# Small Multiples with Gaps

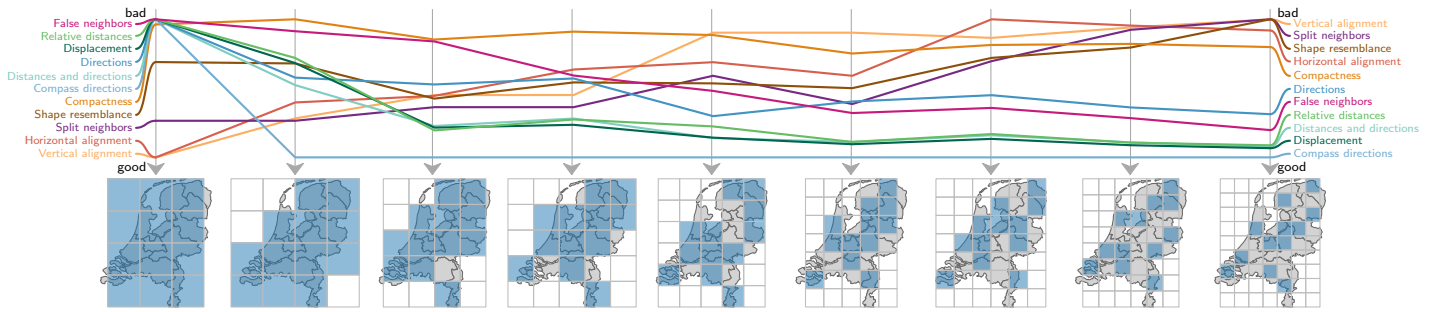Wouter Meulemans, Jason Dykes, Aidan Slingsby, Cagatay Turkay, and Jo Wood *Member, IEEE*

Fig. 1: The effects of adding gaps (whitespace) to a small-multiples layout (blue squares) representing the 12 provinces of the Netherlands, measured via our suite of metrics. Each metric is represented by a colored line in the chart, indicating at each axis how well the layout below performs in the metric. The layout algorithm here optimizes for the *displacement* metric—which aims to preserve the spatial (geographic) distribution—as whitespace is increased from left to right. Although some metrics show improvement as gaps are added, others reflect the resulting smaller and more dispersed distribution, which may hinder comparison.

**Abstract**—Small multiples enable comparison by providing different views of a single data set in a dense and aligned manner. A common frame defines each view, which varies based upon values of a conditioning variable. An increasingly popular use of this technique is to project two-dimensional locations into a gridded space (e.g. grid maps), using the underlying distribution both as the conditioning variable and to determine the grid layout. Using whitespace in this layout has the potential to carry information, especially in a geographic context. Yet, the effects of doing so on the spatial properties of the original units are not understood. We explore the design space offered by such *small multiples with gaps*. We do so by constructing a comprehensive suite of metrics that capture properties of the layout used to arrange the small multiples for comparison (e.g. compactness and alignment) and the preservation of the original data (e.g. distance, topology and shape). We study these metrics in geographic data sets with varying properties and numbers of gaps. We use simulated annealing to optimize for each metric and measure the effects on the others. To explore these effects systematically, we take a new approach, developing a system to visualize this design space using a set of interactive matrices. We find that adding small amounts of whitespace to small multiple arrays improves some of the characteristics of 2D layouts, such as shape, distance and direction. This comes at the cost of other metrics, such as the retention of topology. Effects vary according to the input maps, with degree of variation in size of input regions found to be a factor. Optima exist for particular metrics in many cases, but at different amounts of whitespace for different maps. We suggest multiple metrics be used in optimized layouts, finding topology to be a primary factor in existing manually-crafted solutions, followed by a trade-off between shape and displacement. But the rich range of possible optimized layouts leads us to challenge single-solution thinking; we suggest to consider alternative optimized layouts for small multiples with gaps. Key to our work is the systematic, quantified and visual approach to exploring design spaces when facing a trade-off between many competing criteria—an approach likely to be of value to the analysis of other design spaces.

