

Predominance Tag Maps

Martin Reckziegel, Muhammad Faisal Cheema, Gerik Scheuermann, *Member, IEEE*, Stefan Jänicke

Abstract—A *predominance map* expresses the predominant data category for each geographical entity and colors are used to differentiate a small number of data categories. In *tag maps*, many data categories are present in the form of different tags, but related tag map approaches do not account for predominance, as tags are either displaced from their respective geographical locations or visual clutter occurs. We propose *predominance tag maps*, a layout algorithm that accounts for predominance for arbitrary aggregation granularities. The algorithm is able to utilize the font sizes of the tags as visual variable and it is further configurable to implement aggregation strategies beyond visualizing predominance. We introduce various measures to evaluate numerically the qualitative aspects of tag maps regarding local predominance, global features, and layout stability and we comparatively analyze our method to the tag map approach by Thom et al. [1] on the basis of real world data sets.

Index Terms—Geospatial visualization, point-based data, data aggregation

1 INTRODUCTION

ACCORDING to Slocum et al. [2], a thematic map is "used to emphasize the spatial pattern of one or more geographic attributes, such as population density, family income, and daily temperature maximums." In order to relate geospatial phenomena to real places, a thematic map is laid atop a general-reference map. Different techniques have been developed, e.g., choropleth maps that color polygonal regions after mapping the values of the observed attribute(s) to color, or symbol maps where for each geo-referenced object a symbol is drawn at its dedicated location, and the object's attributes are expressed using visual channels such as color, shape or size. In this paper, we focus on the thematic mapping techniques predominance maps and tag maps, which we combine as *predominance tag maps*.

Consider a geo-referenced dataset that maps for each data item a specific data category to a specific geographical location. The major goal of a *predominance map* [3] is to show which data category is predominant for each geographical entity, e.g., a polygonal district or a 2-dimensional raster cell. Thus, the totals of all data categories occurring within a geographical entity are determined, and the visual representative, e.g., a polygon or a symbol, receives the color associated with the most frequent, that is, the predominant category. The magnitude of the predominance can be expressed with varying symbol size, or with various saturation levels. As the human ability to distinguish colors is limited, the major drawback of this method is that the number of distinct categories needs to be small in order to keep the map examinable. An example is the forest type predominance map proposed by Ruefenacht et al. [4] that gives an overview of predominant forest cover in the United States. For displaying the result, 142 forest types were merged into 28 forest type groups, which are displayed using 19 different colors. So, different forest types are associated with similar colors or the same forest type

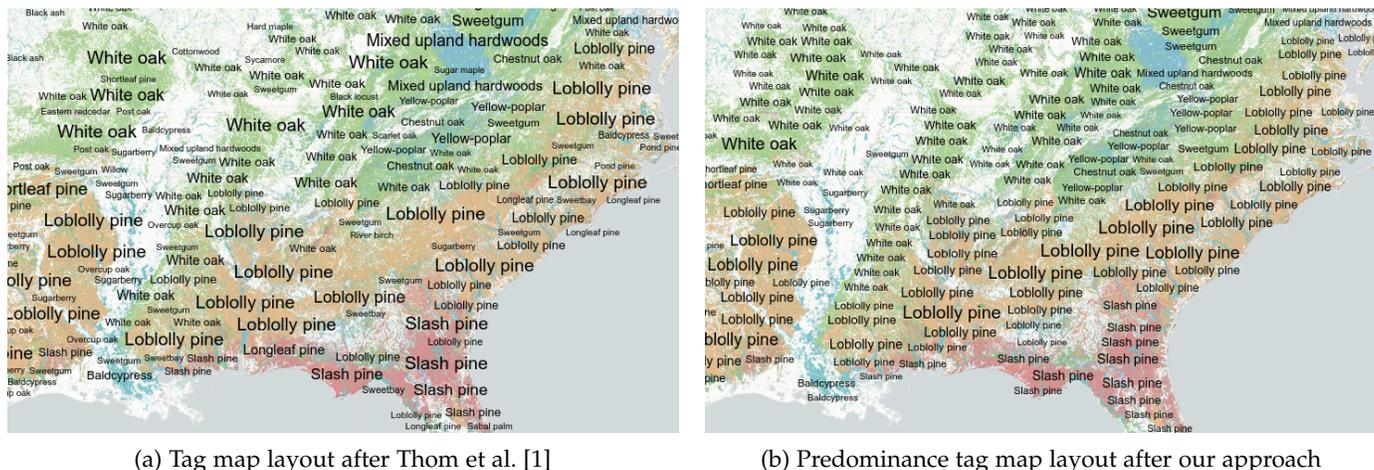
group, which decreases the diversity of the predominance map.

A *tag map* is a tag cloud where tags are placed depending on their geographic location [5]. Having a dataset as defined above, drawing a tag¹ (the data category) for each data item may lead to an enormous amount of overlaps, rendering the tag map illegible. Different strategies can be applied to resolve occlusions algorithmically. Thom et al. [1] propose a strategy that removes the smaller of each two occluding equal tags in a first iteration. In a second iteration, the largest of occluding unequal tags is fixed at its respective position, and smaller tags are displaced following an Archimedian spiral. Unfortunately, this approach leads to tags drifting apart from their original geospatial reference points, which harms the general principle of preventing from ambiguous label placement [6]. As a consequence, tags are not necessarily representatives of the predominant data category of the area they are placed. But, the major opportunity of using a *tag map* as opposed to a predominance map is the ability to distinguish numerous data categories easily. This can be seen in figure 1a, where we applied Thom et al.'s approach to the US Forest Types dataset (see section 5.2 for details). Using both Ruefenacht et al.'s predominance map and Thom et al.'s tag map as overlays illustrates a larger capacity of displaying diversity using tags over colors. But, the layout strategy displaces tags, so that tags often intersect forest area boundaries, or they are even placed at locations without data points.

Predominance tag maps account for predominance in tag maps. Our result for the US Forest Types dataset is shown in figure 1b. In contrast to the output produced by Thom et al.'s method, our tag distribution resembles the underlying color-coded predominance map more accurately. In this paper, we propose a method that computes predominance tag maps ensuring a monotonic relationship between font size and frequency typical for tag clouds. This method is abstracted to be adaptable to applications outside the scope

• M. Reckziegel, M.F. Cheema, G. Scheuermann, and S. Jänicke are with Leipzig University. E-mails: {reckziegel,faisal,scheuermann,stjaenicke}@informatik.uni-leipzig.de

1. We use the term *tag* for a rendered textual item on the map that represents either a single label or an aggregate of multiple labels.

Fig. 1: US Forest Types (*us*) dataset tag maps

of visualizing predominance. We further define measures to estimate the quality of tag maps regarding predominance, the capacity of displaying the dataset’s inherent categorial diversity, as well as layout stability. Using these measures, we comparatively analyze the results of our approach and Thom et al.’s method on the basis of different real world datasets.

2 RELATED WORK

Our work on elaborating a predominant tag map layout is related to three different aspects of research. We first take a look at related geovisualization techniques for geospatial point datasets. Then, we outline general information about tag clouds, and finally, we discuss related tag mapping approaches combining geovisualizations and tag clouds.

2.1 Visualizing Geospatial Point Data

Geospatial data is collected by the continuing efforts of scientific communities, government agencies and private companies. Also, crowdsourcing approaches [7] lead to growing amounts of geographic information [8]. Different geospatial data types exist, e.g., line or polygon data. In the following, we will focus on point data in the plane, where each observed entity has a geo-reference in the form of a latitude/longitude-pair.

When large geospatial point datasets shall be rendered on a map, one has to overcome overplotting issues in order to support the sensemaking process with maps. Traditionally, scatterplots solve this problem with top-down binning approaches [9], [10]—the plane is split into bins, and bin coloring reflects the number of points in a bin. Binning is also applied in geo-applications [11], but the result can be misleading due to the pre-defined bin segmentation that might split existing clusters into distinct bins [12]. In cartography, choropleth maps represent a sophisticated binning strategy using geographical regions as bins [2]. They are widely used and studied. Most recently, Zhang et al. [13] evaluated the stability of clusterings in choropleth maps with respect to geographical variations and Beecham et al. [14] analyzed how spatial auto-correlation in differing choropleth maps affects their perception.

