Received: 18 September 2021 | Revised: 1 December 2022 | Accepted: 10 January 2023

DOI: 10.1111/tgis.13028

Transactions in GIS WILEY

RESEARCH ARTICLE

Visual analytics for diagnosing spatial interpolation models

André de Oliveira Bastos[1] | Emanuele Santos[1] | Darniton Viana[2] | Luciano Barbosa[2]

[1]Departamento de Computação, Universidade Federal do Ceará, Fortaleza, Brazil

[2]Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil

Correspondence
Emanuele Santos, Departamento de Computação, Universidade Federal do Ceará, Fortaleza - CE, Brazil.
Email: emanuele@dc.ufc.br

Funding information
Coordenação de Aperfeiçoamento de Pessoal de Nível Superior; Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico

Abstract

The development and widespread use of statistical learning models have brought the need for tools that help analysts diagnose, build, and refine those models. In this work, in particular, we focus on interpolation models, which spatially predict the value of a variable based on the values of its neighborhood. Investigating these results spatially or comparing them with other models at different levels of granularity is still a challenge for the analysts trying to understand and refine their models. To deal with that, we propose a visual analytics model-agnostic tool for facilitating the comparison and refinement of spatial models at different levels of granularity using interactive visualization techniques. The tool was built in collaboration with specialists who used it to diagnose and improve a spatial model for predicting residential real estate prices.

1 | INTRODUCTION

A fundamental task in real-world scenarios of statistical learning methods is model diagnostics.[1] Its goal is to identify possible weaknesses in the performance of a model in order to refine it. In the case of spatial models for prediction, analysts wish to investigate their models' output, usually predicted points associated with a target variable, as well as compare them with other models' output.

Visual analytics, the science of analytical reasoning facilitated by interactive visual interfaces (Thomas & Cook, 2005), can help with model diagnostics by combining a reasoning framework with visual representations, data transformations, and interaction techniques to support visualization and analysis.

In this article, we propose S-VIDIA (Spatial VIsualization DIAgnostics), a visual analytics model-agnostic tool for facilitating the comparison and refinement of interpolation models at different levels of granularity using interactive visualization techniques.

Although there are many different specialized visual analytics systems for analyzing geospatial data, for example, in crime analysis (Beecham et al., 2015; de Queiroz Neto et al., 2020; García et al., 2021; Malik et al., 2010) and movement analysis (Adrienko & Adrienko, 2011; Shi et al., 2020; Wang et al., 2013), most analysts still rely on general-purpose geographic information systems (GIS) to conduct spatial analysis of their models (Chainey et al., 2008; Ciski et al., 2019; Feizizadeh et al., 2017). These systems usually have a wide variety of functionalities and offer an integrated environment to load the data and different ways to display the data.

Even though a general-purpose GIS tool has different spatial interpolation techniques available to deal with the data's granularity, such as Kernel Density Estimation (KDE) (Parzen, 1962) or Kriging (Cressie, 1990), the generated visualizations are only suitable for visual inspection of a single model. As occlusion prevents the users from overlapping the visualizations efficiently, the visual analysis of these comparisons is often done side-by-side or in different moments in time (Ferreira et al., 2013; Kong et al., 2018; Machwitz et al., 2018; Plouffe et al., 2015). In these strategies, the viewers must rely on their memory to make the comparison, causing an extra cognitive load. Another alternative is to explicitly show the difference between the models by providing a visual encoding of them (Gleicher et al., 2011).

In addition, there are existing approaches that compare different spatial interpolation techniques. The work by Thakali et al. (2015), for instance, compares the use of Kriging and KDE techniques. Their solution, also used in other works (Chainey et al., 2008; Ciski et al., 2019), consists of creating hotspot maps from density maps using GIS tools and later comparing them side-by-side. As these techniques rely on appropriate parameter settings to be successfully used, some works focus on finding ideal parameters for a given set of points (Bachoc, 2013, 2018; Chiu, 1991; Krisp, 2010). However, sometimes the parameters suggested by a tool like ArcGIS can fail in being satisfactory (Krisp & Špatenková, 2010). For that reason, it is important to allow users to visualize the result for different parameters, making them choose the most suitable parameters for their needs (Flanagan et al., 2013; Krisp et al., 2009; Krisp & Špatenková, 2010).

Some studies focus on comparing different spatial interpolation models to find the most appropriate model for a given geospatial dataset (Li & Heap, 2014). For example, models influenced by multiple factors, such as yield, soil fertility (Brus et al., 1996; Wang et al., 2020), or pollution (Liao et al., 2018), and hazard mapping, in which an accurate model that is essential for safety (Bui et al., 2015; Chainey et al., 2008; Chung & Fabbri, 2003).

Analysts can also benefit from comparing numerical metrics (Caruso & Quarta, 1998; Drawve, 2014), but usually, the metrics are globally defined and do not provide enough information for understanding all the characteristics of the model. Moreover, there are cases in which two different spatial models may present the same metric, for example, average error, but the error distribution is very distinct in both models. Here, a visual approach may also help inspect these models locally.

In our approach, besides displaying the models separately, S-VIDIA allows analysts to directly visualize the difference between two models using shaded isoline maps. If the spatial models' data are provided as a discrete set of points, our approach transforms the set into a unified density function across a regular grid and generates polygon layers for visual comparison. We test this process with KDE and Kriging. Our approach also builds on the concept of relative accuracy (Yang & Hodler, 2000), in which the relative variation of errors across the grid can also be visualized and stored as a separate geospatial data layer. Analysts can benefit from both having a numerical value to quantify the difference and visualizing how that difference relates to the dataset and geography. S-VIDIA was built in collaboration with machine learning specialists, who helped us define S-VIDIA's analytical tasks (ATs) and design goals. We demonstrate S-VIDIA's utility in a real-world case study, in which the specialists used S-VIDIA to diagnose and improve a spatial model for predicting residential real estate prices.

S-VIDIA is an attempt to answer the call in the Visual Analytics research agenda (Thomas & Cook, 2005) to build more interactive tools that enable users to obtain insights that directly support decision-making.

The goal of spatial interpolation is to predict the value of an observation based on the values of its geographic neighbors. In Section 1.1, we define the problem we are solving more precisely, describing the ATs that analysts perform when building and diagnosing spatial interpolation models and contextualizing them in an established geovisualization taxonomy (Roth, 2013). In Section 1.2, we summarize the design goals that a visual analytics system must incorporate to support performing the ATs.

1.1 | Problem definition and analytical tasks

During the collaboration with the specialists, we identified five different ATs they needed to perform when building and diagnosing a new model for spatial interpolation. These five tasks were defined based on their experience (Viana & Barbosa, 2021) and other previous work research on spatial model comparison and visualization we discussed above, such as Ciski et al. (2019), Thakali et al. (2015), and Yang and Hodler (2000). The tasks can be performed to verify the quality of spatial interpolation models independently of the model used and the application area. Below, we first describe the *Domain Tasks* (DTs) using the specialists' domain language, and we then unify their terminology in geovisual analytics by expressing them as *Analytical Tasks* (ATs) based on the taxonomy proposed by Roth (2013).

1.1.1 | DT1: Visualizing data sparseness

Since spatial interpolation models rely on points in the geographic vicinity of unknown observations to predict their values, performing predictions in sparse regions might lead to high errors. In this task, we aim to inspect the overall geographic distribution of the observations to identify possible "blind" spots[2] for the model in order to try to relate them with the performance of the model in these regions, and possibly provide some solution that fills these gaps in the data.

1.1.2 | DT2: Inspecting the spatial autocorrelation of attributes

The key idea of spatial autocorrelation is that observations of the same variable at different locations might be correlated. Examples of autocorrelated variables are the temperature and rainfall in a certain location. In spatial interpolation, in particular, a fundamental premise is that model's target attribute shows spatial autocorrelation. In addition, for the prediction quality, it is important that the attributes used by the model also present spatial autocorrelation, since it might be associated with the model's target. This task, therefore, identifies how useful attributes are for spatial interpolation based on their spatial autocorrelation.

1.1.3 | DT3: Evaluating the random spatial error distribution

Spatial interpolation models capture the spatial dependence of the target variable when the errors of the models are randomly spread on the geographic space (Hamylton et al., 2015). The main assumption of spatial interpolation is that the spatial distribution of high values or low values is more spatially clustered than it would be expected if the underlying spatial processes were random. When the modeling captures the factors that generate this spatial dependence, the residual (error) between the real value (ground truth) and the predicted value from the model has its magnitude due to the non-modeling of factors unrelated to the geographic positioning. In other words, the pattern is random and does not have spatial autocorrelation. On the other hand, if a cluster pattern is present in the spatial error distribution, closer residuals tend to have more similar values, meaning it has spatial autocorrelation. The goal of this task is, hence, to inspect the model's spatial error distribution to verify the quality of the model's prediction spatially.

1.1.4 | DT4: Verifying the regions with high errors

The objective of this task is to inspect the geographic regions with high prediction errors of the interpolation model. This is helpful, for instance, to understand the possible causes of the errors in order to improve the predictions' quality.

1.1.5 | DT5: Comparing visually the model's prediction and the ground truth

In this scenario, we aim to compare the model's prediction with the ground truth in specific regions on the map. This would allow a better understanding of the prediction quality of a model in these regions, which also facilitates a comparison with other models.

1.1.6 | DT6: Inspecting the raw values

Another useful task to better understand the model's performance is to inspect its predicted values in a particular geographic region, comparing them with the real values or another model. In contrast with Task 5, this task is a numerical comparison, being also relevant for validation and debugging purposes.

According to Roth's (2013) taxonomy, which focuses on map-based visualization, we rewrite the DTs as more general ATs in the form of a goal + objective + operand combination. Table 1 details the relationship between DTs and ATs. Although the ultimate goal of the spatial interpolation model is prediction, individually, all the tasks are under the cognitively simpler goal *Procure* in the taxonomy, and because of that, the *Goal* column is omitted in Table 1. By inspecting this table, we notice that the six DTs generated eight ATs and that all five objective primitives in the taxonomy are present (*Identify*, *Compare*, *Rank*, *Associate*, *and Delineate*). As for the operands, only the search target *space-in-time* is not present because the time dimension is irrelevant for the domain specialists. Tasks DT5 and DT6 are more complex and produce two ATs each. When performing DT5, for example, the specialists can compare the shape of the regions (AT5) or look into the difference between attribute values (AT6). In the case of DT6, the specialists can inspect specific values in a location (AT7) or compare them (AT8). Tasks AT7 and AT8 are the only ATs whose search level is elementary, involving only one map feature. All the other tasks are general, involving several-to-all map features.

The ATs and the taxonomy will be referred to again in Section 2 when we describe the interaction operators.

1.2 | Design goals

During the regular meetings with the specialists, when we were iterating versions of our prototype, it became clear that the visual analytics system should contain a set of features in order to support them performing the ATs. These

TABLE 1 Mapping the domain tasks to analytical tasks according to Roth's (2013) taxonomy.

			Operand	
Domain task	Analytical task	Objective	Search target	Search level
DT1: Visualizing data sparseness	AT1: Identify spatial regions	Identify	Space-Alone	General
DT2: Inspecting spatial autocorrelation of attributes	AT2: Associate attributes for correlation	Associate	Attributes-in-space	General
DT3: Evaluating random spatial error distribution	AT3: Delineate spatial clusters	Delineate	Space-Alone	General
DT4: Verifying regions with high errors	AT4: Rank areas by attribute	Rank	Attributes-in-space	General
DT5: Comparing visually the model's prediction and the ground truth	AT5: Compare spatial shapes	Compare	Space-Alone	General
	AT6: Compare spatial regions by attribute	Compare	Attributes-in-space	General
DT6: Inspecting the raw values	AT7: Identify attribute values	Identify	Attributes-in-space	Elementary
	AT8: Compare attribute values	Compare	Attributes-in-space	Elementary

features are similar to the concept of non-functional requirement in software development (Glinz, 2007), but the term design goal is more frequently used in the visualization literature (Sedlmair et al., 2012). The design goals listed below guided the development of our approach and the visualization choices we made.