**Index Terms**—Geographic visualization, small multiples, whitespace, design space, metrics, optimization

---◆---

## 1 INTRODUCTION

One of the ideas most closely associated with Edward Tufte [33] is the use of "*small multiples to present data in a dense fashion that supports comparison and enquiry*". Whilst such an ordered "*series of graphics, showing the same combination of variables, indexed by changes in another variable*" predates this description [6], the exposition is persuasive. Tufte introduced small multiples by example with a series of 23 maps originally produced by McRae et al. [23] to present the results of an air pollution model [33]. Each individual map—or *small multiple*—shows modeled hydrocarbon emissions with the same spatial frame and visual encoding, but for a different hour of the day. Originally presented sequentially as part of an animated video, Tufte jux-taposes the maps in a regular array of small multiples, whose spatial arrangement conveys the temporal relationships between each, with comparison facilitated by order and alignment [9, 19].

Since the conditioning variable in the example above is ordinal, a 1D arrangement is appropriate. Long ordinal sequences are frequently split into rows for a more compact array that facilitates comparison and allows for larger small-multiples graphics. Where the conditioning variable is two-dimensional, a 2D arrangement may be appropriate. This is often the case for geography [16, 38, 40], but also for 2D time such as hour & day [35], two independent parameters controlling a model [28] and other "abstract" projections such as scatter plots (Fig. 3) or multidimensional scaling (MDS) [21].

**Whitespace.** We investigate how whitespace (gaps) can help convey the distribution of space informing the small-multiples ordering. For example, small-multiple maps by month, where some months are missing, may benefit from gaps to reflect such omissions. Such gaps, however, reduce the size of each small multiple, perhaps making them less readable. In 2D, geographically arranged layouts may benefit from gaps to better convey the spatial distribution of the conditioning variable (Fig. 2). This may give a more accurate geographic depiction, but limits the size of each small multiple. We emphasize that this can apply to any 2D projection, as illustrated in Fig. 3 for a scatterplot-

- *All authors are at the giCentre, City University London. E-mail: { wouter.meulemans | j.dykes | a.slingsby | cagatay.turkay.1 | j.d.wood } @city.ac.uk.*
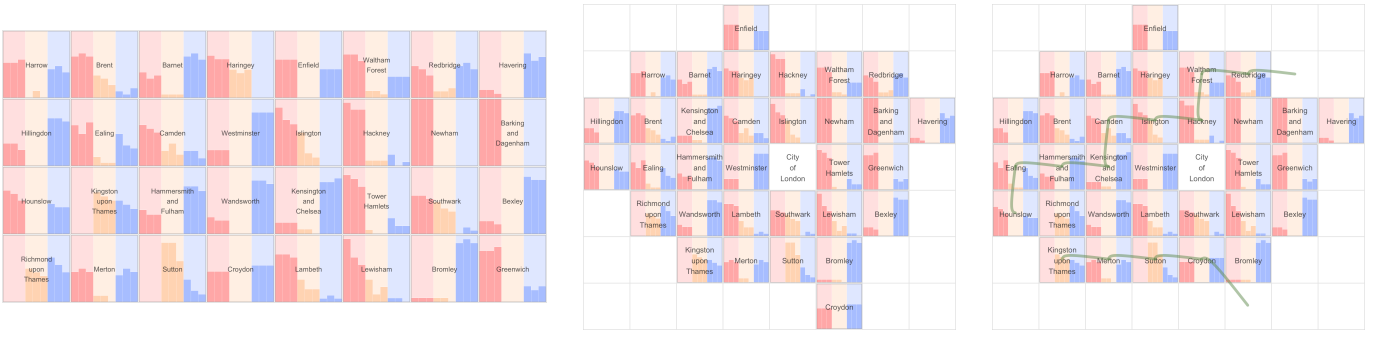
Fig. 2: A BallotMap [38] in the original $8 \times 4$ layout without gaps (left) and rearranged with our optimization method as a small multiples with gaps (42.9% whitespace) to more clearly show the underlying spatial distribution (boroughs of London) in terms of shape (middle) and displacement (right). The green curves in the right map show how small multiples (boroughs) moved with respect to the middle map. Note how a gap is used to visually show the absence of data for City of London.