Without pre-defined regions like in the case of choropleth maps, glyph-based approaches like dot maps, where a dot represents a single data point, or proportional symbol maps with different-sized glyphs reflecting weight values have been established. An overview is given by Slocum et al. [2]. Typically, closely located points form clusters which are drawn in circles with varying diameters in order to gain an uncluttered layout [15], [16]. A yet similar approach is given by Scheepens et al. [17] who uses pie charts to visualize the categorial distribution within the clusters. In contrast, Aman et al. [18] use glyphs to visualize temporal aspects. Tag maps as the subject of this paper are rather abstract glyph-based approaches. Each data point is represented by a tag, and in order to gain a legible map overlay, tag occlusions must be treated.

2.2 Tag Cloud Visualizations

The primary purpose of tag clouds is to present a visual summary of textual data [19]. First introduced by Stanley Milgram’s mental map of Paris [20] in 1976, tag clouds later became popular in the social web community. Although originally used for non-specific information discovery, tag clouds can also be used to support analytical tasks such as the examination of text collections [21]. Furthermore, tag clouds obtained wide acceptance as interfaces for navigation purposes on databases [22]. Traditionally, a tag cloud is a simple list of words placed on multiple lines, either ordered alphabetically or by the importance of a tag, which is encoded by variable font size [23]. Portals such as ManyEyes could be used to create such kind of tag cloud visualizations on demand [24]. A user study on the utility of tag clouds revealed that the usual alphabetic order is not obvious for the observer, but tag clouds are generally seen as a popular social component [22]. Potentially, this was one of the reasons that later more sophisticated tag cloud layout approaches were developed, which rather emphasized aesthetics than meaningful orderings, e.g., Wordle [25] is a popular technique for generating visually appealing tag clouds on demand.

Tag clouds are used for a wide range of applications. This includes their usage in text analysis environments,

e.g., to encode the number of word occurrences within a selected section of a text, a whole document or an entire text corpus [26], [27]. Also, for the analysis of topic modeling results, the application of tag clouds have been proven useful, e.g., for analyzing the evolution of topics over time [28], [29], tag clouds serve to explore the temporal change of a topic's terminology. In contrast, some tag cloud approaches illustrate trends in a text corpus. Parallel Tag Clouds generate alphabetically ordered tag lists as columns for a number of time slices and highlight the temporal evolution of a tag placed in various columns on demand [30]. SparkClouds attach a graph showing the tag's evolution over time [31]. Hinrichs et al. [32] links tag clouds to a classification schema in the form of a tree structure to help humanities scholars get access to texts of a speculative fiction anthology corpus. Semantic tag clouds support the detection of tags belonging to a certain topic [33] by placing related tags close to each other in visual groups [34], [35]. Force directed approaches, where semantically close terms attract each other, are quite popular for this task [36], [37].

The foremost difference of tag maps is that each tag to be drawn has a predefined position—which is not the case for the above mentioned tag cloud layout approaches—, and a displacement of the tag inevitably leads to inhibiting the reliability of the tag map.

2.3 Surveyed Tag Map Approaches

A number of methods combine geovisualizations with tag clouds—denoted as *tag maps* [5]—in order to visually link space to its related topics. In the following, we distinguish two kinds of tag maps. The methods of the first kind are mainly driven by tag cloud layout approaches, thus, the geospatial reference plays secondary role. By contrast, methods of the second kind favor placing tags at their dedicated locations, and tag cloud strategies are rather applied in the form of a postprocessing step. We did not consider novel labeling techniques, [38], and methods where tag clouds are interactively superimposed for a small subset of data points, e.g., [16] and [39].

2.3.1 Tag-cloud-driven Tag Maps

Rolled-out Wordles, presented by Strobelt et al. [40], are a heuristic method to remove overlaps in tag clouds. An example is provided illustrating overlap removal of tags in the United Kingdom, but the proposed strategy entirely disregards geographical relations. Hahmann et al. [41] provide a yet comparable approach that roughly considers geographical locations. Dependent on a given set of tags for a specific region, they use an external word cloud processor that computes a tag cloud used as a thematic layer on top of the map. But this approach leads to misinterpretations as tags are arbitrarily placed in their respective region, thus, a tag is not directly associated to its location. Some approaches generate tag maps within a polygonal shape that represents a political entity (e.g., district, region, country) with a space filling strategy [42]. One example is Taggram [43], which does not take the tags' geospatial information into account. Instead, the most frequent tag is positioned in the shape's center, and the remaining tags are iteratively placed on vertical layers as long as free space is

available. As Taggrams, Geo Word Clouds aim to entirely fill a given shape of a political boundary with tags mimicking a visually appealing infographic [44]. But unlike Taggrams, here, tags are placed as near as possible to their original position within the shape. Based on a k-means clustering of geospatial data points, the tags representing different clusters are subsequently placed by decreasing font size. Initial font sizes are set in relation to the given shape area, tags are rotated as necessary, and if a tag cannot be placed without occluding an already positioned tag, its size is reduced. Thus, Geo Word Clouds do not guarantee the major property of tag clouds, that is, that a bigger sized tag is always a tag with a higher frequency.

None of the tag-cloud-driven tag map approaches is directly comparable to our method, as placing coherently-sized tags at their designated geospatial coordinates is not focused. But, having an arbitrary set of tags to be placed on a map and a predefined geospatial subdivision, e.g., all European countries, Taggram as well as Geo Word Cloud could be applied to each area in order to make global topical relations seizable. Still, local structures would remain fuzzy. A further drawback of using geospatial subdivisions to generate a tag map are statistical biases that occur when aggregating data into predefined areas, a problem known as the modifiable areal unit problem (MAUP) [45], e.g., numerous phenomena (meteorological, geological, cultural) are not necessarily bound to predefined areas. Another argument against a tag map algorithm on the basis of an existing geospatial subdivision is that they are not available for all magnifications, e.g., for maritime phenomena geospatial subdivisions are at most roughly given.

2.3.2 Location-driven Tag Maps

The first paper aiming to put the tags of geo-referenced photos on their designated map locations introduced the term *tag maps* [5]. In order to provide a geospatial summary of Flickr photos, a hierarchical clustering is first performed. Second, the resulting clusters are scored according to specific properties, e.g., the number of photos in a cluster, and for each cluster a representative tag is placed in the geospatial centroid of the cluster's associated points. This way, tags of adjacent clusters may occlude. A similar approach, the World Explorer, is presented by Ahern et al. [46], who chose a k-means instead of a hierarchical clustering approach. But the drawn "primary tags" of the clusters likely occlude each other—an issue well documented by Slingsby et al. [47] who compare the results of the Jaffe et al. [5] method to using Google Earth², which reduces the number of occlusions at an expense of reducing the overall number of tags to be shown by applying a *culling* strategy. The drawback of this approach is apparent when zooming into a dedicated region, which occasionally completely changes the shown tag set. Slingsby pointed out that both approaches are limited in terms of map legibility concerning positional accuracy and data omission, and he asks for future consideration. Thom et al. [1] addressed this legibility issue for visualizing geo-referenced Twitter messages. Also using a k-means like clustering approach, occluding tags are recognized and resolved. Equally labeled tags are merged, whereas unequally

2. <https://www.google.com/earth>

labeled tags are moved following an Archimedean spiral originating from the center of the tag. When the tag cannot be placed in a certain distance to the spiral's origin, it is not drawn.

As opposed to tag-cloud-driven tag maps, location-driven tag maps try to place tags as close as possible to their respective locations, keeping the font sizes coherent throughout the whole layout. Only one existing method described by Thom et al. [1] is able to fulfill this goal without occlusions, but by displacing tags from their aggregation areas. Referring to predominance maps [3] introduced before, this displacement dissolves the linkage between tag and location the way that the tag does not necessarily reflect the predominant category of the data items located in an aggregation area centered around the tag. By accounting for predominance in tag maps, predominant tags would be drawn centered at the corresponding aggregation area, and all non-predominant tags would be omitted. We have not found an occlusion-free location-driven method accounting for predominance. In order to close this gap, we present a layout framework accounting for this linkage between tag and location in the next section and describe in section 4 an implementation thereof to generate *predominance tag maps*.

