

VisGL: an Online Tool for Visualization of Bivariate Georeferenced Data

Tarsus Magnus Pinheiro¹, Claudio Esperança¹

¹PESC – COPPE – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brazil

tarsus.pinheiro@ibge.gov.br, esperanc@cos.ufrj.br

Abstract. *This paper describes an online interactive thematic map for simultaneously visualizing two scalar variables and which supports filtering, configurable category classes, as well as panning and zooming in levels of detail. The user experience is improved by means of queries posed through manipulation tools that produce instant responses at screen. This is possible through the high rendering rates get by the system, that uses GPU programming to assemble and manipulate previously rasterized tiles with location information recorded in the color space of pixels. This procedure allows the implementation of interactive animated actions and spatial data decomposition.*

1. Introduction

Data visualization studies the forms of visual communication that relate to the registration and organization of data, so as to use vision to reveal hidden characteristics, patterns and trends in massive amounts of data [Gershon et al. 1998].

Georeferenced data are data related to the geographic space. Their visualization has gained relevance with the popularization of map systems for the Internet, which have turned the use of maps into an everyday practice. Thematic maps are objects whose main function is to express geographical information by means of graphic signs designed to present relations of data similarity, ordering and quantification [Archela and Théry 2008].

Choropleth maps are a specific type of thematic map that use color as a graphic sign to communicate data relations. These types of maps are usually static and depict only one variable at a time. The system we have created allows the interactive online display of maps of that kind and supports an arbitrary number of regions with two variables plotted simultaneously, allowing inspection in levels of detail. The visual mapping of the two variables uses color and texture scales and allows interactive adjustment. The system is based on the assembling of a group of previously processed raster images, and adopts Graphics Processing Unit (GPU) programming to reach a speed high enough to provide instant responses to user action, thus improving one's learning experience from the visualization of simultaneous variables. The contributions of this work can be summarized as follows:

1. Allows online rendering of bivariate choropleth maps with adjustable categories.
2. In addition to the traditional visual mapping that strictly observes the region boundaries of the original data, it is possible to compute on-the-fly mappings based on the idea of statistical grids.

3. All data structures are encoded as binary images, which are compact and make efficient use of browser and server caches, as well as using little network bandwidth.
4. Other than requesting static data images from the server, all of the system's processing is done on the client side, thus requiring very little use of server resources.
5. Dynamic image composition uses GPU programming, which leads to high frame rates.
6. A fully functional prototype is available online.

Next section presents some related works about multivariate data visualization. The system operation and implementation is presented in section 3. Section 4 shows some results and in section 5 we discuss the conclusions.

2. Related Works

The visualization of multivariate data is a relevant challenge. Examples such as the Chernoff Faces [Chernoff 1973], glyphs similar to stick figures [Pickett and Grinstein 1988], the use of parameterized naturalistic textures [Interrante 2000] and simulations of impressionist paintings in which the characteristics of brushstrokes are summarized according to the values of associated variables [Tateosian et al. 2007] show different ways of dealing with the theme. A common problem for all of them is the difficulty to represent quantification, as already discussed by Bertin [Bertin 1980]. He states that size is the only variable capable of representing proportion relations, and that other graphic signs are limited to simple visual representations, such as differentiation and ordering of categories.

One more issue in common among the cited methods is that none of them presents interactivity as a way to improve user experience. Dynamic queries, defined as the interactive control of filters over a group of data that produces an immediate visual reply in a time span of less than 100 milliseconds [Shneiderman 1994] deal with the importance of interaction in the learning process. That brings users closer to the data and turns the formality of database queries into intuitive actions to help the user to try data and improve their search for patterns that reveal new points of view about a given amount of information.

That idea counters the attempts at creating complex signs to promote multivariate visualization, being closer to the proposals of traditional cartography, where the principles set by the Semiology of Graphics [Bertin 1980] established a safe starting point for the elaboration of thematic maps. The application of technologies that improve the quality and speed of rendering, and, as a result, the experience of visualization, should enhance the result of traditional methods.

Choropleth maps are a type of cartogram which represents figures grouped into classes associated with a chromatic scale [Archela and Théry 2008, p. 8] and are the right tool for the visualization of data associated with political-administrative divisions. Creating a map like that requires correct color selection [Stone 2006], and adequate methods for the organization of classes [Andrienko 2001]. [Newman 2012] discusses the use of such maps for the visualization of data relative to American elections of 2010 showing how the changes geographic subdivisions, in color scale and in the application of cartographic anamorphosis can affect data representation, and, as a result, the assimilation of information.