1.2.1 | DG1 enable interactivity and exploration

The users should be able to modify: data selection (filtering), visual mappings, and view transformation (navigation). In other words, the users have to be able to visualize their model as a whole, in an overview, and obtain details in certain regions on demand (Shneiderman, 1996). In addition, the interactive visualizations should provide rapid response times.

1.2.2 | DG2 facilitate data loading

The data input format of the system should be standard and flexible. For example, the users should be able to enter a list of data points in a CSV file as well as a data grid of values with custom cell size.

1.2.3 | DG3 facilitate multi-resolution analysis

The users should be able to analyze their data at different levels of aggregation, from a single data point, or a grid cell to bigger geometric shapes. They may provide their own aggregation model if they want.

1.2.4 | DG4 facilitate comparison

Users should be able to at least compare two datasets in the same window in different ways. In the case of more than two models, facilitate side-by-side comparisons by sharing view parameters between them (map center and zoom level).

2 | METHODS

In this section, we present S-VIDIA, a visual analytics system designed to facilitate the diagnoses and comparisons of spatial interpolation models. The components of the system are described in Sections 2.1 and 2.2, we present the workflows for using the system.

2.1 | Our approach

Our approach follows a web-based client–server architecture (Figure 1), in which the analysts access the system through a web page, and the computations are performed on the server side. This means that users input the data and settings required to generate the visualizations and later explore the results by interacting with the web interface. S-VIDIA has five main components: the web interface, the Grid Manager, the Visual Spatial Interpolator, the Tier Divider, and the Polygon Generator. As Figure 1 shows, each component is responsible for a step in the process of generating the visualizations and two types of data flow are supported depending on the input provided. The

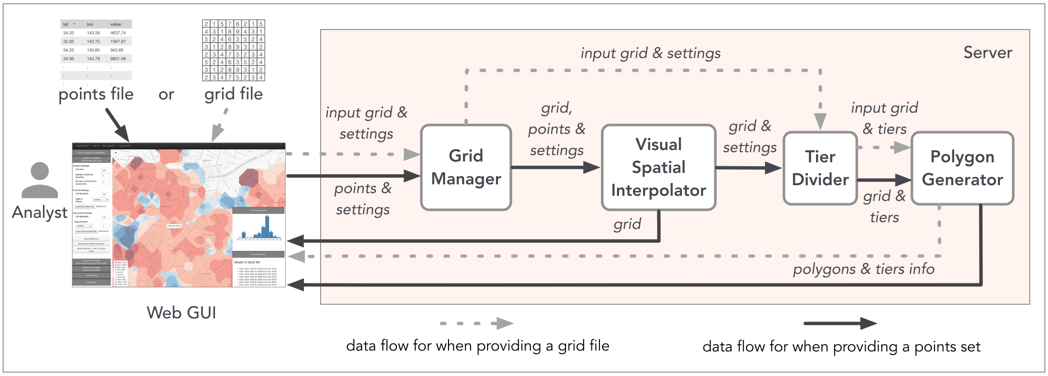


FIGURE 1 S-VIDIA's architecture overview. Two types of data flow are supported depending on the input provided to the system: a point file (darker solid arrows) or a grid file (lighter dotted arrows).

Grid Manager controls the generation of the grid in which the spatial interpolation occurs, but the analysts may also provide their own interpolated grid. In this case, the Grid Manager will skip the interpolation and pass the grid directly to the Tier Divider, following the lighter dotted arrow. When the analysts upload sets of points, the Visual Spatial Interpolator uses the grid generated by the Grid Manager and the user settings to interpolate the input points. The analysts can visualize a single model, display two models overlapped or compare them. The details of the different flows will be discussed in Section 2.2. Then, given a grid, the Tier Divider computes the range of values for each tier (or layer) into which the polygons are distributed. Finally, the Polygon Generator draws the polygons for each tier. The resulting polygons are returned to the interface along with the information about the tiers and the grid. Next, we describe each component in more detail.

2.1.1 | Grid manager

When using S-VIDIA, the analysts may input to the system lists of points or precomputed grids. This simple input format supports design goal *DG2—Facilitate data loading*. As the users already manipulate lists of points when developing their models, it is easy for them to input them to the system.

The Grid Manager is responsible for creating the data structure representing the grid of cells and may process the inputs differently depending on them being lists of points or precomputed grids. The most straightforward case is when the analyst provides a single grid file as input. This means that the data are already interpolated, so the Grid Manager can build the grid representation from the file and send it directly to the Tier Divider. If the user provides two grid files, the system requires that they have the exact same dimensions and cell size. The more interesting cases are when the analyst provides lists of points and the cell size in the settings to create the grid. In this case, the Grid Manager computes the bounding box that contains all the points across the lists and generates a grid representation with the given cell size that fits the bounding box. This grid representation is used for all lists of points in the input. Next, the Grid Manager sends the grid representation, the list(s) of points, and the settings to the Visual Spatial Interpolator component.

2.1.2 | Visual spatial interpolator

The visual spatial interpolator component is responsible for interpolating the points on the grid for visualization. Besides being used for a smooth visualization, this spatial interpolation allows comparing different sets of points on the same map without clutter. In other words, the interpolation performed in the visual spatial interpolator component is for visualization purposes only, and must be distinguished from the interpolation method that the analyst aims to diagnose using S-VIDIA, to which the system is oblivious.

The visual spatial interpolator performs two types of operations, depending on the analyst's input: a spatial interpolation or a comparison operation. In the spatial interpolation, the component receives the grid(s), the list(s) of points, the interpolation method and its parameter settings. In this version of the system, we implemented two common interpolation methods: KDE and Kriging. This component, however, was designed in such a way that other spatial interpolation methods can be easily added, as long as the interpolation method returns a grid of values representing a continuous surface. The parameters required by the interpolator can be exposed in the Web interface as html elements for the analyst's use. We explain this process in further details in Section *Interpolation Methods*.

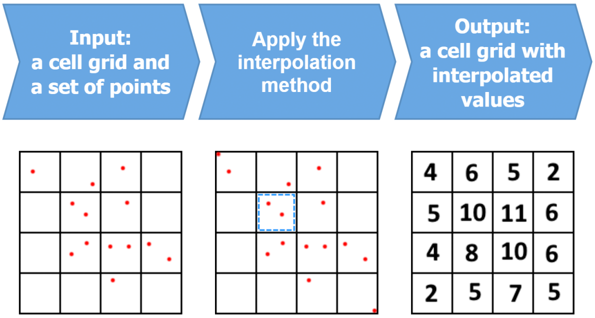
The other type of operation this component performs is comparing two grids. When the analyst requests this task, the Visual Spatial Interpolator computes a cell-by-cell difference of the grids and stores the resulting values in a third grid. This procedure is detailed in Section *Computing differences between grids*.

The Visual Spatial Interpolator contributes to design goal *DG3—Facilitate multi-resolution analysis*. By changing the interpolation methods and their parameter settings, users can achieve different levels of aggregation.

Interpolation methods

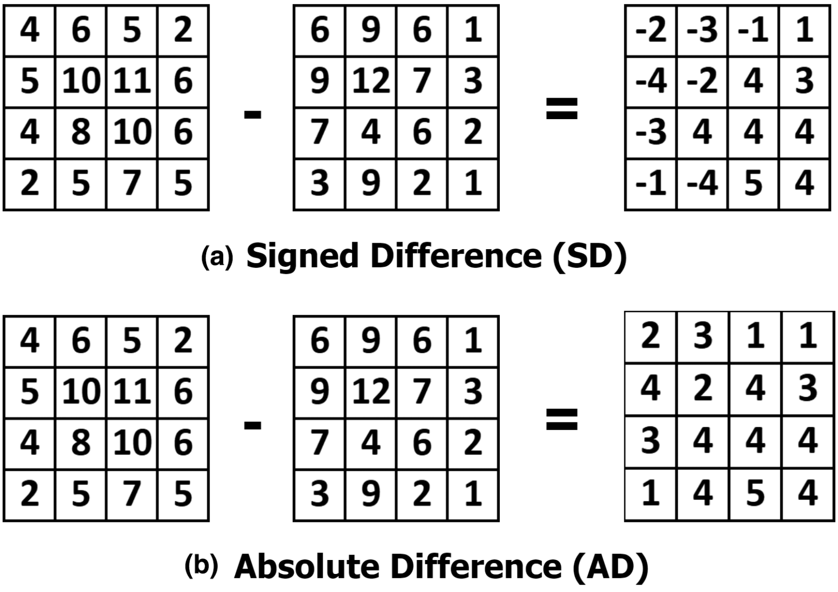
Figure 2 illustrates the general workflow for the spatial interpolation methods supported by Visual Spatial Interpolator. Any method that receives as input a grid of cells and a set of points and produces a grid of cells containing values representing a continuous surface can be integrated as a plugin in the system. The constraint of being a continuous surface is required to generate valid polygons at the final step of the visualization.

To validate our solution, we implemented two spatial interpolation methods: KDE and Kriging. When choosing KDE, the analysts must provide, besides the cell size already used to build the grid, a bandwidth value, also called a smoothing parameter, and a kernel function, which is a window non-negative function that computes the weight of a point based on its distance to the cell center. Several kernel functions commonly used with KDE are available for the analysts (Silverman, 1986): uniform, triangular, quartic, Gaussian, and Epanechnikov (1969). When choosing Kriging as the interpolation method, the analysts can use Ordinary Kriging (OK) and Universal Kriging (UK). In terms of usage, the input of OK is the geo-location position of the points (e.g., latitude and longitude), whereas UK performs the interpolation based on auxiliary variables associated with the points in addition to their geo-location. Although both are geostatistical-based methods, the main issue concerning OK is that it assumes a stationary, that is, constant mean over each region, and the predictions rely solely on point observations of the target variable. In contrast, UK assumes that the mean has a functional dependence on the spatial location and makes predictions by modeling the relationship between the target and auxiliary variables (Mesic Kiš, 2016; Thakali et al., 2015). The settings for each method can be set through the web interface.



4	6	5	2
5	10	11	6
4	8	10	6
2	5	7	5

FIGURE 2 General spatial interpolation workflow.



4	6	5	2
5	10	11	6
4	8	10	6
2	5	7	5

-

6	9	6	1
9	12	7	3
7	4	6	2
3	9	2	1

=

-2	-3	-1	1
-4	-2	4	3
-3	4	4	4
-1	-4	5	4

(a) **Signed Difference (SD)**

4	6	5	2
5	10	11	6
4	8	10	6
2	5	7	5

-

6	9	6	1
9	12	7	3
7	4	6	2
3	9	2	1

=

2	3	1	1
4	2	4	3
3	4	4	4
1	4	5	4

(b) **Absolute Difference (AD)**

FIGURE 3 A visual explanation of computing the difference between two grids.

Computing differences between grids

When the analyst requests a model comparison, another processing step is applied to the grids before sending them to the Tier Divider. The two grids can be compared in three different ways: no difference, signed difference, and absolute difference.

In the first type of comparison, the grids are displayed as they were originally computed, so no operation is performed on the grids. Each grid will be displayed in a different layer and can be shown separately or overlapped on the map.