3 LAYOUT METHOD

In this chapter we present an occlusion-free location-driven tag map algorithm with a tight spatial connection between the drawn tags and the underlying data. Concluding from the Gestalt principle of proximity [48] we assume that locations of areas where no glyph is rendered in a tag map layout are perceived as being associated to their nearest tag. If the nearest tag to such location is very distant, i.e., more distant than the average size of all tags on the map, we assume that this location is perceived as not relating to any tag, but for simplicity we ignore that assumption. Following our simplified perception model, an optimal aggregation strategy would be context-sensitive. The aggregation area of one tag would depend on its size and the distance to its neighbors. Furthermore, the sizes of the tags on the map vary because of different word lengths as well as different font sizes used. In an overlap-free tag map, the distance between two visible neighboring tags depend on their sizes. But to calculate those sizes, their aggregation areas need to be known, and these depend again on their size and distance, resulting in a complex optimization problem. We simplify this problem by using an aggregation area which centers in the middle of a tag but that is not dependent on the location of neighboring tags. We will show in section 5.5 that this simplification improves performance in terms of predominance compared to the approach by Thom et al. [1] even under our perceptual assumption.

The input of the layout method is the set of data points P . Each data point $p \in P$ is associated with a label, and all labels constitute the set of label categories L . The data points are geo-referenced, so that for all $p \in P$ a location in WGS84 coordinate system³ is provided in the form of latitude and

longitude information. Further, the algorithm is configured with a minimum font size f_{min} and a maximum font size f_{max} given in geospatial units. These specify the desired range of font sizes in the generated tag map.

Let T be the resulting set of tags drawn on the map by our algorithm. A given tag $t \in T$ can be described by its center t_{pos} , font size t_f and label category t_l to be displayed. Our algorithm is divided in three main parts. The first part generates a set of seed positions. Each will be used as center position for a tag candidate. The second part calculates the font size and label category of each candidate. At this point a set of tag candidates is fully described but still overlaps between the candidates exist. The last part of the algorithm will select an overlap-free subset of the candidates which is the resulting tag map T .

3.1 Seed Position Generation

At first, we generate a set of seed positions S where tags can be placed in the final step of our layout method. Although arbitrary geographical locations could be used, we compose S of all distinct positions inherent in the data point set P . This way, we avoid placing tags in regions where no data points are located. To improve the run-time performance, we use a pseudo-random sample of the data points. To remain deterministic, this is implemented by incrementally adding the set of input points in a deterministic order to S , maintaining a maximal distance between two seed locations of $\frac{f_{min}}{10}$, ignoring closer locations.

3.2 Tag Candidate Calculation

For each seed position $s \in S$, we initialize a tag candidate $c \in C$ that is a tuple consistent of four attributes:

$$c = \{c_p, c_f, c_l, c_s\}.$$

Whereas the candidate's position c_p is a final value taken from s , the remaining attributes font size c_f , label category the tag will display c_l , and score c_s will be iteratively computed with a bisection method.

3.2.1 Aggregation

For a tag candidate $c \in C$, we define a customizable rectangular aggregation area $R_c(c_f)$ centered at position c_p dependent on the font size c_f delivered by the bisection font scaling procedure explained in section 3.2.2.

The goal of one aggregation step is finding the best fitting label c_l with font size c_f placed at position c_p that reaches the highest score c_s . In order to account for diverse implementations of our layout method, we define two customizable independent functions L and S , both aggregating the information of the data points overlaid by $R_c(c_f)$ to compute c_l and c_s as

$$c_l = L(R_c(c_f)) \quad \text{and} \quad c_s = S(R_c(c_f)),$$

which fully describe the tag candidate's tuple. Section 4 describes implementation variants of the functions R_c , L and S .

3. Widely used by web mapping services such as Google Maps and Bing Maps, we transform the WGS84 coordinates into Web Mercator. We use this planar coordinate system to calculate distances in our layout method and to state units in this paper. Thus, the values given only match geographic distances on the equator.

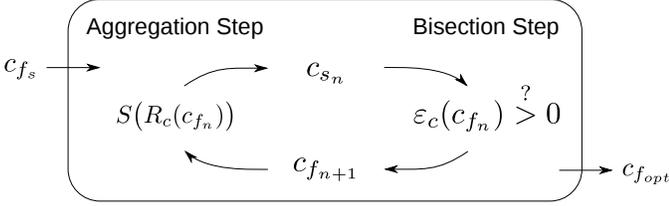


Fig. 2: Illustration of the bisection method for finding the optimal font size $c_{f_{opt}}$ for one tag candidate

3.2.2 Font Scaling

Usually, in tag clouds and tag maps font size encodes the frequency of data points associated to it—a higher font size reflects a larger frequency. We formalize this in an abstract way by targeting a linear relationship between c_f and c_s . So, for each visible tag $t \in T \subseteq C$ we target to hold

$$S(R_t(t_f)) = m \cdot t_f + n$$

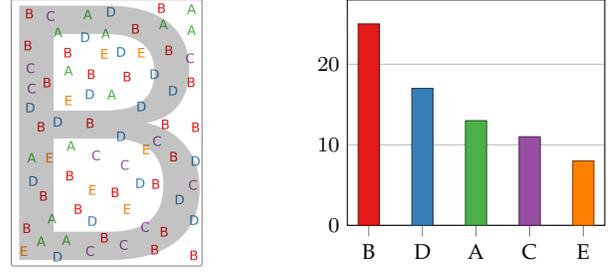
where m and n are the parameters of a line the algorithm will automatically determine. We formalize this as an optimization problem our method tries to solve by minimizing the following scaling energy function:

$$E(T) = \sum_{t \in T} S(R_t(t_f)) - m \cdot t_f + n$$

The scoring function S is allowed to return a floating point number. The font size of a tag is defined as height of its bounding box, as such also a floating point number. Given the set of tag candidates C , we determine their font sizes and labels minimizing E and respecting the configured minimum and maximum font sizes f_{min} and f_{max} . First, for each candidate $c \in C$ we calculate the scores $S(R_c(f_{min}))$ and $S(R_c(f_{max}))$. The minimum of all scores for f_{min} will be assigned to a global minimum $score_{min}$, likewise, $score_{max}$ holds the global maximum. Both values are used to define the linear function $m \cdot t_f + n$ in E to go through the end points $(f_{min}, score_{min})$ and $(f_{max}, score_{max})$ in the respective Euclidean space, thus, setting the parameters m and n . For the corresponding tag candidates which achieve the global $score_{min}$ and $score_{max}$ we set the appropriate minimal and maximal font sizes accordingly. In order to minimize E , we independently search for a font size c_f for each candidate left, such that the error

$$\epsilon_c(c_f) = S(R_c(c_f)) - m \cdot c_f + n$$

is as near as possible to zero. We do this using a bisection method in the interval $[f_{min}, f_{max}]$. Each possible function $\epsilon_c(c_f)$ can be evaluated to zero: For the font size f_{min} all values $S(R_c(f_{min}))$ are higher or equal to $score_{min}$, and $m \cdot f_{min} + n$ will evaluate to $score_{min}$, so that $\forall c \in C: \epsilon_c(f_{min}) \geq 0$. Accordingly, all values $S(R_c(f_{max}))$ are lower or equal to $score_{max}$, so that $\forall c \in C: \epsilon_c(f_{max}) \leq 0$. Figure 2 illustrates the bisection method for one tag candidate c . Starting with an initial font size $c_{f_s} = \frac{f_{min} + f_{max}}{2}$ the scoring function S and the error function ϵ_c are iteratively evaluated to approach an optimal font size $c_{f_{opt}}$. Iterating the bisection method ten times for each candidate c delivered stable results for all examples presented in this paper.



(a) The data points inside R_c and its representative tag B. (b) The histogram of frequencies inside R_c

Fig. 3: An example of the histogram visualizing the aggregation function L for calculating the predominant representative inside the aggregation rectangle R_c

For each candidate c , we set $c_l = L(R_c(c_{f_{opt}}))$ and $c_s = S(R_c(c_{f_{opt}}))$ with the optimal font size determined by the bisection method for the location c_p to generate the set of final tag candidates C .