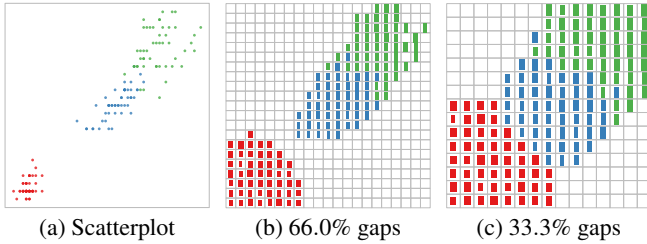| (a) Scatterplot | (b) 66.0% gaps | (c) 33.3% gaps |

Fig. 3: (a) Scatterplot of petal width and height for 150 samples of 3 Iris species [34]. (b) Each sample is placed in a small-multiples array, using a glyph whose width and height are sepal width and height respectively. (c) Using fewer gaps reduces precision of the petal distribution, but increases the comparability of the sepal glyph.

based small-multiples graphic. Such simultaneous representation of underlying distribution and other data in the small multiples allows for a juxtaposed comparison; whitespace enables a trade-off in emphasis between the components. Moreover, adding gaps allows for placing exterior graphic annotation with small multiples in the resulting whitespace.

In this paper we investigate the technique of *small multiples with gaps*, formalizing the idea that has been implicitly informing existing designs but lacking in scientific attention. As the existing designs with gaps use spatial conditioning and ordering, we focus on such geographically-informed layouts, to study the effects of the number and position of gaps on the characteristics of the resulting small-multiple arrays. Note that we do so from the perspective of a designer's intent, rather than a user's interpretation thereof. We offer three related contributions that inform and support researchers and practitioners using and studying small multiples with gaps:

1. a comprehensive suite of metrics to capture quality aspects of a small-multiples layout, implemented in an optimization method to compute layouts with good performance in these metrics;
2. an exploratory analysis of the impact of and interaction between the metrics and the effects of adding whitespace in a range of structurally varied cases;
3. a method for structurally conducting such analysis through visualization of a design space defined by metrics.

In this paper the third contribution is instantiated in an interactive tool for exploring trade-offs between the metrics used in computing and assessing small multiples with gaps. We consider it to be an approach that has the potential of being applied to comparable problems facing a set of potentially conflicting design criteria.

**Methodology.** We introduce a set of metrics (Sect. 2) to quantify characteristics of a small-multiples layout. They are used as optimization criteria in an algorithm to compute such layouts (Sect. 3). Through a series of experiments supported by interactive visualization software

(Sect. 4), we systematically "optimize" for one metric and introduce varying degrees of whitespace to generate a multitude of layout alternatives, measuring the quality of the layouts according to all metrics. We analyze the data for relations and conflicts between the metrics (Sect. 5) and study them for existing layouts (Sect. 6).

## 2 MEASURING SMALL MULTIPLES

To assess the quality of the layout used in a small-multiples array, we propose a rich set of metrics. The importance of a metric, of course, depends on what is being shown in each multiple and the purpose of the graphic. We categorize them according to the characteristic of the layout that they aim to capture. We distinguish two types of metrics: *array characteristics* that aim to capture how well we can see and compare the multiples and *projection characteristics* that capture how well the layout reflects the original continuous distribution[1]. Design of small multiples with gaps involves *trade-offs* between these two objectives: arrays that allow us to discern and compare graphics effectively and those that more closely reflect the original positions and/or relationships between each multiple. Consider e.g. the balances sought in Fig. 2 and Fig. 3 between map or distribution fidelity and frame size. Our metrics can equally apply to non-rectangular small multiples (e.g. hexagon-based), but here we focus on square multiples in regular arrays based on geographic distributions. Our metrics are based on averages, to avoid the number of small multiples dominating the effects when comparing cases of varying sizes.