Political boundaries are not the only possible representation for choropleth maps. Statistical grids are a method of cartographic visualization that has become quite popular recently and establishes an arbitrary division of geographic space as a grid formed by squares that rasterize the plane and redistribute information according to specific requirements [Bueno 2014, IBGE 2015]. As some advantages of this option we can mention formal stability, once this subdivision is not subject to the history and the setting of political-administrative boundaries [IBGE 2010], and the best visualization of regions which might not be visible at more detailed levels of display. It is also worth mentioning the fact that data result, many times, from spatial decomposition and that can lead to distortion [Bueno and D'Antona 2014].

3. System Description

The ideas and concepts proposed in this work were implemented in a proof-of-concept prototype which is readily available online in the following address: <http://www.tarsusmagnus.com.br/mestrado/visgl/index50.html>. Interested readers can try the prototype which only requires a modern browser and a good internet connection. The next sections explain the main ideas of our proposal and how they were assembled in our prototype.

3.1. Data sets

In our experiments, we have used the 2013 Municipal Grid [IBGE 2013] and selected, from the IBGE's channel Cities, 140 socioeconomic variables originated from count processes, such as the Population count, the municipal vehicle fleet, and the number of votes in the second round of the 2014 Presidential Elections, with information about all 5570 Brazilian municipalities.

3.2. Interface

The work proposed herein considers the visualization possibilities provided by the statistical grid. We use an instance of a real time decomposition based on statistical information associated with regions, in order to show a representation where abstract limits are derived from the classification of data, adapted and changed by user interaction.

Figure 1 shows the system interface, which is divided into a visualization panel on the left and a set of graphical user interface (GUI) elements on the right. These GUI elements control several aspects of the visualization and the selection of the two scalar quantities to be visualized. Each such quantity or "theme" is a rate defined as a fraction between two socioeconomic variables selected from the data sets.

3.3. Architecture and Data Structure

The system is implemented in HTML5, that is the latest version of the markup language and content organization that is the basis for crafting pages for the Internet and was created to has met the growing demands of more elaborate multimedia content such as sound, video and pretty elaborate images, and applies JavaScript and WebGL to enable GPU (Graphics Processing Unit) programming so as to reach faster rendering directly in standard web browsers.

The Javascript is a scripting language based on the technologies available on the Internet that allows manipulation of the Document Object Model (DOM) generated by

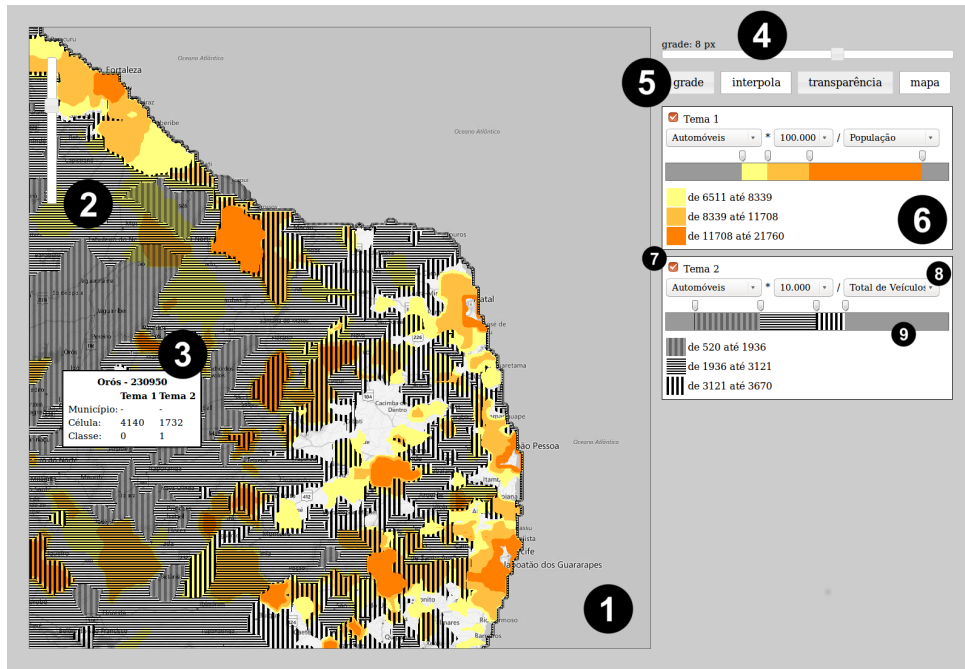


Figure 1. System interface. 1. Area for map rendering where panning is realized by click and drag. 2. Zoom slider to change the level of detail. 3. Box with information about the cell; 4. Slider controlling the size of visualization cells; 5. Toggle buttons controlling grid display, visualization smoothing, theme transparency, and background map. 6. Theme control panel. 7. Theme on/off button. 8. Theme variable selection. 9. Multi-slider controlling category class ranges.