3.3 Tag Placement

The result of the last step is a set of tag candidates C . Each candidate $c \in C$ consists of a score c_s and a label c_l to be rendered centered at location c_p with font size $c_{f_{opt}}$. Placing all tag candidates on the map would lead to occlusions. In order to determine a subset $T \subseteq C$ of non-overlapping tags to render, we perform a greedy approach. First, we sort C by descending candidate scores c_s . If two candidates share the same score, we favor the one with the lexicographically smaller label, if these are equal, the one with the smaller position values. Then, we iterate the sorted set, incrementally rendering a new candidate as long as its bounding box does not overlap any of the already rendered tags' bounding boxes. The final result T compiles the tag map.

4 PREDOMINANCE METHOD VARIANTS

This section outlines how our tag map layout framework can be customized to implement *predominance tag maps*. First, we list different aggregation strategies, second, we discuss two font scaling variants.

4.1 Aggregation Types

In order to approximate the above introduced perception model following the Gestalt law of proximity, we define aggregation areas the way that shown tags are always derived from their underlying data. A simple solution is using the bounding box of a tag as aggregation area. Figure 3a illustrates a basic example for a single candidate c and its aggregation area R_c for which a representative label needs to be determined. Figure 3b shows the number of data points for each label category inside R_c as a sorted histogram. To account for predominance, we naturally choose the relative majority or the highest valued label in the histogram defined as

$$L(R_c) = \operatorname{argmax}_{l \in L} (|P_l(R_c)|)$$

where $P_l(R_c)$ is the set of data points associated with label category l in the rectangle R_c . If multiple labels reach the same number of data points $|P_l(R_c)|$, we choose the lexicographically smallest label as *argumentum maximi* to ensure a deterministic algorithm. We use this implementation in all examples of this paper though the proposed framework allows for a different definition, e.g., by choosing outliers to be displayed or rarely occurring labels reaching a high total.

In the outlined example, the aggregation areas of all possible labels A to E would be similar. Most often, the word lengths of the label categories vary, which affects the definition of the respective aggregation areas. Below, we will show two strategies to deal with this situation.

4.1.1 approximate bounding box mode (aPTM)

As we do not know the representative label category c_l for a candidate $c \in C$ before defining the aggregation area, we neither know its bounding box nor how to calculate the aggregation area. We implement a fast and simple strategy to overcome this problem by always defining the aggregation area R_c depending on the average aspect ratio r_{avg} of all possible label categories, thereby ignoring actual word lengths. The height of $R_c(c_f)$ is defined by the currently applied font size c_f , so that the width of $R_c(c_f)$ is defined as $r_{avg} \cdot c_f$. As a consequence, wider tags have an under-represented area while narrower tags have an over-represented area. Although this might not be an issue for some applications, it is difficult for a user to sense the aggregation area of a tag on the map without further visual cues.

4.1.2 exact bounding box mode (ePTM)

To avoid the problems of under- and over-representation, this mode considers individual word lengths, thus, it tests all possible labels with their individual bounding boxes as aggregation areas for finding the best fitting representative c_l . Where the heights of all aggregation areas $R_c(c_f)$ are equally defined by c_f , the width $r_l \cdot c_f$ of an individual label to be tested depends on its aspect ratio r_l . Applying this strategy, we need to check if

$$L(R_c) = \operatorname{argmax}_{l \in L} (|P_l(R_c)|) = l$$

for each label category l . In the majority of cases, this test will pass for exactly one category, which will be used to define R_c and L . If the test fails for all categories, we invalidate the tag candidate and ignore it for further processing. If multiple categories pass the test, we choose the category with the highest value for $|P_l(R_c)|$. If there is a draw, we choose the lexicographically smallest label. Applying this strategy, we ensure that each tag to be placed on the final tag map will always represent the predominant label category of the data points enclosed by the tag's bounding box, thereby implementing *predominance tag maps*.

4.2 Font Scaling Types

The font sizes of tags to be placed on the map depend on the scoring function S . In tag clouds and tag map approaches like Thom et al. [1], usually, the font size of a tag correlates with the number of data points it represents. Similarly, in

our case, we target the font size of a tag to correlate with $|P_l(R_c)|$, that is, the number of data points associated with the mostly occurring label category l within the respective aggregation area. We implemented two variants of the scoring function, a linear and a cubic root scaling:

$$S_l(R) = \max_{l \in L} (|P_l(R_c)|) \quad S_c(R) = \max_{l \in L} (\sqrt[3]{|P_l(R_c)|})$$

Although these variants return an integer, thus, not guaranteeing a zero position in ε_c , the bisection method will approach sufficiently enough to zero as shown in section 5.3. The proposed framework can be used to map also other data characteristics to the font size of the tags, such as the overall number of the points inside each aggregation area in order to visualize the density of the data set. This enables font size to be used as a configurable visual variable.

5 EVALUATION

In order to evaluate the quality of our tag map layout method, we applied it to different datasets and compared the results to the tag maps generated by the algorithm of Thom et al. [1] with respect to different characteristics, both quantitatively and qualitatively. In addition, we discuss font scaling and run-time characteristics of our method.

5.1 Computational Complexity

Our algorithm uses two types of acceleration data structures. First, a 2D range tree [49] capable of performing range counting queries in $O(\log^2 n)$. The complexity of such a query is independent of the size of the rectangle and the number (of points) reported. Second, an implementation of a dynamic R-Tree⁴ capable of fast nearest neighbor queries. As preprocessing step, we build one range tree for each label category existing in the dataset, with the data points associated to that category inserted. For the seed position generation, as well as the tag placement steps we use the dynamic R-Tree to find neighbors respectively overlapping tags. For the examples calculated, these steps had minor run-time, not contributing to the overall complexity.

The most time our algorithm spends in the tag candidate calculation step. Given the number of data points n , the number of label categories l , and the number of seed locations s our algorithm is evaluating the aggregation step ten times for each location in the bisection method. In each aggregation step, we evaluate R_c , L and S once. The complexity of these differ for the two modes:

For the aPTM mode evaluating R_c is constant. L and S can be calculated together by evaluating the histogram of categories in a rectangular area. By using one range counting query for each category, this can be done in $O(l \cdot \log^2 n)$ and sums up to $O(s \cdot l \cdot \log^2 n)$ for the whole bisection method.

For the ePTM mode evaluating R_c is the dominant function while the other two can be calculated within the computation of R_c . Here, we need to evaluate the histogram for each label category which sums up to $O(s \cdot l^2 \cdot \log^2 n)$ for the whole bisection method.

The average run-time for each of the 20 instances per dataset and method described in section 5.5 can be seen in

4. <https://github.com/mourner/rbush>

TABLE 1: Average run-time of the quantitative evaluation of section 5.5 in seconds.

	Thom et al.	aPTM	ePTM
<i>atlanta</i>	7.6	13.6	51.9
<i>germany</i>	3.0	69.8	2932.1
<i>us</i>	22.1	79.2	7281.5

table 1. The factor between the aPTM and ePTM modes is roughly the number of categories in the respective dataset, which matches the complexity explained above. Further, Thom et al.’s approach is faster than ours. This slow performance could be addressed within future work, as we have not yet exploited the full potential of possible optimizations. For example the calculations of the bisection method is independent for each tag candidate, as such could be parallelized.

5.2 Datasets

We use three real world datasets to evaluate our method. These datasets vary with respect to the number of categories—7, 53 and 92—which we consider a good basis for a comparative evaluation.

City Name Suffixes Germany (*germany*): Moritz Stefaner⁵ defined a set of 53 typical German city name suffixes. Joining those categories with a geo-referenced list of more than 50,000 German cities⁶, we receive a dataset where each data item represents a city, and the assigned category is one of the 53 city name suffixes. The results reveal a look at prominent city name suffixes in different German regions outlined in figure 4.

Atlanta Crime Data (*atlanta*): Atlanta Crime Data was obtained from the Atlanta police department website⁷. The dataset contains all crime records since 2009, for each incident a geo-reference is provided as well as one of the following crime types: Aggravated Assault, Auto Theft, Burglary, Homocide, Larceny, Rape and Robbery. We chose the crime records of 2009 totalling a number of 39,627 data items for evaluation. Using the crime type as tag illustrates an overview of the most often committed crimes in different neighborhoods (see figure 5).