In the second type, the system computes the difference of the two grids as the Signed Difference (SD). Thus, for each grid cell *c*:

$$SD(c) = Z'_1(c) - Z'_2(c) \tag{1}$$

where $Z'_1(c)$ and $Z'_2(c)$ are the values associated with that cell in the first and the second grids, respectively. Notice that as this is the signed difference, the cells may have negative values as displayed in Figure 3a.

Finally, in the third type, the system computes the absolute difference between the two grids, by calculating the absolute value of the signed difference defined above. So for each cell *c*:

$$AD(c) = |\, SD(c)\, | = |\, Z'_1(c) - Z'_2(c)\, | \tag{2}$$

where $Z'_1(c)$ and $Z'_2(c)$ are the values associated with that cell in the first and the second grids, respectively (see Figure 3b).

After performing all the operations, the output of the Visual Spatial Interpolator is sent to the next component in the pipeline: the Tier Divider. In the case of computing the signed or absolute difference, the output contains only the resulting grid.

2.1.3 | Tier divider

The Tier Divider splits the grid values into a predefined number *k* of tiers. When comparing two grids, this division can be performed in two ways: an equal division of cells or an absolute max-value wise division. Figure 4 illustrates

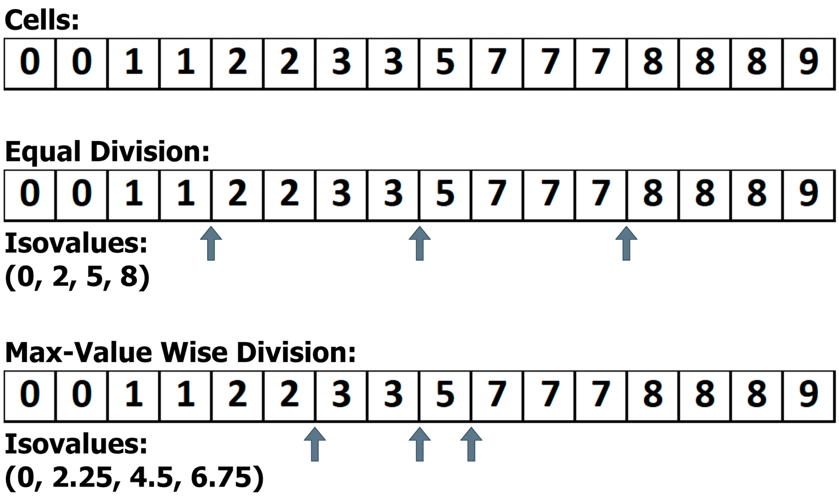


FIGURE 4 An illustration of the two ways the Tier Divider splits the cells into tiers. In this example, a grid of 16 cells is split into $k = 4$ tiers. In the Equal Division, each tier has approximately an equal number of cells. In the Max-Value Wise Division, the tiers are computed in proportion to the maximum absolute value $M = 9$ and each tier has a range of size $\frac{M}{k} = \frac{9}{4} = 2.25$.

these two types of division of a grid of 16 cells into $k = 4$ tiers. First, the cells are sorted in a linear array. In the equal division, to avoid tiers with few cells due to outliers, we choose the interval values of each tier so that all tiers have about the same number of cells. In the max-value wise division, on the other hand, we wish to focus our attention on the extreme values. We then compute the tiers proportionally to the maximum absolute value. For example, in Figure 4, the maximum value is 9 and there are $k = 4$ tiers so that each tier has a range of size $\frac{M}{k} = \frac{9}{4} = 2.25$. The first tier corresponds to the [0, 2.25) interval, the second tier corresponds to [2.25, 4.5) interval, and so on. Notice that the top tier contains the highest values in the grid and can be used to find a range of the extreme values.

The division of the data into tiers allows another level of data aggregation, also contributing to design goal *DG3—Facilitate multi-resolution analysis*.

In Section 2.2, we show that, by combining how the tiers are divided with the ways the grids are compared, our solution can present several different visualizations, so the analyst is able to compare the models from different perspectives.

2.1.4 | Polygon generator

The Polygon Generator component generates polygons using the technique called Marching Squares Kernel Density Estimation—MSKDE (de Queiroz Neto et al., 2016). Figure 5 shows an overview of how the Polygon Generator uses MSKDE. First, the Polygon Generator receives the interpolated grids and the tier limits from the Tier Divider (Figure 5a). For each input grid, it builds a secondary grid in such a way that each cell center becomes a vertex in the secondary grid (Figure 5b). Then, for each tier starting at the second bottom tier, "march" through the vertices in the secondary grid, using the Marching Squares algorithm (Lorensen & Cline, 1987), to generate the contour lines for each tier's lower limit (Figure 5c). The system computes the final polygons by applying polygon differences between two consecutive tiers starting from the bottom tier (Figure 5d). The resulting polygons of each tier are returned with the tier's interval limits to the web interface. The interval limits are used to populate the map legend.

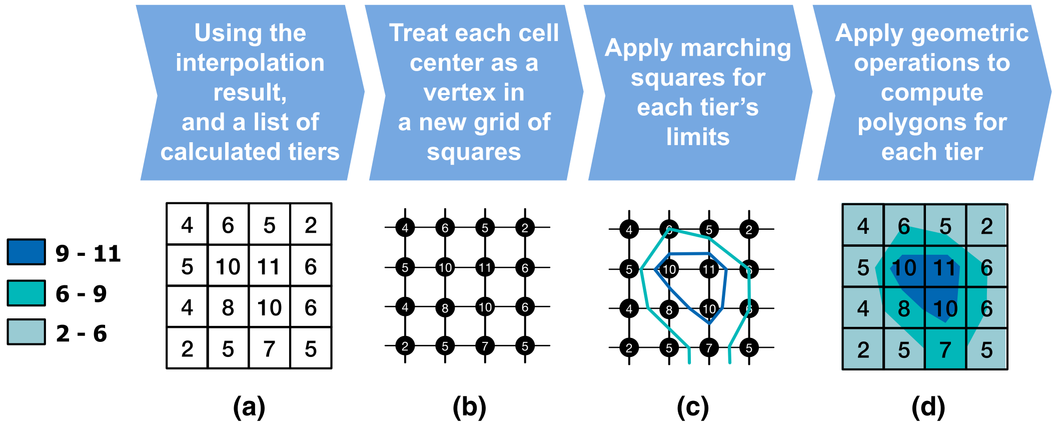

Using the interpolation result, and a list of calculated tiers
Treat each cell center as a vertex in a new grid of squares
Apply marching squares for each tier's limits
Apply geometric operations to compute polygons for each tier
9 - 11
6 - 9
2 - 6
4 6 5 2
5 10 11 6
4 8 10 6
2 5 7 5
(a)
(b)
(c)
(d)

FIGURE 5 The workflow to generate the polygons.

2.1.5 | Web interface

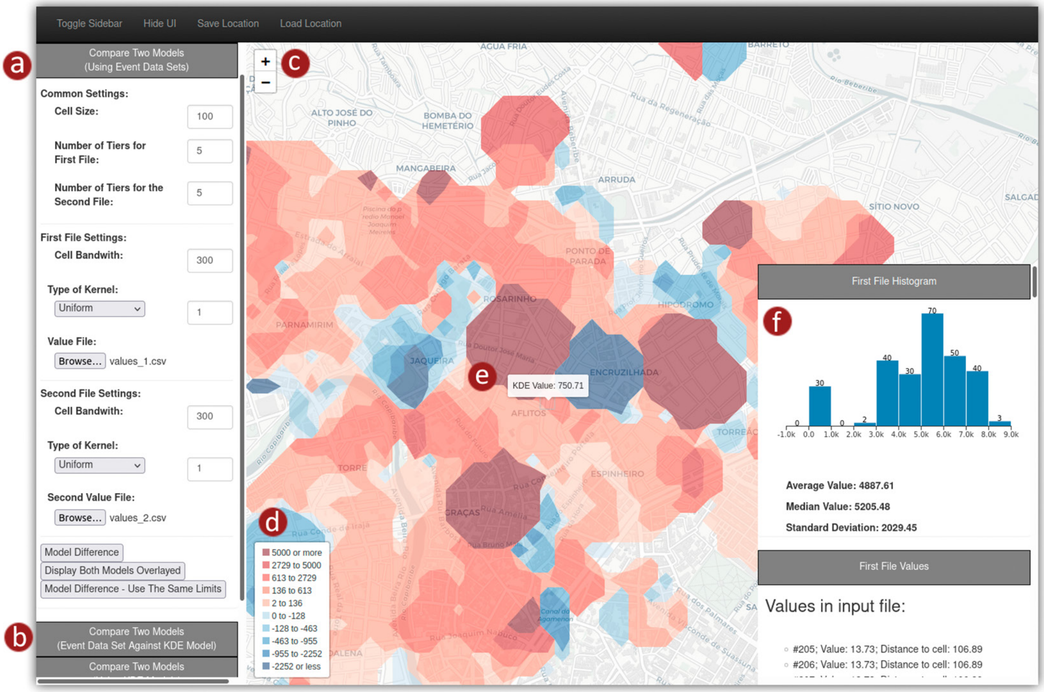
S-VIDIA's web interface is displayed in Figure 6. It has two main components: a map view (Figure 6c) and a sidebar (Figure 6a). The sidebar contains all the configuration controls for working with the datasets (Figure 6a) and the visualizations (Figure 6b). These components were developed to support design goal DG1—Enable interactivity and exploration and the *Operator Primitives* in Roth's (2013) taxonomy. The *Enabling* operators available on the sidebar are *Import*, *Export*, and *Save*. The *Work* operators available on the sidebar are *Reexpress*, *Resymbolize*, *Overlay*, and *Filter*, and on the map view are *Pan*, *Zoom*, *Retrieve*, and *Calculate*. For example, by using the sidebar, the analysts can turn on and off each tier's visibility (*Overlay*) and select the comparison mode displayed on the map (*Reexpress or Resymbolize*). We include references to the relevant operators in the remainder of this section.

The map view is where the visualizations are displayed once the server completes computing them and contains all elements relevant to the visualizations, including showing details of each cell on demand, using a tooltip with the cell value (Figure 6e) and a Details panel (Figure 6f). Figure 7 shows a more detailed view of these visual elements. The types of maps displayed on the map view are either shaded isoline maps or choropleth maps using a regular grid (Golebiowska et al., 2021). The map colors are based on ColorBrewer color schemes (Brewer, 2009).

When selecting a cell of the grid, a square will appear overlaying the cell alongside a tooltip (*Retrieve* operator). The square shows the exact boundaries of a cell, as used by the server in the analysis, and the tooltip indicates the value for that cell (or values, in case the visualization is showing both models overlapping each other). This allows the analyst to understand why a cell is in a certain tier, as well as how the interpolation was computed in that cell. This feature fulfills the analytical task AT7 (see Table 1).

If the analysts are visualizing a dataset of points, a toggle bar on the bottom right of the map appears when selecting a cell. When toggled, it displays a Details panel with a histogram and descriptive statistics of the point values used for computing the estimate in the cell in the interpolation (*Calculate* operator). The histogram is a quick overview of all values involved in the calculation. It allows the analysts to, at a glance, notice inconsistencies such as outliers or if there are too many or too few values influencing a given cell, as in AT8 (see Table 1). Below the histogram other descriptive statistics (average, median, and standard deviation) are also shown. The list below the statistics provides even more details by showing all data points used in the evaluation of that cell, with their position in the file. This allows the analysts to verify their files in search of that event. The list also displays the datapoints' distance to the center of the cell, completing the contributing factors for computing the estimate of that cell.

In the next section, we describe how the analysts can use the system to create visualizations and compare models in more detail.