browsers from the organization arranged in an HTML page, thus enabling dynamic editing of its content. WebGL, in turn, is a Javascript API that allows manipulation of the HTML5 canvas element, thus providing support for rendering 2D and 3D graphics. That is possible due to the architecture of modern graphics hardware, which is organized around rigid flow of operations that only allow interference from specific GPU programs called shaders. WebGL supports two types of these: vertex shaders, responsible for the processing of each vertex of the object's geometry; and fragment shaders, which process each pixel of the image output by the rasterization of the graphics primitives. The normal output of this flow is the computer screen, but it is possible to record the result in memory buffers so that it will return to the flow and be reprocessed aiming at more complex operations. This technique is called "render to texture" and the proposed system uses it extensively to process the map.

Vector graphics files are appropriate for the elaboration of choropleth maps and can be implemented for the Web by using the SVG language. The SVG is also a markup language like HTML that specifically allows the creation of animated vector graphics. It was incorporated by HTML5 and started to function natively in modern browsers. However, maps converted into vector files usually result in very detailed polygons, very often with more than a million vertices. Raster images are also common, and most popular map services work using raster images. However, raster maps are frequently more costly in terms of bandwidth and memory. For more information about these technologies, see

[(W3C) 2017, Tutorials 2017, Inc 2017].

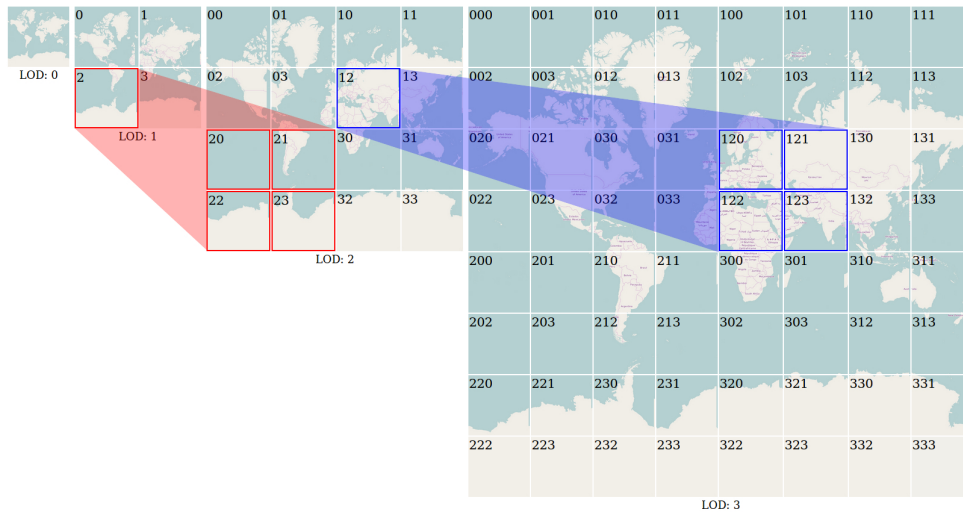


Figure 2. World map tiles organized by LOD code strings, that have important properties: total map size is duplicated at every new LOD, whose level is represented by the string code length and its last character – 0, 1, 2 or 3 – sets the new tile’s relative position.

As the system uses GPU programming directly, it was possible to implement it based on raster images. In order to do so, the images need to be processed previously, so that location information (municipal codes, in our case) is recorded in the color space of each one of its pixels, thus making it possible to associate the map and its data.

The map displayed on the screen is an assembling of smaller pieces cut from bigger maps, previously rasterized at several levels of detail, as explained in [Schwartz 2016]. Each level of detail is composed of a set of square image tiles forming a pyramid starting at level 0 - a single tile – with twice that size at each new level until it hits the maximum level established for the system, which is 32. The dimensions ($mapWidth \times mapHeight$) change according to the level of detail (lod) and are defined by: $mapWidth = mapHeight = 256 \times 2^{lod}$. It is possible to project a point, (lat, lng), on a pixel, (x, y), like in:

$$x = \left(\frac{lng + 180}{360} \right) \times mapWidth$$

$$y = \left(0.5 - \frac{\log \left(\frac{1 + \sin lat}{1 - \sin lat} \right)}{4 \times \pi} \right) \times mapWidth$$

The selected images, or tiles, are squares sized 256×256 organized into a $m \times m$ matrix, where $m = 2^{lod}$, and where the position of a tile is defined by two indices ranging from (0, 0) to ($m - 1, m - 1$). That way, the tile that contains pixel (x, y) is at $\left(\lfloor \frac{x}{256} \rfloor, \lfloor \frac{y}{256} \rfloor \right)$.