US Forest Types (*us*): The United States Department of Agriculture published a raster dataset⁸ containing the dominant forest type within a raster of $250m^2$ resolution. We down-sampled this raster to remain 69,696 squares assigned to one forest type. We chose each square’s center as data point associated to the corresponding type. The down-sampled dataset contains 107 unique forest categories. Some of their labels, however, are very long, containing multiple types separated with a slash character. Such wide labels would not be chosen in a real world application, as such we shortened those combined with a slash by only using the first type. This merged some labels and we remained 92 unique categories. As figure 1b shows, this does not distort the overall map picture compared to the colored predominance map rendered using the unmodified raster.

5. <http://truth-and-beauty.net/experiments/ach-ingen-zell>

6. <https://github.com/MoritzStefaner/ach-ingen-zell>

7. <http://www.atlantapd.org/i-want-to/crime-data-downloads>

8. https://data.fs.usda.gov/geodata/rastergateway/forest_type

5.3 Font Scaling Characteristics

As described in section 3.2.2 we minimize the scaling energy function E with a bisection method to result in an approximate linear relationship between the score t_s of a tag $t \in T$ and its font size t_f . An example of the output of both scoring functions, S_l and S_c , is shown in figure 4b and figure 4c. The cubic variant S_c clearly scales lower frequent regions with larger font sizes. In case of the linear scoring function S_l , the font size correlates linearly with the number of data points associated to a tag in its aggregation area. In order to evaluate the quality of the result, we define a set of ordered pairs

$$C_t = \{(t_s, t_f) \mid t \in T\}$$

containing the calculated scores and font sizes of all visible tags T . We test for a linear relationship between those pairs by determining their Pearson correlation r [50], running the algorithm 20 times for each dataset. For each run the ratio between f_{min} and f_{max} increases, keeping the average font size equal. To remain independent of the dataset expansions, we fixed the average at $\frac{b_h}{20}$, where b_h is the height of the bounding box of the whole data set. In all of the performed runs the Pearson correlation r of the respective set C_t is greater than 0.98, except for a few outliers down to 0.90. As such, it can be stated that for both variants a nearly perfect correlation is achieved.

We have observed one behavior which deserves further investigation out of the scope of this paper. For high differences between f_{min} and f_{max} using S_l , the resulting map has often only one tag set to f_{max} (which does not run through the bisection method) while the others are distributed in the lower end of the font size spectrum, changing significantly often for only small variations in f_{min} and f_{max} . We assume, this is caused by the similarity of $S_l(R_c(f))$ and $m \cdot f + n$ occurring under certain circumstances depending on configuration and dataset. This results in more than one reachable zero position for some tag candidates. Interestingly, we have not observed this behavior for the cubic root scaling S_c on our tested datasets.

Figure 5b and 5c show the effect of increasing both font size parameters f_{min} and f_{max} . Our algorithm aggregates information within rectangular areas defined by the bounding boxes of labels, displaying an area’s most suitable representative. Therefore, f_{min} and f_{max} have similar effects on the output like the kernel size parameters have for kernel density estimation techniques (cf. [51], [52]) specifying their aggregation area.

5.4 Comparative Setup

The tag map layout proposed by Thom et al. [1] is the only concurrent approach we found in the literature that tries to place tags as close as possible to their respective locations while keeping font sizes coherent throughout the whole layout. Thom et al.’s tag map layout is designed to visualize a time ordered stream of twitter messages. To apply their algorithm to time independent, categorical point events, we define a data point with its associated category as what they call a term artifact, feed their clustering algorithm with an arbitrary input sequence, reduce the dimensionality to two for all of their calculations and use $2 \cdot f_{min}$ as cluster splitting



Fig. 4: Thom et al. and two scaling variations of ePTM for the *germany* dataset using $f_{min} = 20km$, $f_{max} = 120km$.

threshold k . We apply a linear font scaling dependent on the number of points in each cluster in **step (iii)** of the algorithm (cf. [1], pp. 6). The parameters of the spiral used in their **step (vi)** (cf. [1], pp. 6) as described by Luboschik et al. [53] are $r = f_{min} \cdot 2$, $c = 5$, $m_{max} = 100$ and $d = 1$. Further, we omit any noise cancellation and filtering in the term cloud layout and set the weights according to the number of data points associated to each cluster.

In order to compare our tag map layout approach to Thom et al., we propose a set of metrics. In the following, let P be the set of geospatial data points, T the set of tags placed by a tag map layout algorithm, b_t the bounding box of a placed tag $t \in T$, $A(b_t)$ the area of that bounding box, A_{sum} the sum of all bounding boxes' areas in T , $P(b_t)$ the set of data points inside the bounding box b_t and $P_t(b_t)$ the set of data points associated with the category of the tag t inside b_t .

Nearest Coverage: According to our simplified perception model, an arbitrary location is assigned to its nearest tag on the map. Based on that assumption, we want to measure how many data points nearest to one particular tag are correctly assigned to the tags' category. The more data points match in this regard for one tag, the better it reflects its perceived aggregation area in terms of predominance. Let $N_t(b_t)$ be the set of points associated to the category of the tag t which are closer to the bounding box of t than to the bounding box of any other tag on the tag map T . To retain one metric for the whole tag map, we accumulate the count of these sets and normalize that value by the number of total data points. Thus, we set M_{ncov} to

$$M_{ncov} = \sum_{t \in T} \frac{|N_t(b_t)|}{|P|}.$$

Global Coverage: Next to the above defined measure that accounts for the relationships between tags and underlying data points, the global coverage measure M_{gcov}

estimates how many data points in total are covered by the gained tag distribution to evaluate the density of the tags on the map. Thus, we set M_{gcov} to

$$M_{gcov} = \sum_{t \in T} \frac{|P(b_t)|}{|P|}.$$

Considering the fact that M_{gcov} reaches an ideal result for a single tag covering all data points as well as when applying minimal font size to all tags, we use the same, reasonable values for f_{min} and f_{max} for both algorithms in the comparative evaluation.

Global Categorical Distribution: To evaluate how well a tag map reflects the global distribution of the categories in the dataset, we calculate the normalized histogram of the whole dataset. For each category, it counts the percentage of data points assigned to it. We further compute a normalized histogram of the visible tag categories weighted with their corresponding visible areas. Let T_l be the set of tags showing the label category l . For each category $l \in L$ the value of that normalized histogram entry $h(l)$ is defined by

$$h(l) = \frac{\sum_{t \in T_l} A(b_t)}{A_{sum}}$$

with $h(l) = 0$ for $|T_l| = 0$. These two histograms can be interpreted as two high dimensional vectors of unit length. We define M_{gcat} as the Euclidean distance between the two vectors, so that it reflects the difference between the visible categorical distribution and the categorical distribution of the dataset.

Matching Visual Overlap: To evaluate the similarity between two tag maps, we calculate the percentage of matching overlap between their tags' bounding boxes. Let A be a reference tag map and B a tag map to compare A to. For a tag $a \in A$ we calculate the matching area $m_a(B)$ by iterating over all tags in B , adding the area of the intersection between the pairs' bounding boxes if they show the