Common Settings:
Cell Size: 100
Number of Tiers for First File: 5
Number of Tiers for the Second File: 5

First File Settings:
Cell Bandwith: 300
Type of Kernel: Uniform 1
Value File: Browse... values_1.csv

Second File Settings:
Cell Bandwith: 300
Type of Kernel: Uniform 1
Second Value File: Browse... values_2.csv

- 🟥 5000 or more
- 🟥 2729 to 5000
- 🟥 613 to 2729
- 🟥 136 to 613
- 🟥 2 to 136
- 🟦 0 to -128
- 🟦 -128 to -463
- 🟦 -463 to -955
- 🟦 -955 to -2252
- 🟦 -2252 or less

Average Value: 4887.61
Median Value: 5205.48
Standard Deviation: 2029.45

Values in input file:

- #205; Value: 13.73; Distance to cell: 106.89
- #206; Value: 13.73; Distance to cell: 106.89

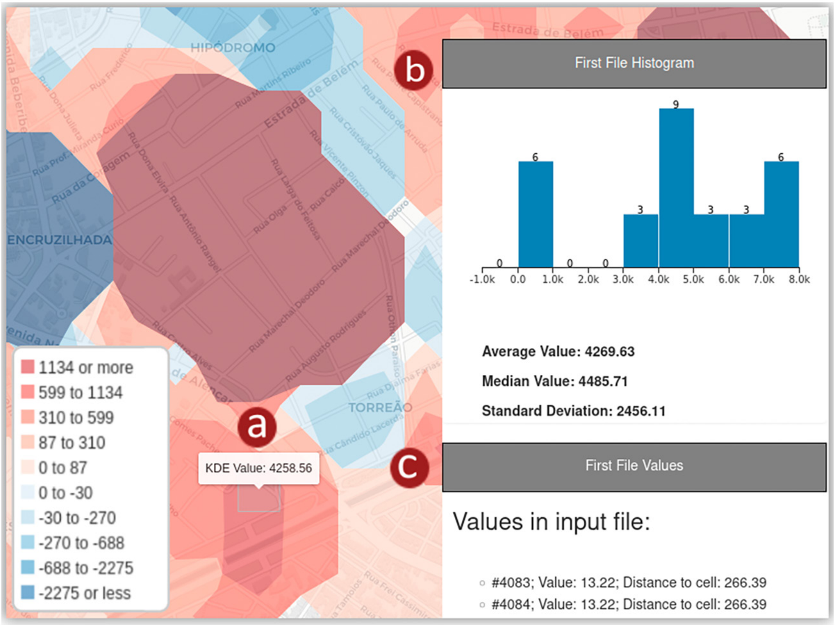
FIGURE 6 S-VIDIA's web interface. (a) The analysis settings toolbar providing *Import*, *Export*, *Save, and Reexpress* operators. (b) Color and visualization settings based on ColorBrewer color schemes (Brewer, 2009) and supporting *Reexpress and Resymbolize* operators. (c) Map view supporting *Pan, Zoom, and Retrieve* interaction operators. (d) Legend box. (e) Tooltip with information for the selected cell. (f) Details panel with information for that cell. This visualization shows a shaded isoline map using a diverging color scheme, in which red polygons indicate regions where the first model has higher values in that region, whereas blue polygons are regions where the second model has higher values.

2.2 | Workflows for using the system

As we mentioned when describing the components of the system, the analysts can visualize their models in different ways, depending on the ATs (see Table 1) they want to accomplish. By using the web interface, the analysts can visualize a single model (needed in AT1 and AT2), show both models overlapped (needed in AT5), or compare two models (needed in AT3, AT4, and AT6). Each of these objectives corresponds to a different workflow which we describe next.

2.2.1 | Visualizing a single model

Figure 8 describes the workflow used for visualizing a single model. In this example, the user selected a file containing the dataset of points and values and adjusted the analysis settings with a cell size of 100 m, chose KDE with a Gaussian kernel function as the interpolation method, a bandwidth of 300 m and five tiers (*Import and Reexpress* operators). The server receives these input data, generates a grid with 100 m sized cells, and then interpolates the grid using KDE with bandwidth 300 m. As it is a single model, the five tiers are created containing approximately the same number of values. After that, the polygons are generated and sent to the web interface to be color mapped and rendered. The analysts can then interact with the map (*Pan*, *Zoom*, *Retrieve* operators), change the color scheme (*Resymbolize* operator), and show and hide tiers (*Overlay* operator) to inspect the model.



First File Histogram

Average Value: 4269.63

Median Value: 4485.71

Standard Deviation: 2456.11

First File Values

Values in input file:

- #4083; Value: 13.22; Distance to cell: 266.39
- #4084; Value: 13.22; Distance to cell: 266.39

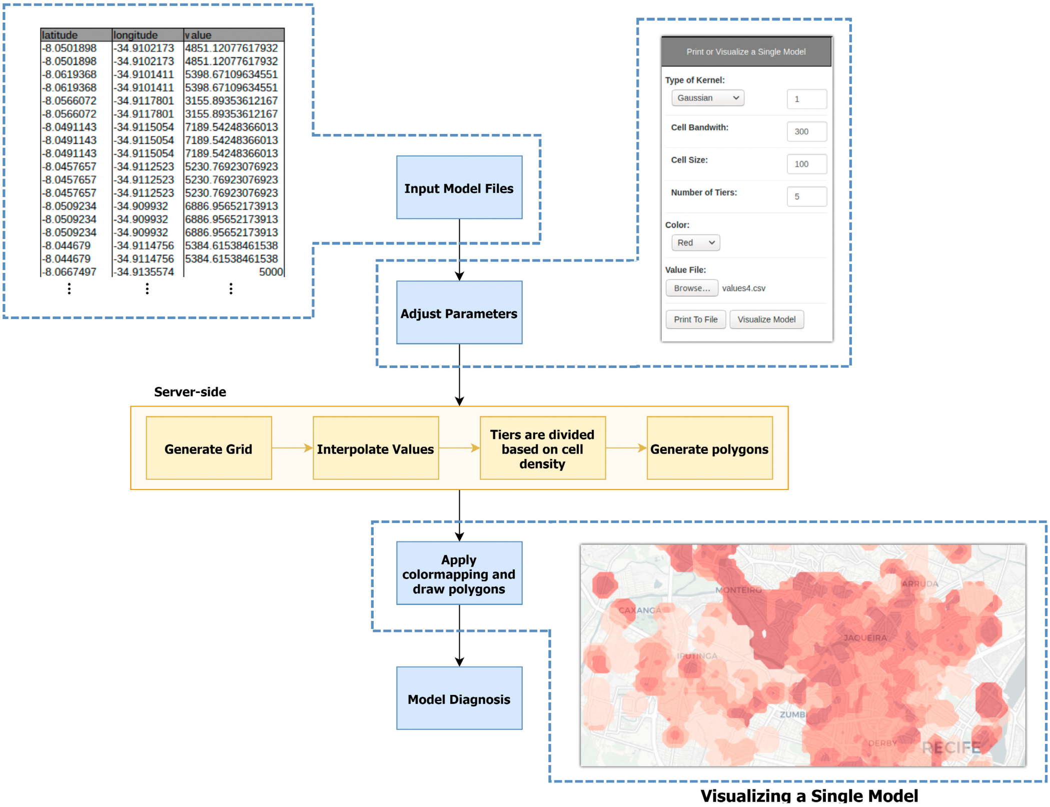
FIGURE 7 Displaying details on demand (*Retrieve* operator). (a) A selected cell is indicated by a square, as well as a tooltip showing the value for that cell. (b) A histogram and descriptive statistics of all values in the first dataset which were inside the bandwidth of that cell (*Calculate* operator). (c) A list of all these values and their respective positions in the file, and distance to the center of the cell (*Calculate* operator). This panel supports AT7 and AT8.

2.2.2 | Visualizing both models overlapped

Visualizing both models overlapped works similarly to current approaches when comparing different models: they are displayed on top of the other. Figure 9 describes the workflow for visualizing two models. This workflow is more complex than the single model workflow because it also captures the ways the analysts can compute the differences between the two models. To visualize the two models overlapped in our approach, the analyst first uploads the data files and adjusts parameter settings for each model (*Import and Reexpress* operators). The server generates the grids with the same cell size and interpolates the values for each model according to the input settings. It then follows the flow shown by the white boxes in Figure 9: for each grid, the server divides the cells, so each tier has approximately the same number of values and then generates the polygons for each tier. The resulting polygons are sent to the web interface, in which they are overlaid since they share equal grid dimensions. Depending on the visualization parameters used, this could lead to information overload for the users. However, as they can show and hide the tiers on demand (*Overlay* operator) and control the filling of the polygons (*Resymbolize* operator), the users can reduce the clutter of the visualization. An extra feature requested by the analysts is displaying the total difference between the models for each tier, that is, regions present in one model but not in the other (see Figure 10). In addition, the system also computes the total area of this difference in squared meters and the total sum of all tiers' differences combined (*Calculate* operator). With this feature, the users can have another metric to compare the models. This computation is performed directly on the browser, according to the user's input.

2.2.3 | Visualizing the difference between two models

As we mentioned in Section 2.1.2, our approach provides multiple ways to visualize the difference between two models. Figure 9 also depicts the flow for computing and visualizing that difference. The user uploads the data files



latitude	longitude	value
-8.0501898	-34.9102173	4851.12077617932
-8.0501898	-34.9102173	4851.12077617932
-8.0619368	-34.9101411	5398.67109634551
-8.0619368	-34.9101411	5398.67109634551
-8.0566072	-34.9117801	3155.89353612167
-8.0566072	-34.9117801	3155.89353612167
-8.0491143	-34.9115054	7189.54248366013
-8.0491143	-34.9115054	7189.54248366013
-8.0491143	-34.9115054	7189.54248366013
-8.0457657	-34.9112523	5230.76923076923
-8.0457657	-34.9112523	5230.76923076923
-8.0457657	-34.9112523	5230.76923076923
-8.0509234	-34.909932	6886.95652173913
-8.0509234	-34.909932	6886.95652173913
-8.0509234	-34.909932	6886.95652173913
-8.044679	-34.9114756	5384.61538461538
-8.044679	-34.9114756	5384.61538461538
-8.0667497	-34.9135574	5000
⋮	⋮	⋮

Print or Visualize a Single Model

Type of Kernel: Gaussian 1

Cell Bandwith: 300

Cell Size: 100

Number of Tiers: 5

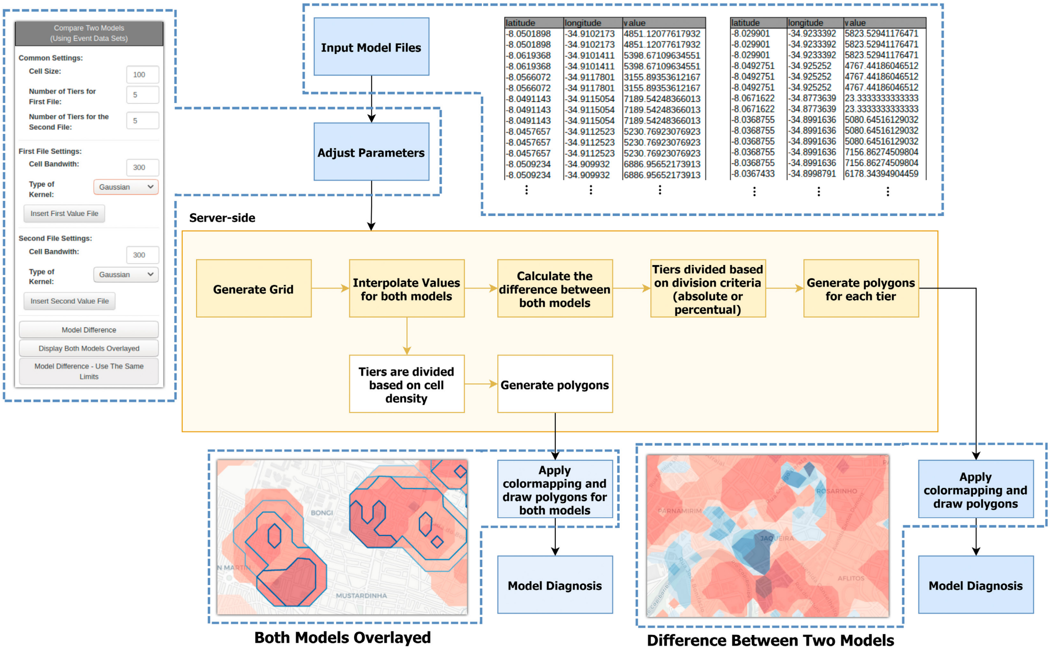
Color: Red

Value File: Browse... values4.csv

FIGURE 8 Workflow for visualizing a single model.