Tiles are identified by Level-Of-Detail (LOD) codes, which are strings composed according to a combination of matrix coordinates. Those codes have important character-

istics, since they identify the level of detail of a tile as well as its position in the hierarchy of the tree.

The size of the screen determines the quantity of tiles and the center can be projected in the space of pixels of the map. So, to load the images and form the screen, one can simply compute the corresponding level of detail codes. The adjustment of position (panning) is done with the displacement vector obtained from the division that defines the tile and the moving of the map consists in changing the center in sequence. The change of level of detail (zooming) only requires loading of the respective *lod* tiles (see Figure 2).

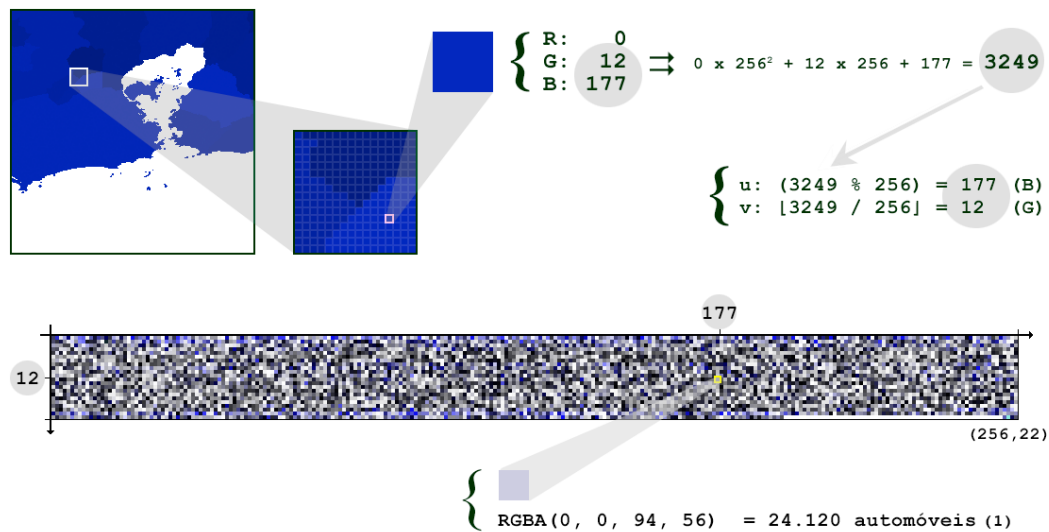


Figure 3. Example of image encoding of socioeconomic variables - in this case, number of cars. Total the municipality corresponding to the highlighted pixel: 24120 cars (1).

The system performs the visualization of the Brazilian territory and, because of that, it requires precomputed image tiles to cover it from level 4 – enough to display Brazil on a 1024×1024 pixel screen – to level 10 – enough to allow visualization of the smallest municipality in the country, Santa Cruz de Minas (MG) [IBGE 2016].

The system's interface requires the assembling of two layers of tiles, a background image which serves as topographic reference and another one containing the location images of the same color of the pixels, where index values are responsible for the association of data and the regions. The RGBA channels of raster images are integers ranging from 0 to 255, allowing the formation of 256^4 (4294967296) color possibilities.

Besides the georeferenced tiles encoding municipalities, the values of the 140 socioeconomic variables are also encoded as images. Each such variable is stored as an image where each pixel encodes the value of that variable for a given municipality. This image has a width of 256 pixels – a convenient value since it is the size of a color channel. In order to store the 5570 municipal identifiers, an image of height $\lceil \frac{5570}{256} \rceil = 22$ is necessary. The position of a pixel in that image is defined by the municipality index, an integer number between 0 and 5569. Thus, if the identifier of a municipality is stored in

a two-byte integer, the first byte corresponds to the row and the second to the column of the pixel in the image. In fact these two bytes are exactly those encoded in the blue (B) and green (G) channels of a tile pixel (see Figure 3).