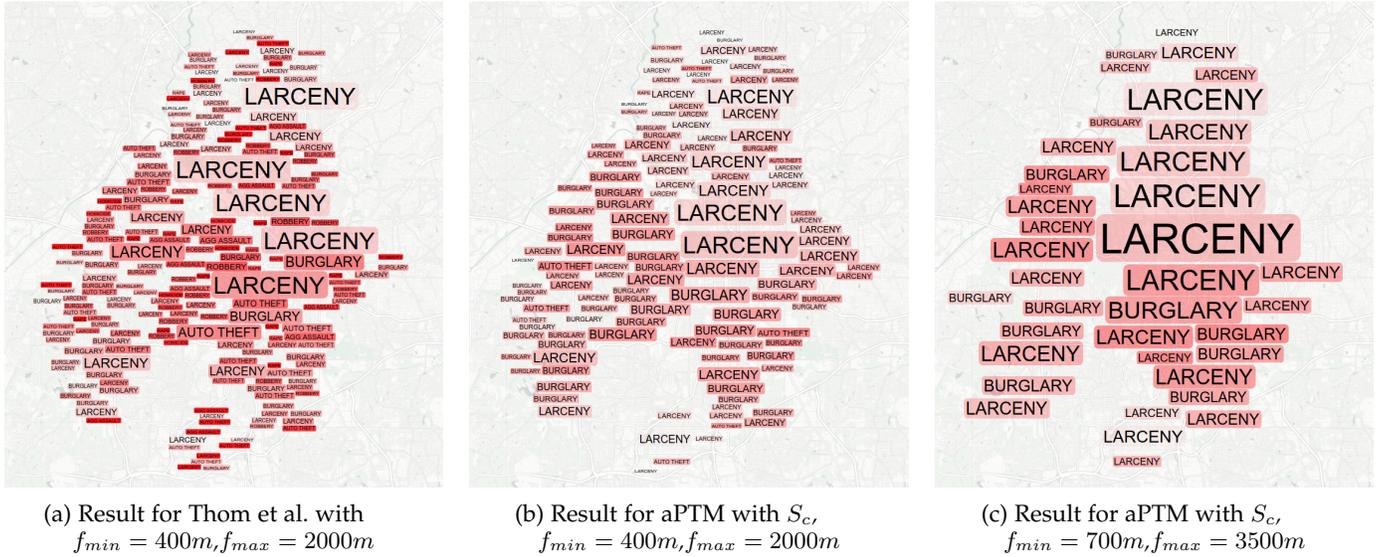


Fig. 5: Results for the *atlanta* dataset. The colored backgrounds show the percentage of data points inside a tags bounding box assigned to that tag to visualize the quality of the output with regards to predominance. A more saturated red indicates a lower percentage.

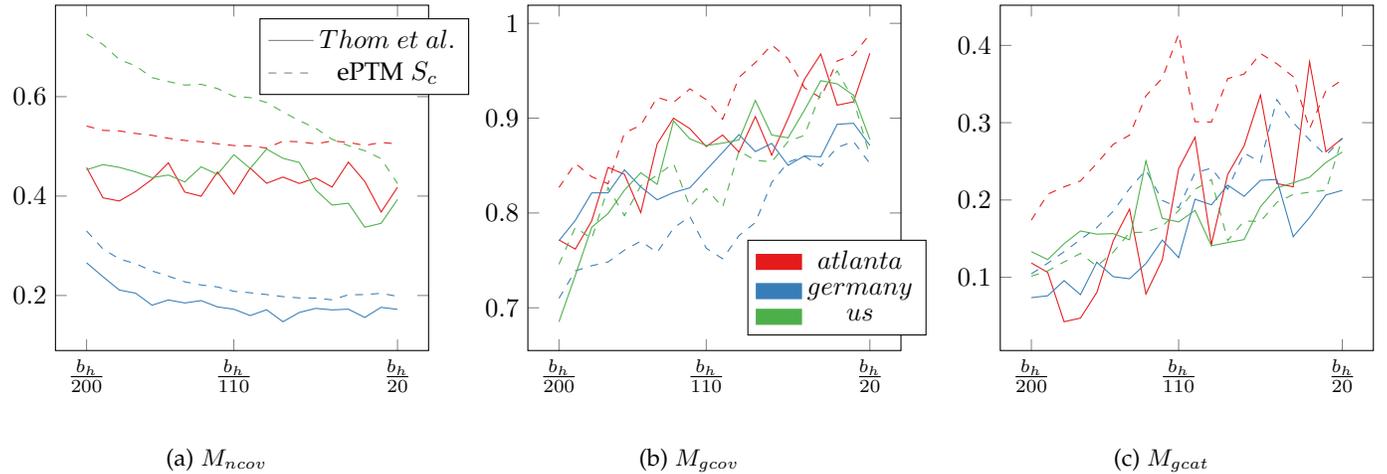


Fig. 6: Quantitative Results for all three datasets encoded by color. A solid line shows Thom et al., a dotted line shows ePTM using S_c . The x-axis denotes f_{min} linearly increasing from left to right, $f_{max} = 5 \cdot f_{min}$.

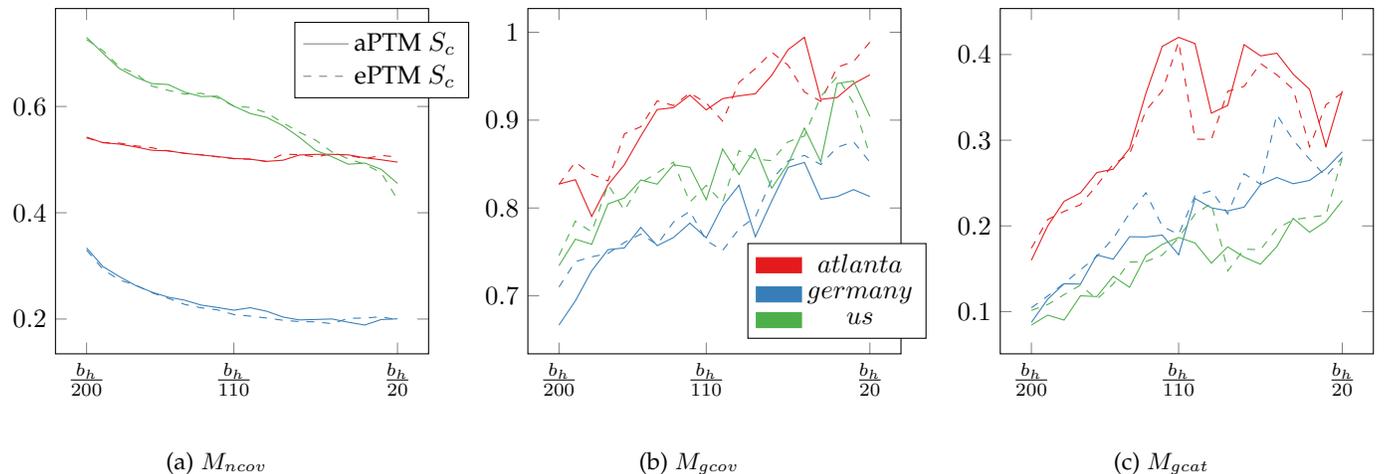


Fig. 7: Quantitative Results for all three datasets encoded by color. A solid line shows Thom et al., a dotted line shows ePTM using S_c . The x-axis denotes f_{min} linearly increasing from left to right, $f_{max} = 5 \cdot f_{min}$.

same label. We calculate the overall percentage of similarity by accumulating the matching areas and normalize with the sum of areas in A :

$$O_A(B) = \frac{\sum_{a \in A} m_a(B)}{A_{sum}}$$

5.5 Comparative Analysis

With these metrics defined we can quantitatively and qualitatively compare the results of our layout algorithm with the algorithm described by Thom et al. [1].

Quantitative Setup: As in the font scaling evaluation, we use the height of the bounding box of each dataset b_h as reference to set the range of font sizes for evaluating the quantitative metrics. We started at $\frac{b_h}{200}$ for f_{min} , linearly increasing until $\frac{b_h}{20}$ and generated 20 results per dataset and method, keeping $f_{max} = 5 \cdot f_{min}$. The results of this empirical analysis are shown in figure 6 and figure 7.

Local Predominance vs. Global Distribution: The idea of our approach is to generate a layout optimizing the local predominance, showing regional hot spots of predominant label categories for arbitrary granularities controlled by the font size parameters. As shown in figure 6a, for all three datasets our method clearly achieves higher scores for M_{ncov} , a metric used to indicate local predominance at the assumption of our perception model. We color coded the tags in figure 5 to visualize this difference. By contrast, our method is neither suited, nor intended to show the global distribution of the categories appropriately. The more the categorical distributions in the rectangular aggregation regions are similar or at least deliver the same winner, which is dataset-dependent, the less the global distribution is shown. This effect can be divined in figure 5 where the result produced by Thom et al. shows a larger number of distinct tags, potentially reflecting the global categorical distribution better within the given dataset. A quantitative analysis showing how well the global categorical distribution is reflected can be seen in figure 6c. With increasing font sizes (decreasing the amount of shown tags), the Euclidean distance between the histogram of the visible tag map and the histogram of the underlying categorical distribution increases as well for both methods. This is expected, as smaller font sizes result in less aggregation and a larger amount of tags, better reflecting the global distribution. The difference of the Euclidean distances between the methods also depend on the dataset, as Thom et al. is performing better for two datasets, while both methods perform similar for the *us* dataset given the same font size bounds. A further examination of this behavior is however outside the scope of this paper.