and adjusts the parameter settings for each model in the same way as displaying both models overlapped. However, this time the system follows another flow in the server (shown in the yellow boxes in Figure 9). After interpolating the values, the system calculates the differences between the two grids as described in Section 2.1.2. The tiers are divided according to the division criteria chosen by the analyst. The system then generates the polygons for all tiers and sends them to the web interface to be displayed as a change map on the map view. This approach follows an intrinsic coincident display (Kinkeldey et al., 2014; MacEachren et al., 1998), in which the differences are coded using the same symbology and shown in an integrated view (coincident). The analysts can then change the diverging color scheme for both positive and negative values and change the visualization modes to whichever suits best his goal at the time (*Resymbolize and Reexpress* operators).

For example, the analysts may select to see *Only Positive Values* (Figures 11a) or *Only Negative Values* (Figure 11b). These are examples of the *Filter* operator. Another option is to select *Both Models' Polygons Combined* to overlay both of these views in one, allowing the user to see all regions where any difference between both models exists (Figure 11c). Since overlaying both views can result in intersections in regions that transition from positive to negative values due to our interpolation algorithm, a fourth option is available called *All Tiers in a Single Polygon Calculation* (Figure 11d). In this approach, the system generates the polygons for both negative and positive values at the same iteration, resulting in interpolation borders that do not intersect. A more detailed view of this effect is shown in Figure 12. It is also possible to select to *Show Interpolated Grid Only* in case the user wants to ignore the polygons (Figure 11e) and visualize the data in a choropleth map. In this mode, each cell is colored separately based on the cell value only, without the polygon generation. All these modes of comparing two models contribute directly to design goal *DG4—Facilitate comparison*.



Compare Two Models (Using Event Data Sets)

Common Settings:
Cell Size: 100
Number of Tiers for First File: 5
Number of Tiers for the Second File: 5

First File Settings:
Cell Bandwith: 300
Type of Kernel: Gaussian

Second File Settings:
Cell Bandwith: 300
Type of Kernel: Gaussian

latitude	longitude	value
-8.0501898	-34.9102173	4851.12077617932
-8.0501898	-34.9102173	4851.12077617932
-8.0619368	-34.9101411	5398.67109634551
-8.0619368	-34.9101411	5398.67109634551
-8.0566072	-34.9117801	3155.89353612167
-8.0566072	-34.9117801	3155.89353612167
-8.0491143	-34.9115054	7189.54248366013
-8.0491143	-34.9115054	7189.54248366013
-8.0491143	-34.9115054	7189.54248366013
-8.0457657	-34.9112523	5230.76923076923
-8.0457657	-34.9112523	5230.76923076923
-8.0457657	-34.9112523	5230.76923076923
-8.0509234	-34.909932	6886.95652173913
-8.0509234	-34.909932	6886.95652173913
⋮	⋮	⋮

latitude	longitude	value
-8.029901	-34.9233392	5823.52941176471
-8.029901	-34.9233392	5823.52941176471
-8.029901	-34.9233392	5823.52941176471
-8.0492751	-34.925252	4767.44186046512
-8.0492751	-34.925252	4767.44186046512
-8.0492751	-34.925252	4767.44186046512
-8.0671622	-34.8773639	23.3333333333333
-8.0671622	-34.8773639	23.3333333333333
-8.0368755	-34.8991636	5080.64516129032
-8.0368755	-34.8991636	5080.64516129032
-8.0368755	-34.8991636	5080.64516129032
-8.0368755	-34.8991636	7156.86274509804
-8.0368755	-34.8991636	7156.86274509804
-8.0367433	-34.8998791	6178.34394904459
⋮	⋮	⋮

Insert First Value File
Insert Second Value File
Model Difference
Display Both Models Overlayed
Model Difference - Use The Same Limits
Input Model Files
Adjust Parameters
Server-side
Generate Grid
Interpolate Values for both models
Calculate the difference between both models
Tiers divided based on division criteria (absolute or percentual)
Generate polygons for each tier
Tiers are divided based on cell density
Generate polygons
Apply colormapping and draw polygons for both models
Model Diagnosis
Apply colormapping and draw polygons
Model Diagnosis
BONGI
N MARTI
MUSTARDINHA
CARNAMIRIM
ROSARINHO
JAQUEIRA
AFLITOS
Both Models Overlayed
Difference Between Two Models

FIGURE 9 Workflow for visualizing two models, whether by showing both overlapped, or by visualizing the difference between them.

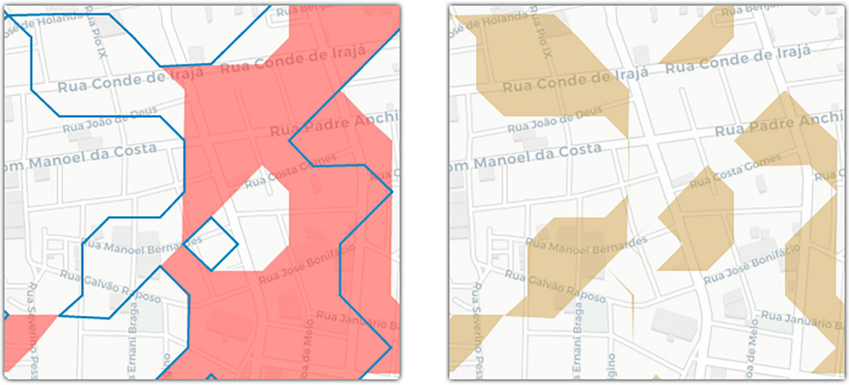


FIGURE 10 Visualizing the total difference between the models in a given tier. On the left: the two models' results overlaid on top of each other in the same tier. On the right: the total difference on that tier, that is, regions present in one model but not in the other. The total area is displayed on the side panel (not shown in the figure).

To analyze the difference between a model and the ground truth (See DT5 in Section 1.1), however, absolute values can be hard to interpret. For that reason, one of the views available is *Percentage Values* (Figures 11f,h), which separates each tier in percentages of the maximum absolute value. *Percentage Fixed Sizes* is a combination of the regular view and percentage values, where instead of percentages of the maximum absolute value, each tier has approximately the same amount of cells. Another particular visualization available is *Absolute Values*, where the user can still use relative values. However, instead of dealing with two color scales and two different tiered polygon visualizations, we combine the positive and negative values using the absolute difference between the models in each region, allowing the system to calculate only one polygon visualization for all differences (as in a change map).

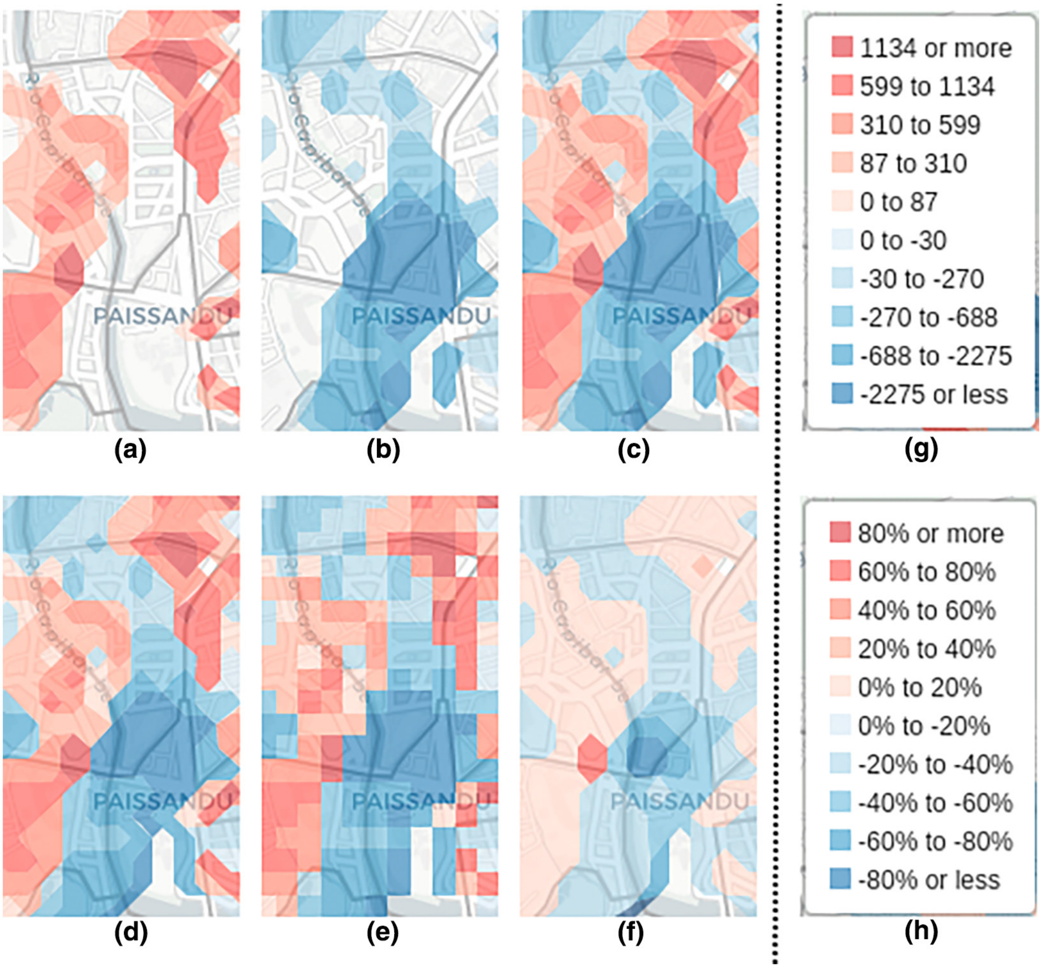


FIGURE 11 The possible ways of visualizing the computed difference between two models. (a) Displaying only positive values. (b) Displaying only negative values. (c) Displaying both positive and negative values. (d) A view showing how the comparison model would look like if negative and positive values were combined in a single pass of our algorithm. (e) A view displaying the interpolated grid without the polygons. (f) A view where each tier corresponds to a percentage of the maximum absolute difference shown in the legend on the right (h). (g) shows the legend used in views (a) to (e). The *Filter* operator is applied in (a) and (b), *Reexpress* is used to change the view between (d), (e), and (f).

2.3 | Summary

Table 2 summarizes how S-VIDIA workflows and features are related with the system's design goals and interaction operators (Roth, 2013) and how they can be used to support the ATs the analysts perform when building and diagnosing spatial interpolation models. In the next section, we present a complete case study that describes the DT and AT being performed in more detail.