Another relevant piece of information that must be encoded in images is the size, in pixels, of each municipality for every level of detail. This is necessary to compute the actual contribution of that municipality per unit area, or rather, per pixel. These contributions are averaged to obtain the value of a given variable within each cell of a grid used for the visualization. A visualization cell is a square group of pixels ranging from 1 to 32 pixels wide used to compose the visual representation of the map. The size of the visualization cells is chosen by the user. A cell of unit size will result in the traditional choropleth visual mapping, where the color of a pixel depends solely on the value of the variable for the region containing that pixel. Larger cells require averaging over all pixels within the cell, allowing a visual decorrelation from the political boundary of each municipality, as suggested by the research on statistical grids (e.g. [Bueno and D’Antona 2014]).

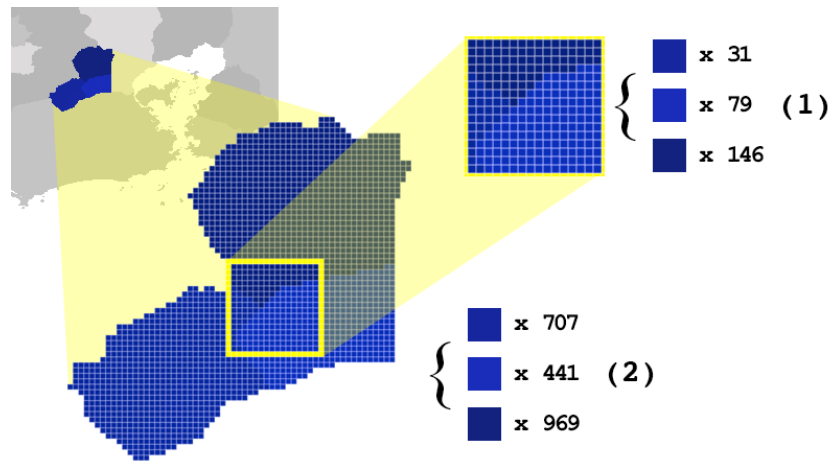


Figure 4. Data decomposition by pixel and recomposition by visualization grid cell. The values shown in (1) represent how many pixels of each municipality are present in this 16 x 16 visualization cell. Values shown in (2) correspond to the total number of pixels of each municipality present in the current level of detail.

An example is shown in Figure 4, where a 16×16 visualization cell covers 3 municipalities represented with 31, 79 and 145 pixels, respectively. Since these municipalities are represented at that level of detail with 707, 441 and 969 total pixels, the average for the cell is given by $\frac{1}{256} \left(v_1 \frac{31}{707} + v_2 \frac{79}{441} + v_3 \frac{145}{969} \right)$, where v_1, v_2, v_3 are the values of the variable in question for each municipality.

3.4. Rendering

All of the heavy lifting of the system is performed in a series of steps implemented by fragment shaders. The process starts by estimating the average value per visualization cell of each socioeconomic variable required by the visualization selected by the user. This follows the process outlined in the previous section. Up to four variable values might be estimated this way, since the visualization itself is a composition of two “themes”, each expressed as a rate between two variables, e.g., cars by population, schools by spending

budget, etc. The values for each theme are classified for display in up to three categories using adjustable scales (see panel 6 in Fig. 1) and, for this reason, minimum and maximum values over all visualization cells must be computed. This is done using a GPU technique called *parallel reduction* [Fernando 2004].

Once the value of the theme for each cell is known, the second step consists of their classification according to the theme's category scales and a suitable visual encoding of the cell can be produced. Cell values falling outside the established categories are disregarded, i.e., rendered with a transparent color/texture. If desired, an additional step can be triggered whereby grid cells are smoothed and interpolated.

The objective here is to promote the organic cutting of the image which, in association with the speed of rendering, results in an animated effect on the screen that facilitates the visualization of peaks and valleys in data (Figure 5).

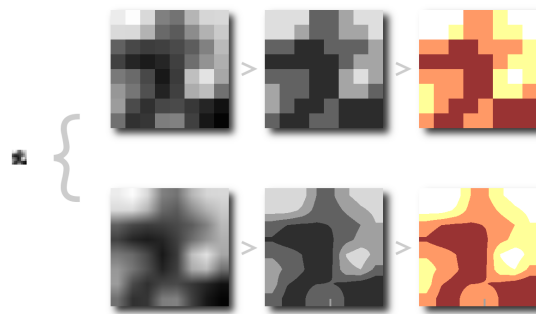


Figure 5. Data classification and optional smoothing.

Finally, the visual encodings of both themes are produced and blended. The first theme is encoded as solid colors, whereas the second theme is encoded as black-and-white striped textures of varying thickness and orientation. This way, a simple blending using color multiplication of the two produces the final result.

The entire process restarts at every new triggered frame. Since rendering is instantaneous, the result is animated transitions according to user actions. Figure 6 illustrates the complete process.