Global Coverage: For the results of the global coverage metric M_{gcov} shown in figure 6b no clear winner can be identified. For the *atlanta* data set our method performs a little better, in contrast to the *germany* dataset and close to the *us* dataset. However, we think that the spiral placement used by Thom et al. is of general advantage here, as it can better fill gaps in the map area, with the drawback of overplotting the boundaries of the dataset as can be observed in figure 4a. This effect will vary with different parameters used for the archimedean spiral which we fixed in the examples of this paper.

TABLE 2: Mean visual overlap for similar input sets derived from random samples. The first column denotes the size of the samples. The other denote the mean visual overlap of the resulting tag maps of the respective algorithm and dataset.

sample size	<i>atlanta</i>		<i>germany</i>	
	Thom et al.	ePTM	Thom et al.	ePTM
100%	42%	100%	29%	100%
95%	42%	91%	32%	84%
90%	46%	89%	31%	76%
80%	43%	84%	31%	70%

Method Variant Analysis: Figure 7 shows the quantitative comparison between the two aggregation variants aPTM and ePTM. Interestingly, the values of M_{ncov} are nearly identical and the values of M_{gcov} and M_{gcat} are very similar. This shows that both modes operate equally good with respect to our perceptual assumption. However, only the ePTM mode ensures that shown tags coincide with the predominant category their bounding boxes cover.

Stability Analysis: The method described by Thom et al. is designed to work for a constant stream of twitter messages and it is nondeterministic as it incrementally builds geospatial clusters using the k-means algorithm. This produces different outputs, even when run on the same input. To investigate the similarity of the output given similar inputs, we evaluate both methods ten times with the same configuration, each run using a fixed sized random sample from a dataset. We calculated the mean of the matching visual overlap $O_A(B)$ for each possible different ordered pair of tag maps (A, B) from the ten runs of a method. Table 2 shows the result of this analysis for the *atlanta* and *germany* datasets using random samples ranging from 100% to 80% of the respective number of data points. The similarity of our approach scales with the similarity of the random sample while further depending on the structure of the dataset, showing determinism when the input is equal. The similarity in the output of Thom et al. however does not seem to relate to the similarity of the random sample. Instead, their approach consists of constantly higher variance, even when fed with the same input.

6 CONCLUSION

Given a set of labels associated to geographical locations, choosing a subset of them to be placed on a map has been an important, longstanding task in cartography. The amount of text to render exceeds the available space, so that strategies avoiding occlusions are needed to keep the map legible.

Tag maps transfer this task to the field of geovisualization. Geospatial data points are associated with textual information, and a tag distribution is visualized as a thematic map overlay. Since vast geo-referenced datasets often contain different labels for one and the same location, tag map layout methods need to decide on aggregations and omissions. We observed that related approaches did not deliver satisfactory results in terms of local predominance that is, a tag on the map always represents the relative majority of data points in a defined local aggregation area.

Predominance maps used in cartography usually implement visualization techniques using color to encode the

predominant category. We evaluated how this encoding could be implemented using tags to improve the number of distinguishable categories in a *predominance tag map*. While the approach by Thom et al. [1], which uses a displacement strategy when tags occlude, distorts the layout by loosening the linkage between themes and geography, our presented layout framework is designed to strengthen that relationship. It is configurable to implement different aggregation strategies, even beyond the implemented *predominance tag maps* while preserving a desired font size mapping. We defined a set of measures to investigate qualitative aspects of tag maps empirically, and we could show that our approach serves better results regarding local predominance than the method by Thom et al. under our assumed perception model. Following our proposed strategy, the resultant label distribution is locally optimized, which can affect the global summary of the given dataset. But due to the nature of real world datasets used in this paper, our approach also retains and visualizes their inherent regional thematic diversities.

It has to be said that the reliability of all tag map layout methods—including ours—suffer from aggregation and omission decisions, and the problem of the interdependency of parameter settings, especially the chosen font sizes, will always deliver local optima if a tag map layout shall be computable in a reasonable amount of time. In as much as it represents locally predominant tags reliably, our method performs well. The scope of this paper was to investigate this fact numerically with the proposed measures under the assumption of a perception model. Future work will include run-time optimization as well as investigating if our method also performs best when it is comparatively evaluated by users. This includes qualitative aspects in terms of aesthetics, which requires future efforts, for example, in applying a color scheme that visually links labels of the same tag.

ACKNOWLEDGMENTS

We thank Tom Liebmann for fruitful discussions and *us* dataset preprocessing, and David J. Wrisley for proof reading.

REFERENCES

- [1] D. Thom, H. Bosch, S. Koch, M. Worner, and T. Ertl, "Spatiotemporal Anomaly Detection Through Visual Analysis of Geolocated Twitter Messages," in *Proceedings of the 2012 IEEE Pacific Visualization Symposium*, ser. PACIFICVIS '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 41–48.
- [2] T. A. Slocum, R. B. McMaster, F. C. Kessler, and H. H. Howard, *Thematic Cartography and Geovisualization*, 3rd ed., ser. Prentice Hall Series in Geographic Information Science. Prentice Hall, 2009.
- [3] ArcGIS, "What is a Predominance map?" 2018, <https://arcgis.com/help/arcgis/10.4/arcgis/10.4-01-03>. (Retrieved 2018-01-03).
- [4] B. Ruefenacht, M. Finco, M. Nelson, R. Czaplowski, E. Helmer, J. Blackard, G. Holden, A. Lister, D. Salajanu, D. Weyermann, and K. Winterberger, "Conterminous U.S. and Alaska Forest Type Mapping Using Forest Inventory and Analysis Data," *Photogrammetric Engineering & Remote Sensing*, vol. 74, no. 11, pp. 1379–1388, 2008.
- [5] A. Jaffe, M. Naaman, T. Tassa, and M. Davis, "Generating Summaries and Visualization for Large Collections of Geo-referenced Photographs," in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, ser. MIR '06. New York, NY, USA: ACM, 2006, pp. 89–98.
- [6] E. Imhof, "Positioning Names on Maps," *The American Cartographer*, vol. 2, no. 2, pp. 128–144, 1975.
- [7] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.
- [8] H. J. Miller and J. Han, *Geographic Data Mining and Knowledge Discovery*. Bristol, PA, USA: Taylor & Francis, Inc., 2001.
- [9] L. Wilkinson, *The Grammar of Graphics (Statistics and Computing)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
- [10] D. B. Carr, R. J. Littlefield, and W. L. Nicholson, "Scatterplot matrix techniques for large n," in *Proceedings of the Seventeenth Symposium on the interface of computer sciences and statistics on Computer science and statistics*. New York, NY, USA: Elsevier North-Holland, Inc., 1986, pp. 297–306.
- [11] N. Andrienko and G. Andrienko, *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, 2005.
- [12] M. Novotny, "Visually effective information visualization of large data," in *In 8th Central European Seminar on Computer Graphics (CESCG 2004)*. CRC Press, 2004, pp. 41–48.
- [13] Y. Zhang, W. Luo, E. A. Mack, and R. Maciejewski, "Visualizing the Impact of Geographical Variations on Multivariate Clustering," *Computer Graphics Forum*, 2016.
- [14] R. Beecham, J. Dykes, W. Meulemans, A. Slingsby, C. Turkay, and J. Wood, "Map lineups: Effects of spatial structure on graphical inference," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 391–400, Jan 2017.
- [15] M. Dörk, S. Carpendale, C. Collins, and C. Williamson, "Vis-Gets: Coordinated Visualizations for Web-based Information Exploration and Discovery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1205–1212, Nov. 2008.
- [16] S. Jänicke, C. Heine, and G. Scheuermann, *GeoTemCo: Comparative Visualization of Geospatial-Temporal Data with Clutter Removal Based on Dynamic Delaunay Triangulations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 160–175.
- [17] R. Scheepens, H. v. d. Wetering, and J. J. v. Wijk, "Non-overlapping Aggregated Multivariate Glyphs for Moving Objects," in *2014 IEEE Pacific Visualization Symposium*, March 2014, pp. 17–24.
- [18] H. Aman, P. Irani, and F. Amini, "Revisiting Crisis Maps with Geotemporal Tag Visualization," in *Proceedings of the 2014 IEEE Pacific Visualization Symposium*, ser. PACIFICVIS '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 177–184.
- [19] J. Sinclair and M. Cardew-Hall, "The folksonomy tag cloud: when is it useful?" *Journal of Information Science*, vol. 34, no. 1, pp. 15–29, 2008.
- [20] S. Milgram and D. Jodelet, "Psychological Maps of Paris," *Environmental Psychology*, pp. 104–124, 1976.
- [21] F. Viegas and M. Wattenberg, "TIMELINES: Tag Clouds and the Case for Vernacular Visualization," *interactions*, vol. 15, no. 4, pp. 49–52, Jul. 2008.
- [22] M. Hearst and D. Rosner, "Tag Clouds: Data Analysis Tool or Social Signaller?" in *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, Jan 2008, pp. 160–160.
- [23] S. Murugesan, "Understanding Web 2.0," *IT Professional*, vol. 9, no. 4, pp. 34–41, July 2007.
- [24] F. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, "ManyEyes: a Site for Visualization at Internet Scale," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1121–1128, Nov 2007.
- [25] F. Viegas, M. Wattenberg, and J. Feinberg, "Participatory Visualization with Wordle," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 15, no. 6, pp. 1137–1144, Nov 2009.
- [26] R. Vuillemot, T. Clement, C. Plaisant, and A. Kumar, "What's being said near 'Martha'? Exploring name entities in literary text collections," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, Oct 2009, pp. 107–114.
- [27] S. Koch, M. John, M. Worner, A. Muller, and T. Ertl, "Varifocal-Reader – In-Depth Visual Analysis of Large Text Documents," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 12, pp. 1723–1732, Dec 2014.
- [28] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong, "TextFlow: Towards Better Understanding of Evolving Topics in Text," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2412–2421, Dec 2011.
- [29] W. Cui, S. Liu, Z. Wu, and H. Wei, "How Hierarchical Topics Evolve in Large Text Corpora," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 12, pp. 2281–2290, Dec 2014.
- [30] C. Collins, F. Viegas, and M. Wattenberg, "Parallel Tag Clouds to explore and analyze faceted text corpora," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, Oct 2009, pp. 91–98.

