – i.e. they are single lines. Furthermore, Figures 1(b) and 2(b) show that the number of branches with 25 and 35 nodes respectively is very low. Figures 1(c) and 2(c) corroborate this fact since it exhibits how fast the cumulative distribution of



(a) Absolute frequency distribution considering all branches.

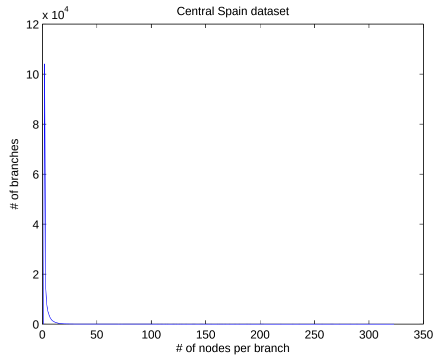


(b) Absolute frequency distribution considering branches formed by 25 or more nodes.

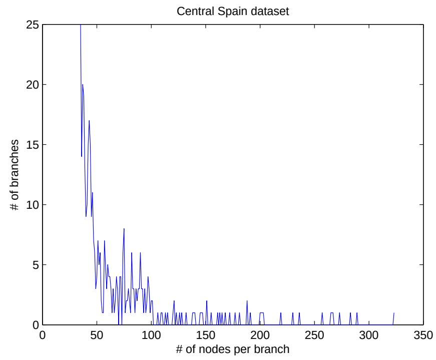


(c) Cumulative distribution.

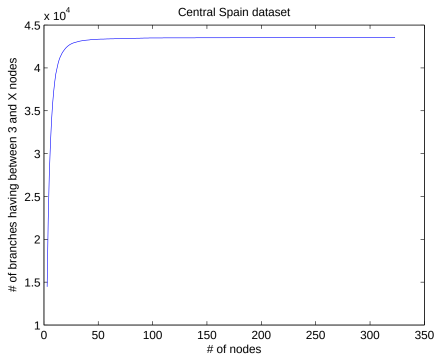
Figure 1: *Distributions of branches per number of nodes they contain for the Galicia dataset.*



(a) Absolute frequency distribution considering all branches.



(b) Absolute frequency distribution considering branches formed by 35 or more nodes.



(c) Cumulative distribution.

Figure 2: *Distributions of branches per number of nodes they contain for the Central Spain dataset.*

branches composed by 3 or more nodes seems to reach its asymptote.

This analysis reveals that the generalization must be focused on the branches composed by only 2 or 3 nodes since there may be little reward in performing complex optimizations over the longest branches.

2.3. Data Visualization Considerations

Whenever the grid is to be visualized on a screen or plotted on paper, the physical space available is very limited and thus, imposes constraints on the representation of the data that can be performed.

Figure 3 shows a very simple visualization of the whole Galicia dataset using unitary-width lines. This dataset contains branches ranging from a few meters up to 13 kilometers which are spread all over the northwestern-most region of Spain. The average length of a branch is 192 meters while the width of the Galicia area is 210 kilometers. When visualizing the whole region, there is no point in processing all the data since a big percentage of the branches will not even be visible – known as the imperceptibility problem – and those which are visible may not be easily discerned if there are many branches in their neighborhood – known as the coalescence problem. This is specially true for the cities, where there are hundreds of branches serving the streets which can

Table 4: *Points colliding in the same pixel of the visualization area.*

Dataset	Collisions	Points	% Collisions	Max. collisions
Galicia	258,285	301,118	86.15 %	331
Central Spain	468,750	493,121	95.06 %	2,991
Moldova	142,319	160,717	88.55 %	1,101
Nicaragua	217,637	245,135	88.78 %	295
Panama	232,122	262,319	88.49 %	309

not be distinguished when using a small scale. Indeed, as the figure shows, the visualization in the city areas is too cluttered because of branches coalescing.