4. Results

In order to evaluate the system, we have conducted two case studies. The first one uses vehicle fleet data. Figure 7 shows the situation of cars in relation to total population and to total motor vehicles. In both cases, it is easy to see that the South and Southeast Regions concentrate the largest proportion of cars, probably because they concentrate Brazil's richest areas. Figure 8 shows the same regarding motorcycles, presenting the total quantity in relation to total population and to total motor vehicles. Here it is possible to observe that, differently from cars, motorcycles are more common in the North and Northeast Regions of Brazil, much probably because they are easier and cheaper to assemble and distribute at these parts of the country.

In the second case, we selected three variables about 2014 Brazilian second round of presidential elections: Total votes for Dilma Rousseff, Total votes for Aécio Neves

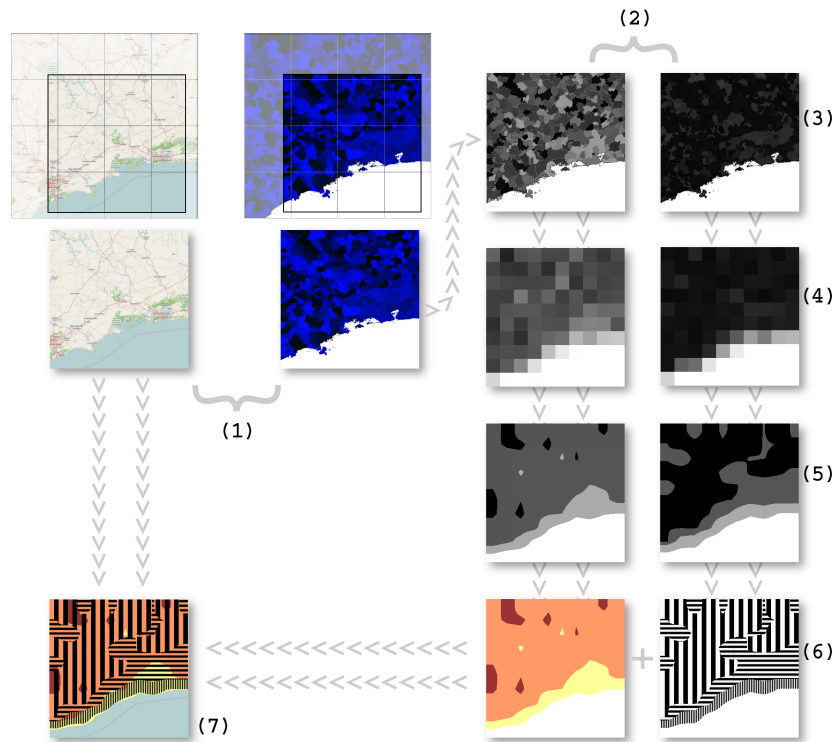


Figure 6. Functional flow: set the tiles (1); themes (2); index setting (3); grid (4); data interpolation and classification (5); visual signs (6); and blending (7);

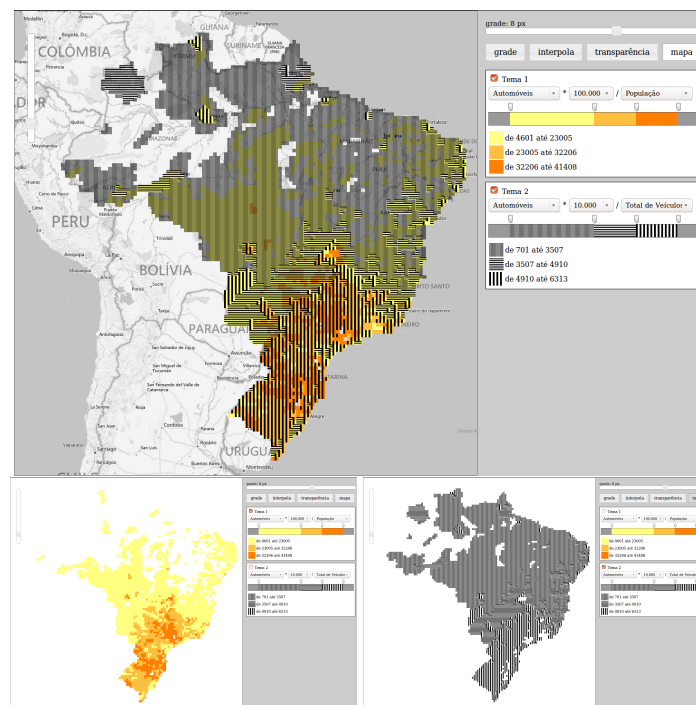


Figure 7. Cars for every 100 thousand inhabitants and for every 10 thousand vehicles. The smaller images present the themes separately.