**Notation.** We denote by $M$ the set of small multiples, effectively data elements that are to be assigned to cells in the arrays to define a layout. The set $T \subseteq M^2$ describes neighbors—in maps, regions that share a boundary—and thus the topology of $M$. We denote by $A$ the array into which we place the multiples in $M$. The set of rows is denoted by $rows(A)$, each row is a subset of $A$; analogously, the set of columns is given by $columns(A)$. The layout $\mathscr{L}$—an assignment between multiples and grid cells in the output array—is given by a bijection between $M$ and a subset of $A$. We use $\mathscr{L}$ as a function mapping multiples to their assigned grid cells and grid cells to their assigned multiples (using NULL as a special value for empty grid cells). We use $\mathscr{F}_{\mathscr{L}}(X) = \{c \mid c \in X \wedge \mathscr{L}(c) \neq \text{NULL}\}$ to denote the cells in some set $X \subseteq A$ with an assigned multiple. We use $|X|$ to denote the size of a set $X$: e.g. $|M|$ and $|A|$ denote the number of multiples and number of cells in the array respectively. Moreover, we use the notation $x, x' \in X$ to denote a pair of distinct elements in $X$, i.e., it is a shorthand for $\{x, x'\} \in \{\{a, b\} \mid a \neq b \wedge (a, b) \in X^2\}$.

We use a metric $d$ to measure distance. We may use a metric between their centroids or between their boundaries. We choose the former, since the latter does not allow us to distinguish compactness between a rook's adjacency and a bishop's adjacency[2]. As a metric, we

---

[1] The geographic map of provinces and boroughs in Fig. 1 and Fig. 2 respectively; the 2D distribution of each sample's petal width and length in Fig. 3.

[2] This refers to chess-piece adjacency: rook's adjacency are the four horizontal and vertical neighbors; bishop's involves the four diagonal neighbors.

use the standard Euclidean distance. Hence, in all cases, we instantiate $d$ with the Euclidean distance between centroids.

## 2.1 Array characteristics

We identify three array characteristics, aimed to capture important factors related to effective reading and comparison within a layout.

**Whitespace.** As we add gaps to small multiples, we need to increase the size of the array containing the layout. Accordingly, we can either: *(i) enlarge*—increasing the space in which the array is depicted to retain resolution, resulting in a *bigger graphic*—, or *(ii) shrink*—decrease the resolution of the individual small-multiple graphic whilst using the same space, resulting in *smaller multiples* (as is the case in Fig. 1 where the individual cells reduce in size as whitespace is introduced from left to right). Tufte identifies numbers per square inch and entries in the data matrix per area of data graphic as means of quantifying data density [33]. These metrics reduce in the *enlarge* case and thus would be regarded as having resulted in a less dense data graphic according to Tufte's metric. They do not vary when adding gaps to small multiples in the *shrink* case above and would result in an equally data dense graphic, but may cause legibility problems. Both cases move us away from the optimum in density and legibility. We use *proportion of gaps* as a key metric for whitespace.

$$\text{WHITESPACE} = 1 - \frac{|M|}{|A|}$$

Note that Wongsuphasawat [37] considers the size of the array, which indirectly measures our WHITESPACE metric, as only layouts for the United States are considered.

**Compactness.** Smaller distances between small multiples is likely to make it easier to compare their contents. Thus, compactness is often a desirable characteristic. Evidently, our core interest is in relaxing this condition and establishing the effects of adding gaps. To measure compactness, we consider all pairwise distances between cells of the layout that contain a multiple (and not a gap).

$$\text{COMPACTNESS} = \underset{m,m' \in M}{\text{average}} \left\{ d\left( \mathcal{L}(m), \mathcal{L}(m') \right) \right\}$$

where $d$ is the Euclidean distance between centroids. Hence, the best layout for COMPACTNESS is an approximate circle.