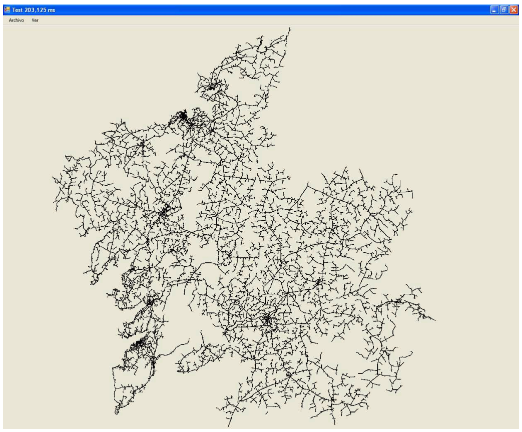


Figure 3: *Visualization of the Galicia dataset using unitary-width lines.*

In order to get an idea of how much redundancy is introduced by processing all the available data to visualize the whole region of a dataset, a matrix simulating a visualization area was created. The size of this matrix was the same as the screen area shown in Figure 3 – 1,440 pixels of width and 820 pixels of height – and thus having a total of 1,180,800 positions. Each position holds the number of times that a point, corresponding to the geographic coordinates of a node from the branches, has been painted on the screen in the corresponding pixel.

Table 4 shows for each dataset how many times a pixel had more than a point assigned (collisions), the total number of points that compose the branches, the percentage of points that end up colliding with others in the same pixel and the maximum number of collisions for a single pixel. Between a 86.15 % and a 95.06 % of the points fall in the same physical position of the visualization area. Even more, in the case of the Central Spain dataset, there is one pixel of the screen which ended up being painted 2,991 times – which means that 2,990 node computations and coordinate translations could have been omitted just for that pixel. By skipping the processing of all those nodes in the first place, a big percentage of the computations required to visualize the data can be avoided, resulting in a faster rendering of the visualization.

3. Cartographic Generalization Techniques

Cartographic generalization is the process of abstracting the representation of geographic information to match the requirements of scale of a map. As a result of this process, different maps carrying different levels of detail are generated for different scales. There is no standard or unified methodology when it comes to cartographic generalization, the process depends on the exact requirements of the maps and their purposes as well as the final users and their knowledge. McMaster and Shea analyzed the factors, conditions and requirements involved in digital cartographic generalization. They divide the generalization process into three fundamental questions: why it may be necessary to generalize a map [MS88], conditions that define when to do it, and how to do it – by using operators that perform the different techniques of generalization [SM89].

As the authors suggest, cartographic generalization helps counteract the undesirable consequences of scale reduction. This work is mainly focused on reducing the complexity without incurring neither spatial accuracy nor aesthetic quality losses. As for the conditions, congestion and imperceptibility are the more dominant forces in this work. Congestion refers to the fact that upon scale reduction too many geographic features need to be represented in a limited physical space on the map. This is the case of geographic points colliding in the same physical pixel, seen in Section 2.3. Imperceptibility occurs when some features of the map are not optically legible for some reason, for instance when visualizing a large region, a branch which is only a few meters long will fall below the minimal portrayal size of the map at that scale. Another condition related with imperceptibility is coalescence. In this case the features can be represented in the map but they are too close or in some kind of juxtaposition with other features, making their area of the map too clogged. This would be the case of the cities when the map is being visualized using a small scale. In order to overcome these problems, simplification and merging operators are used as described in the following sections.

3.1. Selection Process

Prior to the application of the generalization operators, the data over which they will operate must be selected. During this selection it is desirable to eliminate as much irrelevant data as possible.

The most basic requirement for the data is to be optically visible in the final visualization. Thus, any imperceptible feature found in the data will be marked as irrelevant and it will not be selected. To check whenever a feature will be seen in the visualization, its size in meters is compared to the number of meters that each pixel of the visualization area accounts for. In order to do this, it is required to know the resolution of the data – normally in meters – and the resolution of the visualization area – in pixels. Using these numbers, the *meters per pixel* relation (referred to as *mpx* from now on) for the given visualization and scale resolutions is calculated. Any branch from the power grid having a length smaller than the *mpx* value is discarded and therefore not selected to be processed by the generalization operators. In the case of branches formed by more than one line, its bounding box dimensions are checked instead of adding the individual lengths of the lines. The bounding box must be either vertically or horizontally equal or larger than the *mpx*.