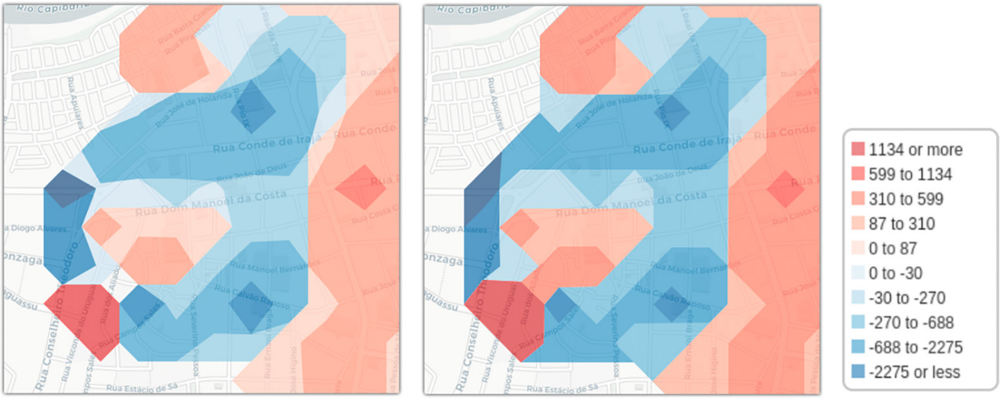


FIGURE 12 Detailed view showing the effects of generating the difference between two models as separate sets and as a single set. On the left: the positive and negative values were rendered separately and the tiers are displayed one over the other. On the right: the positive and negative values were rendered as a single set, so intermediate values between negative and positive are covered by the polygons.

3 | CASE STUDY: PREDICTING REAL ESTATE PRICES

This section describes a case study related to the tasks outlined in Section 1.1 to demonstrate the proposed system's functionality. More specifically, the same machine learning experts who helped to define the DTs shown in Section 1.1 used S-VIDIA to analyze spatial interpolation models that predict real estate prices in a specific region.

3.1 | Setup

For this study, we used two real estate datasets. The first one is from the city of São Paulo (Brazil), which collected real estate advertisements from hundreds of Brazilian websites. In this dataset, each house is represented by six attributes: its selling price in Brazilian currency (reais), latitude, longitude, number of bedrooms, number of parking spaces, and the type of the property (house or apartment). The second dataset (https://github.com/BangLiu/RealEstateModeling/) corresponds to houses of the Canadian province of Alberta. Each property has 13 attributes: effective build year, net area, finished basement, fireplace, assessed value, fully complete, lot size, walkout basement, air conditioning, site coverage, longitude, latitude, and valuation group residential south. We applied Median Absolute Deviation (Leys et al., 2013) on price to remove outliers. For comparison, we create two different predictive models: one based on neural networks (ANN) and another one using linear regression (Montgomery et al., 2021). More specifically, the ANN model predicts the value of a point, applying attention mechanisms to weight neighboring points based on their similarity in terms of features and geographic location. For further details, we refer the reader to Viana and Barbosa (2021). Linear regression is the classical regression model (Montgomery et al., 2021) that assumes a linear relationship between the features and the output value.

Before performing the tasks described in Section 3.2, we prepared the input files for S-VIDIA. The format is standard and corresponds to a file in CSV format structured in three columns. The first two correspond to location information (latitude and longitude), and the third stores the attribute which we want to investigate its spatial behavior. Therefore, each line in the file corresponds to a point on the map. The file corresponding to the city of São Paulo has 68,848 points, while Alberta's file has 6130 points.

TABLE 2 Summary of how S-VIDIA supports analytical tasks, interaction operators (Roth, 2013), and design goals.

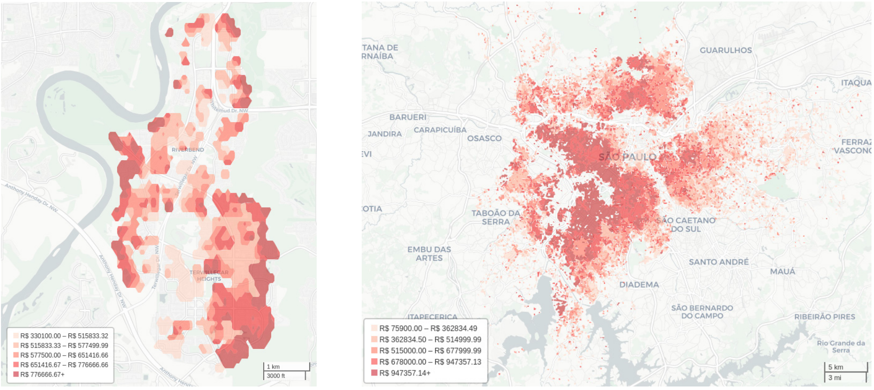
Analytical task	S-VIDIA workflow	S-VIDIA features (*interaction operators*)	Design goals
AT1: Identify spatial regions AT2: Associate attributes for correlation	Visualize a single model	Map view (*Pan and Zoom*), shaded isoline map, change visual interpolation, color, and visualization settings (*Reexpress*, *Resymbolize*)	DG1, DG3
AT3: Delineate spatial clusters	Visualize the difference between two models	Map view (*Pan and Zoom*), display both positive and negative values in shaded isoline map, change visual interpolation, color and visualization settings (*Reexpress*, *Resymbolize*)	DG1, DG3
AT4: Rank areas by attribute	Visualize the difference between two models	Map view (*Pan and Zoom*), display only highest positive or negative values in shaded isoline map (*Filter*), change visual interpolation, color and visualization settings (*Reexpress*, *Resymbolize*)	DG1, DG4
AT5: Compare spatial shapes	Visualize both models overlapped	Map view (*Pan and Zoom*), isoline maps, change visual interpolation, color and visualization settings (*Reexpress*, *Resymbolize*), select specific layers in both models for visualization (*Overlay*)	DG1, DG2, DG3, DG4
AT6: Compare spatial regions by attribute	Visualize both models overlapped	Map view (*Pan and Zoom*), isoline maps, change visual interpolation, color and visualization settings (*Reexpress*, *Resymbolize*), select specific layers in both models for visualization (*Overlay*)	DG1, DG2, DG3, DG4
AT7: Identify attribute value	Visualize a single model	Map view (*Pan*, *Zoom*, *Retrieve*), tooltips, Details panel with histogram and descriptive statistics (*Calculate*)	DG1, DG3
AT8: Compare attribute values	Visualize both models overlapped	Map view (*Pan*, *Zoom*, *Retrieve*), isoline and choropleth maps, color and visualization settings (*Reexpress*, *Resymbolize*) tooltips, Details panel with histogram and descriptive statistics (*Calculate*)	DG1, DG3

3.2 | Performing the analyses

We start our investigation by verifying the geographical distribution of the observations in the datasets (DT1 and AT1 in Section 1.1). In this analysis, we aim to identify, for instance, regions with high and low density of points in order to check their possible impact on the models' performance. In Figure 13, we can visualize the prediction points in polygons with low granularity for the two cities, using the option *Visualize Model* with a bandwidth of 100. For the Alberta dataset, Figure 13a, it is clear that central regions of the map are not well covered, and for the São Paulo dataset, in contrast, the sparse regions are in the outskirts, Figure 13b. The missing data in these datasets might occur due to low sampling of observations in these regions or other natural and artificial landscapes such as mountains, forests, parks, and water bodies. A geographic view that presents a physical map of the region can be helpful to verify if this is, in fact, the case.

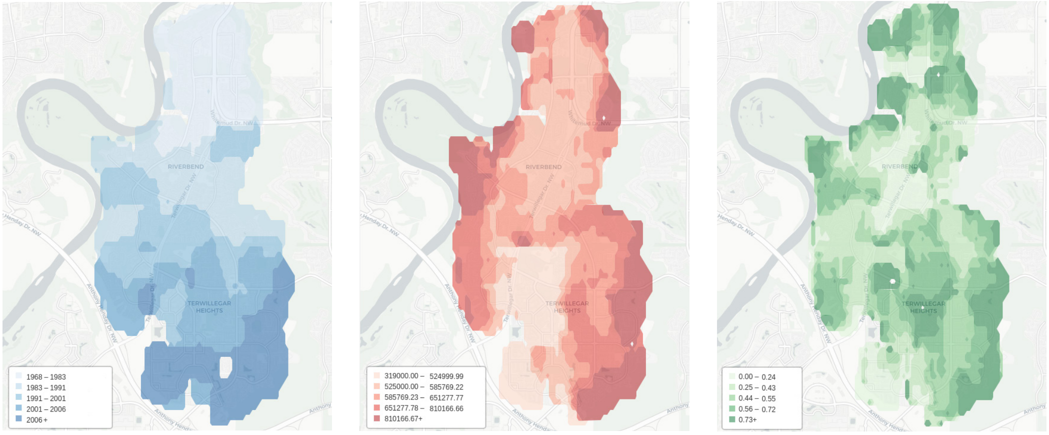
Our subsequent analysis is to verify the quality of the attributes of the models regarding their spatial distribution (DT2 and AT2 in Section 1.1). This verification is possible by inspecting how spatially spread the values of an attribute are. Figure 14 shows the spatial distribution of the attributes year, price, and finished basement on the Alberta dataset. The features year and price present a strong spatial autocorrelation since they have similar values clustered in space, whereas the attribute finished basement has its values more spatially spread. This visual analysis may indicate that price and year are better spatial predictors than the finished basement.

Now, we aim to evaluate the quality of the predictive models by looking at their predictions and the truth values on a holdout set, not used for training the models. For that, first, we examine how much a given predictive model can capture the spatial dependence existing in the target variable (DT3 in Section 1.1). Using the option *Model Difference*, we visualized the spatial distribution of the error of the linear regression model for house price prediction



(a) Alberta **(b)** São Paulo

FIGURE 13 Visualization of predictions on both cities showing the regions with missing data in complete transparency to perform DT1 (and AT1).



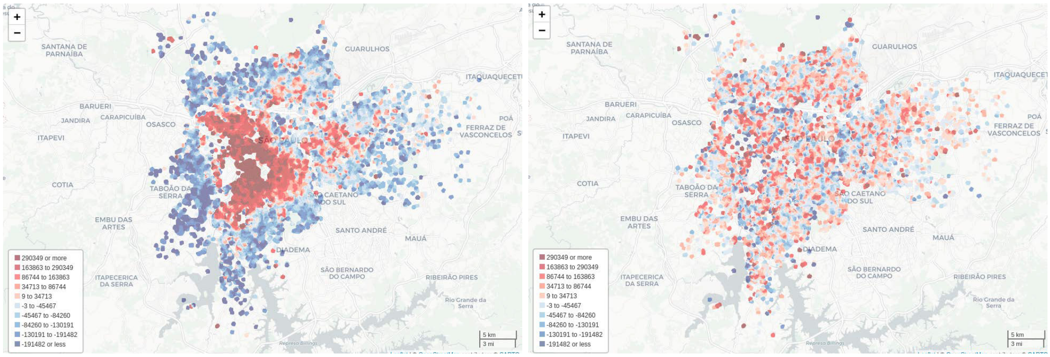
(a) **(b)** **(c)**

FIGURE 14 DT2: Spatial correlation of the attributes: (a) year; (b) price; and (c) finished basement in the Alberta dataset.

in São Paulo, shown in Figure 15a, and the error distribution of the ANN model in Figure 15b. As can be seen, the ANN model is able to better capture the spatial autocorrelation of the house prices than the linear regression model, since its errors are more spatially spread out, that is, they look more random than the ones from the linear regression model.