**Alignment.** Estimation tasks are performed more successfully when stimuli are aligned [9, 19] and as such aligning small multiples can help in their comparison. The degree to which alignment occurs is captured by our alignment metric. We distinguish between horizontal (multiples in the same row) and vertical alignment (multiples in the same column) separately, as the importance of each depends on the visualization used with a multiple. For example, a vertical-bar chart benefits from horizontal alignment, whereas a horizontal-bar chart benefits from vertical alignment. To quantify alignment, we measure the number of pairs of multiples that lie within the same row or column.

$$\text{HORIZONTAL ALIGNMENT} = \underset{r \in rows(A)}{\text{average}} \left\{ \frac{1}{2} |\mathscr{F}_{\mathscr{L}}(r)| \cdot (|\mathscr{F}_{\mathscr{L}}(r)| - 1) \right\}$$

$$\text{VERTICAL ALIGNMENT} = \underset{c \in columns(A)}{\text{average}} \left\{ \frac{1}{2} |\mathscr{F}_{\mathscr{L}}(c)| \cdot (|\mathscr{F}_{\mathscr{L}}(c)| - 1) \right\}$$

## 2.2 Projection characteristics

One of Tufte's key objectives in achieving graphical integrity is that "*the representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numeric quantities represented*" [33]. For small multiples, this implies a need for relations in the spatial distribution to be maintained in the layout. However, Tufte's edict is unachievable as we force irregular continuous space to a discrete array to ensure that small multiples are comparable and do not overlap. This makes the problem at hand very similar to that of map projections [31]: obtaining a good 2D representation of a spherical world. This inspires us to consider measurements for

distance (equidistant projections), directions (azimuthal projections), topology (interrupted projections) and shape (conformal projections).

Various efforts to reflect underlying spatial relations in regular grids have been reported under a range of conditions, e.g. [12, 13, 21, 39]. Most relevant to our efforts is the work of Eppstein et al. [13]. We identify the extent to which the following characteristics of the underlying space are retained as indicative of the degree to which the small-multiples layout reflects this space.

**Displacement.** We want to ensure little movement of multiples when transforming the underlying space into the layout. A simple but effective method is to measure the displacement of each small multiple.

$$\text{DISPLACEMENT} = \underset{m \in M}{\text{average}} \left\{ d\left( m, \mathcal{L}(m') \right)^2 \right\}$$

This metric (and variants of it) are used in the algorithmic work of Eppstein et al. [13]. As defined here, it is generally not invariant under any operation, but an optimal translation can be computed to minimize this measure [10, 13]. We assume that the grid and the map have been aligned already, permitting no translation or scaling to optimize this measure further. Although minor translations could have a big effect on the measurement, the following three arguments support our choice: *(i)* we have no optimization algorithm that accurately deals with the same restrictions for our other measures, potentially making for unfair comparisons; *(ii)* it allows us to assess how well this basic measure performs while avoiding the costly computational overhead of optimizing the alignment; *(iii)* it is also not immediately obvious how different a computed layout may be when translations are included, when a reasonable alignment is already provided.

**Distance.** The DISPLACEMENT metric does not directly correspond to Tufte's principle mentioned above: we measure relative distances only indirectly. To more accurately capture that relative distances are preserved, we consider the change in distance between pairs.

$$\text{DISTANCE (X)} = \underset{(m,m') \in X}{\text{average}} \left\{ \left( d\left( m, m' \right) - d\left( \mathcal{L}(m), \mathcal{L}(m') \right) \right)^2 \right\}$$

where X is the set of pairs to compare. For this we consider $\text{ALL} = M^2$ (all pairs) and $\text{NBR} = T$ (neighbors in the input map). As for COMPACTNESS, we have multiple choices for defining distance but focus on the Euclidean distance between the centroids.