3.2. Line Simplification

Line simplification produces a reduction in the number of nodes composing a branch without modifying their coordinate positions. This reduction in the number of nodes will provide an overall increase in the performance since less processing is required. It must be noted that in the simplification context, the term line refers to the *LineString* GIS datatype which consists of a set of interconnected line segments [OC99]. This term is analogous to the branches of the power grids which are formed by interconnected power lines.

In this work, the Ramer-Douglas-Peucker algorithm was used to perform the simplification. This algorithm operates by discarding the nodes of the branch that are not significant based on a certain threshold – the *mpx* measurement in this case, since anything smaller is visually imperceptible. Starting with the straight line formed by the beginning and end nodes, the algorithm selects the node with the largest ortogonal distance to that imaginary line. As long as that distance is larger than the threshold, it recursively calls itself for the lines formed by the beginning and the selected nodes and by the selected and the end nodes. Once there are no nodes left or no more nodes with a distance to the corresponding line larger than the threshold, the algorithm returns the simplified branch formed by the original start and end nodes and all the selected nodes [DP73].

3.3. Merging

Aggregation – understood as the general process including also merging and amalgamation – consists on joining map objects on the basis of their proximity and the objective is generally a reduction of the spatial resolution [Orm96]. This is needed not only for the symbols to remain legible but also to avoid excessive accumulation of symbols in small areas which is the motivation for its use in this work. In the case of electrical distribution grids which are composed of branches, the working units are the two-dimensional lines that form those branches and the process is known as merging. The operations carried over the data in this work are two:

1. Merge branches that share their beginning and end nodes. Since the resulting merged branches can share their beginning and end nodes with other branches too, this is an iterative process. As a result, all the branches which can be connected become a single branch, saving nodes which would need to be processed several times otherwise. As a side effect, the merged branches become eligible for line simplification since they are composed of more nodes. During this process, the logical organization of the lines change but their geographic coordinates remain the same.

2. Merge branches which are too close to be individually distinguished. This is also an iterative process which operates on two branches at a time and merges them into a single branch formed by the equidistant points to those of the original branches. As a result, the geographic coordinates of the nodes change and thus, merges must be carefully overseen to avoid distorting the aesthetics and accuracy of the map. This process needs a neighborhood parameter that determines up to how far away from a branch to look for its neighbors. The bigger this parameter is, the more branches will be merged and more aesthetically noticeable the process will result.

The first kind of merge reduces the overall number of nodes by saving the redundant ones while the second one eliminates entire branches since two whole branches are merged into one. It should be noted that while line simplification operates within branches, both merge operations work at a higher level.

4. Implementation

In order to perform the generalization process and to store its results, an approach based on a multi-scale database architecture was used. Multiple representations of the spatial data are stored using different resolutions – i.e. scales – and a set of rules are applied to support the generalization decisions, selecting the appropriate representations, governing updates and maintaining database integrity [Jon91]. A database has been created for each dataset using PostgreSQL an its spatial extension PostGIS. In each database, a base table is created to keep the electrical distribution grid topology in its original scale. The generalization operators have been implemented as PL/pgSQL stored procedures.

The generalization is an offline process fired by a procedure taking the *mpx* for the desired scale resolution and according to the visualization area resolution. The neighborhood parameter for the merging process is also required. The value of this parameter is used as a coefficient of the *mpx* to calculate how far around the branch to look for its neighbors. Thus, the value of the neighborhood parameter corresponds to the maximum number of pixels between branches considered as neighbors in the visualization – the bigger this parameter is, the more aesthetically noticeable the process becomes. The result is a table containing the data generalized for the scale. The generalization procedure invokes the selection, line simplification and merging procedures implementing the operators described in Section 3. These procedures make intensive use of the spatial indexing system available in PostGIS [Gut84], especially the merging operator which uses it to find the branches intersecting with every branch and its neighbors. In electrical power grids, the same physical line can be considered as two different branches composed of the same nodes but with different directions. Thus, the first time that intersecting lines are found, a check is made to find whenever some of them are equivalent. If they are, only one of them will be kept and all the equivalent ones will be removed.