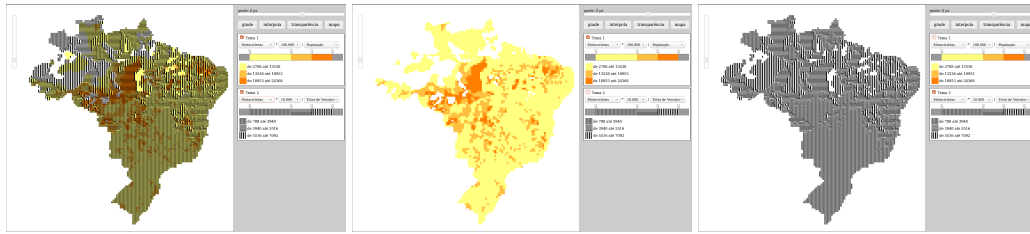


Figure 8. Motorcycles for each 100 thousand inhabitants and for each 10 thousand vehicles.

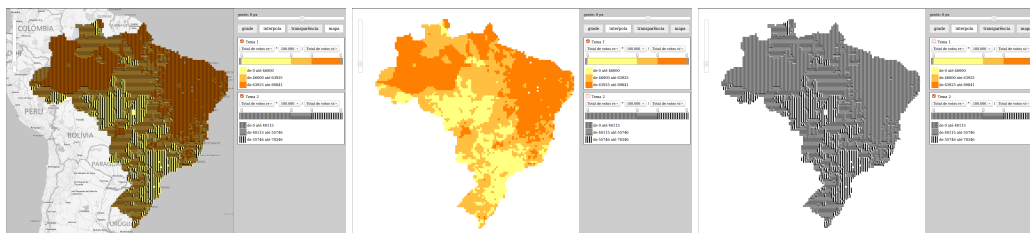


Figure 9. Status of the second round of the Brazilian presidential elections.

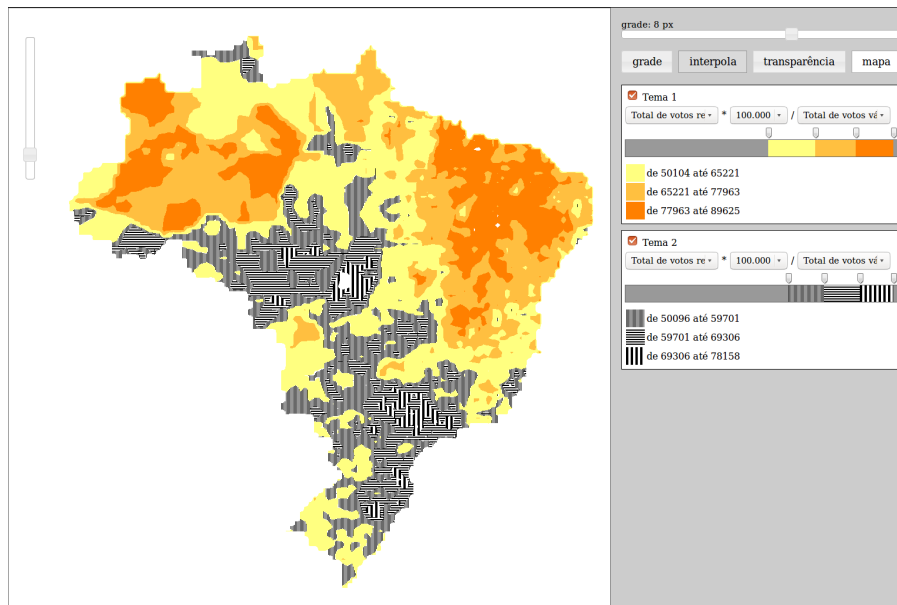


Figure 10. The map shows only the areas where the candidate has had more than half of the valid votes.

and Total valid votes. Figure 9 shows the votes received by each candidate for each 100 thousand valid votes. Theme 1 shows the votes received by Dilma Rousseff, the winner. Theme 2 shows the votes received by Aécio Neves. Both themes, when viewed separately clearly inform in what areas each candidate beat the opponent. Dilma Rousseff stands out in the country's North and Northeast regions. Aécio Neves, in turn, had a significant majority in the Southeast, South and Central-West regions. Figure 10 makes that even more explicit, because the applied classification makes the map show precisely where each candidate won and got the majority of valid votes.