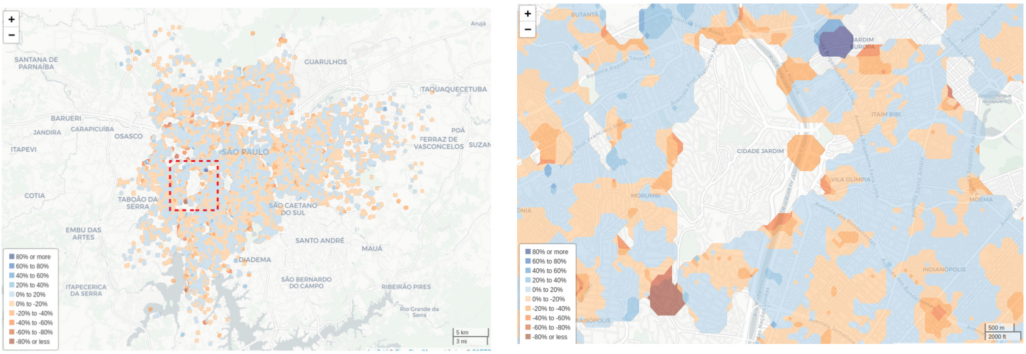
We perform another error-related analysis by finding the regions with high errors to improve the prediction model's quality (DT4 and AT4 in Section 1.1). Figure 16a shows the percentage distribution of the errors of the ANN model in the São Paulo dataset. The darkest red and blue regions represent high error values. They are mainly found in the border areas of the data, suggesting that they might have been caused by the lack of data in these areas. To better illustrate this, we depict in Figure 16b a sub-region of Figure 16a which contains regions with high errors.

Another way to inspect the quality of the model is to compare the polygons concerning the predictive model's results and the ground truth in a specific price range (DT5, AT5, and AT6 in Section 1.1). For that, we use the option *Display Both Models Overlaid*, from the *Model Comparison* feature. The more exact the overlap between the polygons, the better fitting the model in the sub-region under analysis. Figures 17a,b shows polygons that represent this



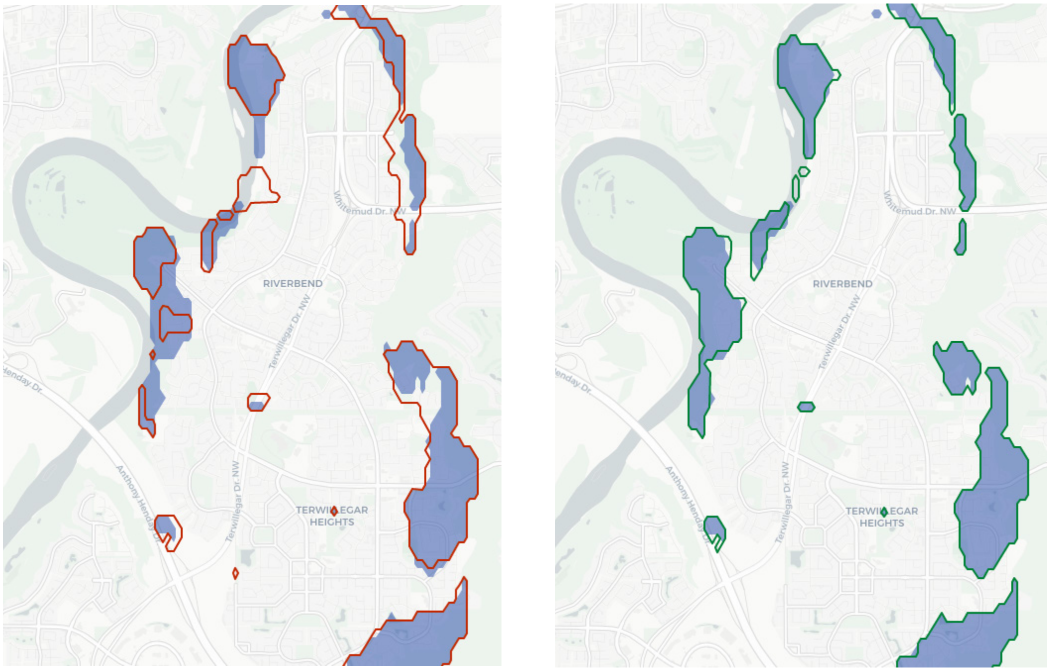
(a) Linear Regression (b) Artificial Neural Network

FIGURE 15 Comparison of the spatial distribution of the errors of the linear regression model (on the left) and the ANN model (on the right). In this visualization, the analyst visualized the computed difference between each model and the ground truth, displaying both positive and negative values (DT3 and AT3).



(a) (b)

FIGURE 16 Overall distribution of the error percentage of the ANN model in the São Paulo city (a) and in a specific region with high error values (b). In this visualization, the analyst visualized the computed difference between the ANN model and the ground truth, displaying both positive and negative values as a percentage of the maximum absolute difference (DT4 and AT4).



(a) Linear Regression

(b) ANN

FIGURE 17 DT5 (AT5 and AT6): Comparison between predictions and the ground truth using polygons in a specific price range. In this visualization, the analyst displayed both models overlapped, selected a single tier to be visualized, and adjusted the colors of the polygons. The ground truth is displayed in solid blue polygons in both (a) and (b), Linear Regression is displayed with unfilled red polygons (a), and ANN is displayed with unfilled green polygons (b).

comparison in a particular region of Alberta for the Linear Regression and ANN model, respectively. The blue-filled polygons represent the ground truth, while the unfilled ones, the model's prediction. The more fitted the polygons are, the better the model is. By this analysis, we can visually understand that in this particular sub-region, the ANN model presents better quality than Linear Regression since there is a more exact overlap between the filled and unfilled polygons.

During the analyses, we might want to inspect the raw values of the model's predictions and the truth values in order to understand, for instance, why a model achieves a high error in a particular region (DT6 in Section 1.1). Figure 7 presents an example of this functionality. It shows not only the values for a specific polygon, but also its distribution in a histogram.

After using S-VIDIA to analyze attributes and models that deal with spatially correlated phenomena, we identified the need to investigate solutions to mitigate some of the model's weaknesses. For instance, the perception that the model has significant errors in the bordering regions (internal holes) led us to search for solutions that deal with this phenomenon such as spline-based methods, as suggested in Sangalli et al. (2013). Another possible direction for investigation is creating an ensemble of the models oriented to geographic location, as S-VIDIA allows us to identify, in a straightforward way, the model that presents greater precision in a given region of interest.

4 | DISCUSSION AND LIMITATIONS

The main goal of our approach is to facilitate visual inspection and comparison of spatial interpolation models. To support this, we defined four design goals which imposed some limitations.

As explained in the Introduction, we preferred adopting intrinsic coincident displays to show model differences to reduce cognitive effort while facilitating comparisons. The main reason for this design choice was that comparing models was the most important task for the specialists; in fact, they performed the comparison tasks easily. Even though we tried to minimize its drawbacks, this approach could be better, and other approaches, such as extrinsic coincident displays and animation, could be added to the system.

Although our tool fully supports visually comparing two models, comparing three models such as performed in Figures 15 and 17 was also possible by allowing synchronization of value limits, map views (camera sync) and displaying them side by side in different windows.

Moreover, S-VIDIA only supports loading a single variable at a time. This was a decision to make our approach more focused on the geospatial aspect of the visualization and to satisfy the design goal DG2—Facilitate data loading. Loading more than one variable at once could make the interface unnecessarily more complex. However, we are planning on supporting loading more than one variable and visualizing them using other visual techniques in addition to the histogram.

Another limitation in our solution is that not all interaction operators were implemented. Operators such as *Edit*, *Annotate*, *Arrange*, *Sequence*, *Reproject*, *and Search* were not necessary for the specialists' daily usage. However, adding support for these operators is mandatory to make S-VIDIA even more useful and general to other users. For example, after using the system for a while, the specialists suggested that annotations would be important for communicating the analysis results. We also plan on expanding operators such as *Filter* to include more parameters.

We are also aware that the basemap in S-VIDIA's map view, which is based on Leaflet (https://leafletjs.com/) and OpenStreetMap (https://www.openstreetmap.org/), use the Web Mercator projection. Although this projection is convenient, it can introduce visual distortion of the relative sizes of color regions at a regional-level scale. The specialists work at a city-level scale, so this projection does not compromise their analysis. For other users whose data are at a higher-level scale, more appropriate projections can be used by exchanging the map library used in the system.

The visual spatial interpolator applies interpolation to the data before visualization which may confuse users. However, as we wanted to facilitate comparison, this step was required to generate the polygon tiers. If users want to avoid this extra interpolation, they can input the computed grid directly to the system and it will be used for the polygon generation.

Although we only presented in this work a case study for house price prediction, S-VIDIA can be employed by researchers and practitioners in any other spatial interpolation problem such as precipitation, temperature or air quality estimation to name a few.

As we wanted our approach to be model agnostic so it could be used in more different scenarios, the visualization techniques available for inspecting the data are straightforward. Despite its simplicity, combining interactivity with the polygon-tier visualization provided great flexibility for our collaborator specialists. They were used to conventional GIS applications to visualize their models' results but needed help to compare them with the ground truth or other models easily. In the case of sparse regions in which the specialists were particularly interested, S-VIDIA allowed them to geographically visualize such regions to verify the possible occurrence of poor performance. Support for more complex information besides model differences, such as uncertainty, is also desired, which brings more challenges for designing and implementing visualization techniques.

More sophisticated approaches could be implemented in S-VIDIA but would require knowing the internals of the machine learning models to provide more tailored diagnostics.

5 | CONCLUSIONS

In this article, we presented S-VIDIA, a visual analytics model-agnostic solution for facilitating the comparison and refinement of interpolation models at different levels of granularity, using interactive visualization techniques. To support this, we defined a set of design goals and the ATs implemented by our approach.

One of the main contributions of our approach is to facilitate geospatial model comparisons using an intrinsic coincident display with customizable shaded isoline and choropleth maps. Visualizing the difference directly on the map does not cause an extra cognitive load for the analysts as they do not need to rely on their memory to make the comparison. Besides, S-VIDIA's interactivity and filtering controls reduce the amount of information displayed on the map, which could be another source of cognitive effort.

The ATs were defined based on the general problem of building and inspecting a new model for spatial interpolation, independently of the techniques used to build the model and the application area. We demonstrated our approach's utility with a case study in which the specialists used S-VIDIA to diagnose and improve a spatial model for predicting real estate prices. However, we believe that our system is a step toward responding to the call to build tools that enable users to obtain insights that directly support decision-making and so, we envision S-VIDIA being used in other real-world scenarios involving the inspection of spatial interpolation models. For example, for estimating new cases of a given disease, such as influenza or COVID-19 in health science; for estimating soil properties, daily precipitation, temperature in environmental science; for estimating evapotranspiration (the consumption of water by a crop), crop yield in agriculture. As S-VIDIA is model agnostic, the geolocated datasets generated in those different scenarios can be easily entered into the system and visualized.

As future work, in addition to improve the system according to the limitations discussed in Section 4, we plan to support additional evaluation measures defined by the analysts, and also collaborate with other specialists to explore other real-world cases in S-VIDIA.

ACKNOWLEDGMENTS

André de Oliveira Bastos was partially supported by a Capes Scholarship. Emanuele Santos was partially supported by a FUNCAP scholarship.

CONFLICT OF INTEREST STATEMENT

We declare that we have no financial or personal relationships with people or organizations that could have inappropriately influenced the work reported in this article.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Emanuele Santos https://orcid.org/0000-0002-2806-4589

ENDNOTES

[1] By diagnostics, we refer to the broader meaning of “techniques of careful examination.” Therefore, this article uses the concepts diagnosis (diagnostics), inspection, and analysis interchangeably.

[2] “Blind” spots in our context are spatial regions where there are few or no occurrences of observations, which we call sparse regions.