Table 5 shows the time required to perform the generalization process over each dataset for both 2 and 3 times the *mpx* as the neighborhood parameter. The *mpx* was calculated using a 1:1 scale and a visualization resolution of 1,440x820 pixels. PostgreSQL 8.4 and PostGIS 1.5.1 were used for the implementation, running on Windows XP SP3 in a Intel Core2 Q6600 2.4 GHz CPU machine with 2 GBs of memory. The bigger the neighborhood factor is, the more time the merging process will take since more branches will be included. This overhead accounted for a 4.5 % average time increase in the tests. These results make it obvious that the process can only be performed offline and therefore the results must be stored so they can be retrieved by the CAD application without incurring in delays.

Table 5: *Time consumed by the generalization process.*

Dataset	**Branches**	**Neighborhood**	
		2x *mpx*	3x *mpx*
Galicia	91,959	427s	430s
Central Spain	147,651	142s	158s
Moldova	45,769	294s	305s
Nicaragua	95,770	100s	101s
Panama	85,319	52s	55s

In order to execute the multi-scale generalization, the number of scales to be generated and the resolutions of both the visualization and the minimum desired scale are required. Using these parameters, the corresponding *mpx* for the different scales can be easily calculated and thus all the generalizations can be made in batch. Each generalized scale is stored in a dedicated table which is kept at sync with the base table using trigger functions. This way, each time the original data is updated, the tables containing the multi-scale generalizations are also updated accordingly. When the grid data is requested to the database, a two-dimensional box is used to select the data. The dimensions of this box determine the resolution of the desired scale which is used to choose the table holding the proper generalized scale.

The components of the implementation are the following:

- Stored procedures library: contains the procedures implementing the generalization process, the different techniques it uses and the batch multi-scale generation producing the tables with the different generalized scales.
- Multi-scale tables: contain the results of applying the generalization process to the base table to obtain the different desired scales.
- Trigger functions: in charge of keeping the multi-scale tables synchronized with the base table so that any change in the power grid data is reflected in all the generalized scales.

Both the generalization process and the multi-scale generation procedures require some parameters which are:

- Generalization process: requires the *mpx* and the merge neighborhood factor. Based on the experimentation, typical values of the neighborhood factor are 2 and 3 times the *mpx* which correspond to 2 and 3 pixels of proximity respectively. Obviously, the bigger this parameter is, the more branches are merged and the more noticeable the process becomes in the visualization.
- Multi-scale generation: requires the resolutions of both the smallest desired scale and the visualization, the number of intermediate generalized scales to be generated and the generalization neighborhood factor. The different generalization *mpx* values are automatically calculated by dividing each scale resolution by the visualization resolution. These values are then passed along the neighborhood factor as the parameters of the corresponding batch generalization process invocations.

5. Results

The process described was executed to generalize the different datasets in the most intensive scenario: generalizing the topologies using the same scale resolution of the original data – i.e. a 1:1 scale – but restraining the visualization resolution to 1,440x820 pixels. This results in the whole datasets – some of them covering entire countries – being displayed in the visualization area which corresponds to one of the smaller scales that may be required. As the user zooms into the visualization, a smaller area of the power grid is displayed requiring a larger scale. In that case, while the detail of the area is higher, both the data volume and the generalization required are smaller.

Table 6 shows the results of generalizing the different datasets for a neighborhood factor of 3. *1st Merges* column refers to the number of branches created from merging those that share their ending and beginning nodes while the *2nd Merges* refer to the number of new branches averaging a pair of old ones. The iterations required to complete each kind of merge are also shown. The generalized network contains the 10.76 % of the original data on average and thus, almost a 90 % of the data is skipped and will not be processed.