5. Conclusion

The visualization of two or more variables on a map is a research challenge and rendering efficiency plays an important role in this process. This system proved to be able to perform this task, allowing fast transitions between user interactions in cases of multiple processing steps. The good performance lead us to consider the use of the techniques applied here to visualize challenging vector geometries like the census tracts, whose complexity and size makes it difficult to use them in online systems.

The grid view method was a good choice as well. It is an innovative approach and quite adequate to use with GPU processing, whose pixel shaders allow us to regard them as small real space fractions with instantaneous render and processing capacities. However, it is necessary to consider other aspects, such as the geographic projection distortion producing pixels that represent different areas of real space, thus requiring proper sampling methods for data decomposition.

The system proposed here applies simple techniques to solve the visualization problem using raster images to show and manipulate choropleth interactive maps. Much more can still be done in the field of georeferenced multivariate data visualization and we hope that the techniques implemented here, well as the obtained results, will incentive new research and experiments in order to fill in the existing gaps.

References

- Andrienko, G. (2001). Choropleth Maps: Classification Revisited. <http://geoanalytics.net/and/papers/ica01.pdf>. Access: July 3, 2016.
- Archela, R. S. and Théry, H. (2008). Orientação metodológica para construção e leitura de mapas temáticos. *Confins*, 3(3).
- Bertin, J. (1980). O Teste de Base da Representação Gráfica. *Revista Brasileira de Geografia*, pages 160–182. Access: July 3, 2016.
- Bueno, M. d. C. D. (2014). *Grade estatística: uma abordagem para ampliar o potencial analítico de dados censitários*. PhD thesis, Universidade Estadual de Campinas – Instituto de Filosofia e Ciências Humanas. Access: September 30, 2017.
- Bueno, M. d. C. D. and D'Antona, A. d. O. (2014). Avaliação de métodos de desagregação para geração de grades de população. *Revista Espinhaço*, 3(1):127–137. Access: July 3, 2016.
- Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368.

- Fernando, R. (2004). *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education.
- Gershon, N., Eick, S. G., and Card, S. (1998). Information Visualization. *Interactions*, 5(2):9–15.
- IBGE (2010). Evolucao da Divisao Territorial do Brasil 1872 - 2010. ftp://geoftp.ibge.gov.br/organizacao_territorial/divisao_territorial/evolucao_da_divisao_territorial_do_brasil_1872_2010/evolucao_da_divisao_territorial_mapas.pdf. Access: December 27, 2015.
- IBGE (2013). Malhas Digitais - Municípios 2013. ftp://geoftp.ibge.gov.br/malhas_digitais/municipio_2013/. Access: November 16, 2015.
- IBGE (2015). Grade Estatística - Guia de Utilização. ftp://geoftp.ibge.gov.br/malhas_digitais/censo_2010/grade_estatistica/ge_guia_utilizacao.pdf. Access: November 29, 2015.
- IBGE (2016). Cidades@ - Minas Gerais - Santa Cruz de Minas. <http://cidades.ibge.gov.br/xtras/perfil.php?lang=&codmun=315733>. Access: August 26, 2015.
- Inc, T. K. G. (2017). WebGL Overview. <https://www.khronos.org/webgl/>. Access: November 8, 2017.
- Interrante, V. (2000). Harnessing Natural Textures for Multivariate Visualization. *IEEE Computer Graphics and Applications*, 20-6:6–11. Access: July 3, 2016.
- Newman, M. (2012). Maps of the 2012 US presidential election results. [http://www-personal.umich.edu/~\\\$sim\\\$mejn/election/2012/](http://www-personal.umich.edu/~\$sim\$mejn/election/2012/). Access: July 21, 2015.
- Pickett, R. and Grinstein, G. (1988). Iconographic Displays For Visualizing Multidimensional Data. In *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, volume 1.
- Schwartz, J. (2016). Bing Maps Tile System. <https://msdn.microsoft.com/en-us/library/bb259689.aspx>. Access: August 12, 2015.
- Shneiderman, B. (1994). Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77.
- Stone, M. (2006). Choosing Colors for Data Visualization. *Perceptual Edge*, pages 1–10. Access: July 3, 2016.
- Tateosian, L. G., Healey, C. G., and Enns, J. T. (2007). Engaging Viewers Through Nonphotorealistic Visualizations. *Proceedings of the Fifth International Symposium on Non-Photorealistic Animation and Rendering*, pages 93–102.
- Tutorials, W. O. W. (2017). HTML - The language for building web pages. <https://www.w3schools.com/>. Access: November 8, 2017.
- (W3C), W. W. W. C. (2017). Web Design and Applications. <https://www.w3.org/standards/webdesign/>. Access: November 8, 2017.