REFERENCES

Adrienko, N., & Adrienko, G. (2011). Spatial generalization and aggregation of massive movement data. *IEEE Transactions on Visualization and Computer Graphics*, *17*(2), 205–219. https://doi.org/10.1109/TVCG.2010.44

Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, *66*, 55–69. https://doi.org/10.1016/j.csda.2013.03.016

Bachoc, F. (2018). Asymptotic analysis of covariance parameter estimation for Gaussian processes in the misspecified case. *Bernoulli*, *24*(2), 1531–1575. https://doi.org/10.3150/16-BEJ906

Beecham, R., Dykes, J., Slingsby, A., & Turkay, C. (2015). Supporting crime analysis through visual design. Vis 2015.

Brewer, C. (2009). *Colorbrewer 2.0*. https://colorbrewer2.org/

Brus, D., De Gruijter, J., Marsman, B., Visschers, R., Bregt, A., Breeuwsma, A., & Bouma, J. (1996). The performance of spatial interpolation methods and choropleth maps to estimate properties at points: A soil survey case study. *Environmetrics*, *7*(1), 1–16. https://doi.org/10.1002/(SICI)1099-095X(199601)7:1<1::AID-ENV157>3.0.CO;2-Y

Bui, D., Tuan, T., Klempe, H., Pradhan, B., & Revhaug, I. (2015). Spatial prediction models for shallow landslide hazards: A comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree. *Landslides*, *13*, 1–18. https://doi.org/10.1007/s10346-015-0557-6

Caruso, C., & Quarta, F. (1998). Interpolation methods comparison. *Computers & Mathematics with Applications*, *35*(12), 109–126. https://doi.org/10.1016/S0898-1221(98)00101-1

Chainey, S., Tompson, L., & Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, *21*, 4–28. https://doi.org/10.1057/palgrave.sj.8350066

Chiu, S.-T. (1991). Bandwidth selection for kernel density estimation. *The Annals of Statistics*, *19*(4), 1883–1905. https://doi.org/10.1214/aos/1176348376

Chung, C.-J., & Fabbri, A. (2003). Validation of spatial prediction models for landslide hazard mapping. *Natural Hazards*, *30*, 451–472. https://doi.org/10.1023/B:NHAZ.0000007172.62651.2b

Ciski, M., Rzasa, K., & Ogryzek, M. (2019). Use of gis tools in sustainable heritage management—The importance of data generalization in spatial modeling. *Sustainability*, *11*, 5616. https://doi.org/10.3390/su11205616

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, *22*(3), 239–252. https://doi.org/10.1007/BF00889887

de Queiroz Neto, J. F., Santos, E., & Vidal, C. A. (2016). MSKDE—Using marching squares to quickly make high quality crime hotspot maps. *29th SIBGRAPI Conference on Graphics, Patterns and Images*, San Jose dos Campos, SP, Brazil (pp. 305–312). https://doi.org/10.1109/SIBGRAPI.2016.049

de Queiroz Neto, J. F., Santos, E., Vidal, C. A., & Ebert, D. S. (2020). A visual analytics approach to facilitate crime hotspot analysis. *Computer Graphics Forum*, *39*(3), 139–151. https://doi.org/10.1111/cgf.13969

Drawve, G. (2014). A metric comparison of predictive hot spot techniques and RTM. *Justice Quarterly*, *33*, 1–29. https://doi.org/10.1080/07418825.2014.904393

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, *14*(1), 153–158. https://doi.org/10.1137/1114019

Feizizadeh, B., Roodposhti, M., Blaschke, T., & Aryal, J. (2017). Comparing gis based support vector machine kernel functions for landslide susceptibility mapping. *Arabian Journal of Geosciences*, *10*, 122. https://doi.org/10.1007/s12517-017-2918-z

Ferreira, N., Poco, J., Vo, H. T., Freire, J., & Silva, C. T. (2013). Visual exploration of big spatiotemporal urban data: A study of New York city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2149–2158. https://doi.org/10.1109/TVCG.2013.226

Flanagan, D., Frankenberger, J., Cochrane, T., Renschler, C., & Elliot, W. (2013). Geospatial application of the water erosion prediction project (WEPP) model. *Transactions of the ASABE*, *56*, 591–601. https://doi.org/10.13031/2013.42681

García, G., Silveira, J., Poco, J., Paiva, A., Nery, M. B., Silva, C. T., Adorno, S., & Nonato, L. G. (2021). Crimanalyzer: Understanding crime patterns in São Paulo. *IEEE Transactions on Visualization and Computer Graphics*, *27*(4), 2313–2328. https://doi.org/10.1109/TVCG.2019.2947515

Gleicher, M., Albers, D., Walker, R., Jusufi, I., Hansen, C. D., & Roberts, J. C. (2011). Visual comparison for information visualization. *Information Visualization*, *10*(4), 289–309. https://doi.org/10.1177/1473871611416549

Glinz, M. (2007). On non-functional requirements. *15th IEEE International Requirements Engineering Conference*, New Delhi, India (pp. 21–26). IEEE. https://doi.org/10.1109/RE.2007.45

Golebiowska, I., Korycka-Skorupa, J., & Slomska-Przech, K. (2021). Common thematic map types. In J. P. Wilson (Ed.), *The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2021 Edition)*. UCGIS. https://doi.org/10.22224/gistbok/2021.2.7

Hamylton, S. M., Hedley, J. D., & Beaman, R. J. (2015). Derivation of high-resolution bathymetry from multispectral satellite imagery: A comparison of empirical and optimisation methods through geographical error analysis. *Remote Sensing*, *7*(12), 16257–16273. https://doi.org/10.3390/rs71215829

Kinkeldey, C., MacEachren, A. M., & Schiewe, J. (2014). How to assess visual communication of uncertainty? A systematic review of geospatial uncertainty visualisation user studies. *The Cartographic Journal*, *51*(4), 372–386. https://doi.org/10.1179/1743277414Y.0000000099

Kong, K., Ma, Y., Ye, C., Lu, J., Chen, X., Zhang, W., & Chen, W. (2018). A visual analytics approach for traffic flow prediction ensembles. *26th Pacific Conference on Computer Graphics and Applications: Short Papers*, Hong Kong (pp. 61–64). ACM.

Krisp, J. M. (2010). Planning fire and rescue services by visualizing mobile phone density. *Journal of Urban Technology*, *17*(1), 61–69. https://doi.org/10.1080/10630731003597330

Krisp, J. M., Peters, S., Murphy, C. E., & Fan, H. (2009). Visual bandwidth selection for kernel density maps. *Photogrammetrie-Fernerkundung-Geoinformation*, *2009*(5), 445–454. https://doi.org/10.1127/1432-8364/2009/0032

Krisp, J. M., & Špatenková, O. (2010). Kernel density estimations for visual analysis of emergency response data. In M. Konecny, S. Zlatanova, & T. L. Bandrova (Eds.), *Geographic information and cartography for risk and crisis management: Towards better solutions* (pp. 395–408). Springer. https://doi.org/10.1007/978-3-642-03442-8_27

Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, *49*, 764–766. https://doi.org/10.1016/j.jesp.2013.03.013

Li, J., & Heap, A. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, *53*, 173–189. https://doi.org/10.1016/j.envsoft.2013.12.008

Liao, Y., Li, D., & Zhang, N. (2018). Comparison of interpolation models for estimating heavy metals in soils under various spatial characteristics and sampling methods. *Transactions in GIS*, *22*(2), 409–434. https://doi.org/10.1111/tgis.12319

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *ACMSIGGRAPH Computer Graphics*, *21*(4), 163–169. https://doi.org/10.1145/37401.37422

MacEachren, A. M., Brewer, C. A., & Pickle, L. W. (1998). Visualizing georeferenced data: Representing reliability of health statistics. *Environment and Planning A: Economy and Space*, *30*(9), 1547–1561. https://doi.org/10.1068/a301547

Machwitz, M., Haß, E., Junk, J., Udelhoven, T., & Schlerf, M. (2019). CropGIS—A web application for the spatial and temporal visualization of past, present and future crop biomass development. *Computers and Electronics in Agriculture*, *161*, 185–193. https://doi.org/10.1016/j.compag.2018.04.026

Malik, A., Ebert, D. S., Maciejewski, R., & Collins, T. F. (2010). Visual analytics law enforcement toolkit. In *2010 IEEE International Conference on Technologies for Homeland Security* (pp. 222–228). Waltham, MA: IEEE. https://doi.org/10.1109/THS.2010.5655057

Mesic Kiš, I. (2016). Comparison of ordinary and universal kriging interpolation techniques on a depth variable (a case of linear spatial trend), case study of the Šandrovac field. *Rudarsko Geolosko Naftni Zbornik*, *31*, 41–58. https://doi.org/10.17794/rgn.2016.2.4

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). *Introduction to linear regression analysis*. John Wiley & Sons.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, *33*(3), 1065–1076. https://doi.org/10.1214/aoms/1177704472

Plouffe, C. C., Robertson, C., & Chandrapala, L. (2015). Comparing interpolation techniques for monthly rainfall mapping using multiple evaluation criteria and auxiliary data sources: A case study of Sri Lanka. *Environmental Modelling & Software*, *67*, 57–71. https://doi.org/10.1016/j.envsoft.2015.01.011

Roth, R. E. (2013). An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2356–2365. https://doi.org/10.1109/TVCG.2013.130

Sangalli, L. M., Ramsay, J. O., & Ramsay, T. O. (2013). Spatial spline regression models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *75*(4), 681–703. https://doi.org/10.1111/rssb.12009

Sedlmair, M., Meyer, M., & Munzner, T. (2012). Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, *18*(12), 2431–2440. https://doi.org/10.1109/TVCG.2012.213

Shi, L., Huang, C., Liu, M., Yan, J., Jiang, T., Tan, Z., Hu, Y., Chen, W., & Zhang, X. (2020). UrbanMotion: Visual analysis of metropolitan-scale sparse trajectories. *IEEE Transactions on Visualization and Computer Graphics*, *27*(10), 3881–3899. https://doi.org/10.1109/TVCG.2020.2992200

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*, Boulder, CO (pp. 336–343). IEEE. https://doi.org/10.1109/VL.1996.545307

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman & Hall.

Thakali, L., Kwon, T. J., & Fu, L. (2015). Identification of crash hotspots using kernel density estimation and kriging methods: A comparison. *Journal of Modern Transportation*, *23*(2), 93–106. https://doi.org/10.1007/s40534-015-0068-0

Thomas, J., & Cook, K. A. (2005). *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press.

Viana, D., & Barbosa, L. (2021). Attention-based spatial interpolation for house price prediction. *29th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* Beijing, China (pp. 540–549). ACM.

Wang, H., Yilihamu, Q., Yuan, M., Bai, H., Xu, H., & Wu, J. (2020). Prediction models of soil heavy metal(loid)s concentration for agricultural land in Dongli: A comparison of regression and random forest. *Ecological Indicators*, *119*, 106801. https://doi.org/10.1016/j.ecolind.2020.106801

Wang, Z., Lu, M., Yuan, X., Zhang, J., & Wetering, H. V. D. (2013). Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization & Computer Graphics*, *19*(12), 2159–2168. https://doi.org/10.1109/TVCG.2013.228

Yang, X., & Hodler, T. (2000). Visual and statistical comparisons of surface modeling techniques for point-based environmental data. *Cartography and Geographic Information Science*, *27*, 165–175. https://doi.org/10.1559/152304000783547911

How to cite this article: Oliveira Bastosde, A. et al. (2023). Visual analytics for diagnosing spatial interpolation models. *Transactions in GIS*, *27*, 364–387. https://doi.org/10.1111/tgis.13028