The same process with a neighborhood parameter of 2 times the *mpx* is shown in Table 7. Comparing to the previous results it can be seen that using a factor of 2 instead of 3, yields an average of 27.66 % less *2nd Merges*. The number of *1st Merges* does not change since they are not affected by the neighborhood parameter. The generalized version contains in this case the 12 % of the original data, slightly more than the 10.76 % yielded by the use of a neighborhood factor of 3. The decision about which factor value to use depends mostly on the aesthetical accuracy requirements since the bigger the factor is, the more noticeable the process becomes in the visualization. Furthermore, using a bigger neighborhood factor increases the time required by the generalization process since more merges are performed. As seen in Table 5, this overhead accounted for a 4.5 % average time increase when using 3 instead of 2 times the *mpx* as the neighborhood parameter.

The multi-scale database approach increases the storage requirements since the results of the generalization are stored in dedicated tables. In these tests, one scale was generated and thus, one new table was created to store each generalized dataset. The required disk space was increased an average of 25.54 % and 26.52 % for the different datasets when using a neighborhood factor of 2 and 3 respectively.

Table 6: *Number of branches as a result of the generalization process with a neighborhood factor of 3.*

Dataset	Original	Generalized	1st Merges	2nd Merges
Galicia	91,959	13,767	8,960 / 5 iters	2,120 / 5 iters
Central Spain	147,651	8,660	4,387 / 4 iters	3,035 / 7 iters
Moldova	43,769	9,012	6,049 / 4 iters	1,090 / 5 iters
Nicaragua	95,770	6,482	5,550 / 6 iters	900 / 4 iters
Panama	85,319	4,777	3,632 / 5 iters	831 / 6 iters

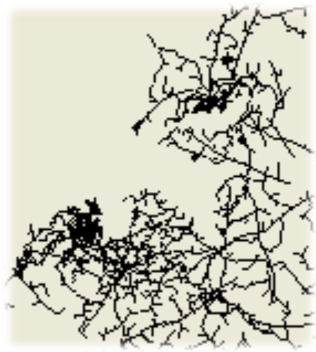
Table 7: *Number of branches as a result of the generalization process with a neighborhood factor of 2.*

Dataset	Original	Generalized	1st Merges	2nd Merges
Galicia	91,959	15,253	8,960 / 5 iters	634 / 4 iters
Central Spain	147,651	10,616	4,387 / 4 iters	1,079 / 7 iters
Moldova	43,769	9,806	6,049 / 4 iters	296 / 4 iters
Nicaragua	95,770	7,198	5,550 / 6 iters	184 / 3 iters
Panama	85,319	5,398	3,632 / 5 iters	210 / 5 iters

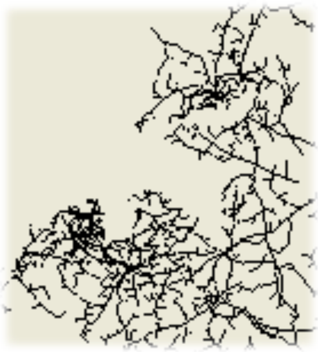
Table 8 exhibits the times required to render the visualization of the non-generalized and generalized datasets using the environment described in Section 4 and a NVIDIA GeForce 8500 GT GPU. The generalized visualizations using a neighborhood parameter of 2 and 3 need respectively an average of only a 17.96 % and a 14.76 % of the time required to visualize the non-generalized datasets. The performance has been increased approximately sixfold as a result of the great reduction of the data accomplished by the generalization. Furthermore, the visualization can be rendered in real-time – i.e. it can be rendered more than 15 times per second. For instance, using a neighborhood parameter of 3 times the *mpx*, the slowest rendering took 62.50 milliseconds – which accounts for 16 frames per second – while the quickest required 31.25 milliseconds – 32 frames per second.

Table 8: *Time required to render the different generalizations (in milliseconds).*

Dataset	Non generalized	Neighborhood	
		2x *mpx*	3x *mpx*
Galicia	343.75	70.32	62.50
Central Spain	521.25	54.69	46.88
Moldova	187.50	54.69	46.88
Nicaragua	296.88	46.88	31.25
Panama	281.26	39.07	31.25



(a) Non-generalized visualization.



(b) Generalized visualization.