- [31] B. Lee, N. Riche, A. Karlson, and S. Carpendale, "SparkClouds: Visualizing Trends in Tag Clouds," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, no. 6, pp. 1182–1189, Nov 2010.
- [32] U. Hinrichs, S. Forlini, and B. Moynihan, "Speculative Practices: Utilizing InfoVis to Explore Untapped Literary Collections," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 429–438, Jan 2016.
- [33] S. Lohmann, J. Ziegler, and L. Tetzlaff, "Comparison of Tag Cloud Layouts: Task-Related Performance and Visual Exploration," in *Human-Computer Interaction - INTERACT 2009*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5726, pp. 392–404.
- [34] J. Schrammel, M. Leitner, and M. Tscheligi, "Semantically Structured Tag Clouds: An Empirical Evaluation of Clustered Presentation Approaches," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. ACM, 2009, pp. 2037–2040.
- [35] S. Jänicke, J. Blumenstein, M. Rücker, D. Zeckzer, and G. Scheuermann, "TagPies: Comparative Visualization of Textual Data," in *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP*. SciTePress, 2018, pp. 40–51.
- [36] W. Cui, Y. Wu, S. Liu, F. Wei, M. Zhou, and H. Qu, "Context preserving dynamic word cloud visualization," in *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, March 2010, pp. 121–128.
- [37] Y. Wu, T. Provan, F. Wei, S. Liu, and K.-L. Ma, "Semantic-Preserving Word Clouds by Seam Carving," in *Computer Graphics Forum*, vol. 30, no. 3. Wiley Online Library, 2011, pp. 741–750.
- [38] S. Afzal, R. Maciejewski, Y. Jang, N. Elmqvist, and D. S. Ebert, "Spatial Text Visualization Using Automatic Typographic Maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2556–2564, Dec 2012.
- [39] H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, and T. Ertl, "ScatterBlogs2: Real-Time Monitoring of Microblog Messages Through User-Guided Filtering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2022–2031, Dec. 2013.
- [40] H. Strobelt, M. Spicker, A. Stoffel, D. Keim, and O. Deussen, "Rolled-out wordles: A heuristic method for overlap removal of 2d data representatives," *Computer Graphics Forum*, vol. 31, no. 3pt3, pp. 1135–1144, 2012.
- [41] S. Hahmann and D. Burghardt, *Maple – a Web Map Service for Verbal Visualisation using Tag Clouds Generated from Map Feature Frequencies*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–12.
- [42] A. Godwin, Y. Wang, and J. T. Stasko, "TypoTweet Maps: Characterizing Urban Areas through Typographic Social Media Visualization," in *EuroVis 2017 - Short Papers*, B. Kozlikova, T. Schreck, and T. Wischgoll, Eds. The Eurographics Association, 2017.
- [43] D.-Q. Nguyen and H. Schumann, "Taggram: Exploring Geo-data on Maps Through a Tag Cloud-Based Visualization," in *Proceedings of the 2010 14th International Conference Information Visualisation*, ser. IV '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 322–328.
- [44] K. Buchin, D. Creemers, A. Lazzarotto, B. Speckmann, and J. Wulms, "Geo word clouds," in *Proceedings of the 2016 IEEE Pacific Visualization Symposium*, ser. PACIFICVIS '16, April 2016, pp. 144–151.
- [45] S. Openshaw, "The Modifiable Areal Unit Problem." Geo Abstracts University of East Anglia, 1984.
- [46] S. Ahern, M. Naaman, R. Nair, and J. H.-I. Yang, "World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-referenced Collections," in *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, ser. JCDL '07. New York, NY, USA: ACM, 2007, pp. 1–10.
- [47] A. Slingsby, J. Dykes, J. Wood, and K. Clarke, "Interactive Tag Maps and Tag Clouds for the Multiscale Exploration of Large Spatio-temporal Datasets," in *Proceedings of the 11th International Conference Information Visualization*, ser. IV '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 497–504.
- [48] C. Ware, *Information Visualization: Perception for Design*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2012.
- [49] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008.
- [50] B. S. Everitt and S. A., *The Cambridge dictionary of statistics*, 4th ed. Cambridge University Press, 2010.
- [51] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, Sep. 1956.
- [52] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.
- [53] M. Luboschik, H. Schumann, and H. Cords, "Particle-based labeling: Fast point-feature labeling without obscuring other visual features," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1237–1244, Nov. 2008.

Martin Reckziegel received his master degree (diploma) in computer science in 2013. Currently, he is a PhD candidate at the Image and Signal Processing group of Leipzig University. His doctoral studies are co-supervised by Prof. Gerik Scheuermann and Dr. Stefan Jänicke. His research interest includes unsolved problems in geovisualization, driven by research questions in application fields such as digital humanities and environmental sciences.

Muhammed Faisal Cheema received his PhD in Computer Science from Leipzig University, Germany in 2017. He collaborated with humanities scholars to design and develop visualization strategies to explore concept change in ancient text corpora. His doctoral work focused on creating a research environment for humanities scholars to perform visual analyses on ancient texts with a user-driven concept search. His research interests include visualization of textual and hierarchical data.

Gerik Scheuermann received the master degree (diploma) in mathematics in 1995 and a PhD degree in computer science in 1999, both from the Technical University of Kaiserslautern. He is a full professor at the University of Leipzig since 2004. He has coauthored more than 200 reviewed book chapters, journal articles, and conference papers. His current research interests focus on visualization and visual analytics, especially on feature and topology-based methods, flow visualization, medical visualization, document visualization and visualization for life sciences. He has served as paper cochair for Eurovis 2008, IEEE SciVis 2011, IEEE SciVis 2012, and IEEE PacificVis 2015. He has organized TopoInVis 2007, AGACSE 2008 and EuroVis 2013.

Stefan Jänicke is a post-doctoral researcher at the Image and Signal Processing Group at Leipzig University, Germany, where he leads a text visualization group focusing on applications in the digital humanities. Over the last years, he has gained experience in developing information visualization and visual analytics techniques within a number of projects. His PhD thesis investigates the utility of visualization techniques to support the comparative analysis of digital humanities data, and his current research relates to information visualization with a focus on text- and geovisualization.