Figure 4: *Comparison of generalized and non-generalized visualizations of a metropolitan area.*

A comparison of the generalized and non-generalized visualizations of a metropolitan area from the Galicia dataset is shown in Figure 4. This area covers two cities with a combined population of about half a million inhabitants over approximately 376 km^2. Figure 4(a) shows a detail of the non-generalized visualization where the cities areas appear cluttered. These areas are cleaner in Figure 4(b), generalized using a neighborhood parameter of 2 times the *mpx*.

6. Conclusions

This work presents the results of applying several digital cartographic generalization techniques to improve the performance of the visualization done by an Electrical Power Grids CAD application. Specifically, electrical distribution grids have been generalized using spatial databases to reduce the high volume of geographical and topological data of these networks. The tests run over the presented datasets exhibit an average reduction of almost the 90 % of the data. This reduction greatly improves the visualization times of the application since there is less information that needs to be processed and rendered by the graphics card. Using the generalized datasets yielded a 84 % average decrease in rendering times – being always less than a tenth of a second and thus, enabling real-time rendering and interaction in the CAD application.

A multi-scale architecture has been implemented using PostgreSQL and its spatial extension PostGIS. Using PL/pgSQL stored procedures, the data is generalized and stored in different tables corresponding to the desired generalized scales. The process uses selection, simplification and merging graphic generalization techniques to solve the congestion, imperceptibility and coalescence conditions found in the data. Since all the heavy processing involved in the generalization is done offline and the overhead from choosing the right table for the desired scale upon retrieval is negligible, the data is greatly simplified without any computational costs – only storage needs are increased proportionally to the number of generalized scales. Hence, the performance of the visualization is boosted since the volume of data to be processed and rendered is much smaller. Results show that the performance increase was approximately sixfold while the extra disk space required to store a single generalized table accounted for an average increase of a 26 % for all the datasets.

The generalization process has an aesthetical impact on the visualization as a result of the data reduction. The behavior of the generalization is controlled by parameters that can be adjusted to find the proper tradeoff between visual accuracy and data reduction. The specific settings must be chosen depending on the exact requirements of the visualization: higher values of the parameters for better optical legibility and faster rendering times or smaller values for maximum visual accuracy.

The architecture and procedures presented in this work should be eligible to be seamlessly implemented in most modern spatial databases or even using ad-hoc libraries, making the described generalization techniques perfect candidates to improve a great range of applications involving high volumes of spatial data beyond electrical distribution grid visualization.

References

[DP73] DOUGLAS D. H., PEUCKER T. K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization 10*, 2 (1973), 112–122.

[Gut84] GUTTMAN A.: R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD international conference on Management of data* (1984), ACM, pp. 47–57.

[Jon91] JONES C. B.: Database architecture for multi-scale GIS. In *Autocarto-Conference* (1991), vol. 6, ASPRS American Society for Photogrammetry, pp. 1–14.

[MS88] MCMASTER R. B., SHEA K. S.: Cartographic generalization in a digital environment: a framework for implementation in a geographic information system. In *GIS/LIS'88: proceedings: accessing the world: third annual International Conference, Exhibits, and Workshops, San Antonio, Marriott Rivercenter Hotel, San Antonio, Texas, November 30-December 2, 1988* (1988), American Society for Photogrammetry and Remote Sensing, pp. 240–249.

[NRCCHP10] NOVO RODRÍGUEZ J., CABRERO CANOSA M., HERNÁNDEZ PEREIRA E.: Electrical distribution grid visualization using programmable GPUs. In *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on* (2010), IEEE, pp. 1231–1235.

[OC99] OPENGIS-CONSORTIUM: Open GIS Simple Features Specification for SQL. Revision 1.1. *OpenGIS Project Document 99-049* (1999).

[Orm96] ORMELING F.: Aggregation objectives and related decision functions. *Methods for the Generalization of Geo-Databases. GEODESY: Delft, Netherlands* (1996), 1–11.

[SM89] SHEA K. S., MCMASTER R. B.: Cartographic generalization in a digital environment: When and how to generalize. In *Proceedings for Auto-Carto* (1989), vol. 9, pp. 56–67.