**Direction.** As with distances, directions in the layout should be equal to directions in the underlying space. We consider an analogous metric (using ALL or NBR for X) to establish differences in the directions between pairs of regions in their geographic context and in the layout.

$$\text{DIRECTION (X)} = \underset{(m,m') \in X}{\text{average}} \left\{ dev\left( m, m', \mathcal{L}(m), \mathcal{L}(m') \right) \right\}$$

where $dev \colon M^2 \times A^2 \to [0, \pi]$ measures absolute angular deviation of the directions between the centroids. This metric is invariant under translation and scaling. We measure precise angles, whereas Wongsuphasawat [37] only counts those exceeding a threshold.

Similar to Eppstein et al. [13], we also define a measure that considers the compass directions (orthogonal order violations: inversions along $x$ or $y$ axis between input and output) between two regions.

$$\text{COMPASS DIRECTION (X)} = \underset{(m,m') \in X}{\text{average}} \left\{ ortho\left( m, m', \mathcal{L}(m), \mathcal{L}(m') \right) \right\}$$

where $ortho \colon M^2 \times A^2 \to [0, 2]$ measures the number of orthogonal-order violations: it returns 0 if both north-south and east-west relations are preserved; 1 if one of these is violated; and 2 if both are violated. Note that Eppstein et al. [13] count only the pairs of regions with a violation, whereas our metric counts the number of violations.

**Vector–distance and direction.** The invariances of DISTANCE can be considered undesirable: flipping a layout upside down has no effect on distances. To mitigate such effects, we combine the effects of distance and direction into a single measure by considering vectors instead.

$$\text{VECTOR (X)} = \underset{(m,m') \in X}{\text{average}} \left\{ d\left( m' - m, \mathcal{L}(m') - \mathcal{L}(m) \right)^2 \right\}$$

**Topology.** Ideally, the topology (adjacencies) of the underlying map is perfectly reflected in the layout. To define measures of topology, we should first consider what it means to be adjacent in the grid. To this end, we use chess-adjacency again. We define sets $R$ and $B$ to contain pairs of *assigned* cells with both rook's and bishop's case.

A primary aspect is measuring which elements should be adjacent but are not adjacent in the layout: which neighbors have been split?

$$\text{SPLIT NEIGHBORS} = \underset{(m,m')\in T}{\text{average}} \left\{ split(m,m') \right\}$$

where $split\colon M^2 \to [0,1]$ assigns as follows: $split(m,m')$ is 0 if $(\mathscr{L}(m),\mathscr{L}(m')) \in R$, 0.3 if this pair is in $B$ and 1 otherwise. The converse—false neighbors—also play a role in preserving topology: are adjacent multiples in the layout adjacent in reality?

$$\text{FALSE NEIGHBORS} = \underset{(c,c')\in R\cup B}{\text{average}} \left\{ false(c,c') \right\}$$

where $false\colon A^2 \to [0,1]$ assigns a cost as follows: $false(c,c')$ is 0 if $(\mathscr{L}(c),\mathscr{L}(c')) \in T$, 0.3 if $(c,c') \in B$ and 1 otherwise. In both cases, we weigh bishop's adjacency at 30 percent, as they are visually less salient than the rook's case. Both of these violations are also measured by Wongsuphasawat [37]. Finally, we also define a complete topology measure, using a weighted average of the above two aspects, weighing a split between neighbors more heavily than a false adjacency.

$$\text{TOPOLOGY} = 2\cdot\text{SPLIT NEIGHBORS} + \text{FALSE NEIGHBORS}$$

**Shape.** As space-filling small multiples are the norm, small multiples with gaps may be initially unfamiliar to many users. As such, it is desirable for the overall shape of the layout to correspond to that of the underlying map, to lower the acceptance threshold of the layout. Crucial here is that we look at the overall appearance, a first impression, without necessarily considering which regions are assigned to which cells. Whereas Wongsuphasawat [37] captures this as a subjective Boolean test, we make this quantifiable. Many geometric measures (e.g. [2, 3, 22]) require a matching between geometric objects; though we have such implicitly, this defeats the purpose of quantifying a first impression through the overall composition. The popular Hausdorff distance [18] is a "bottleneck" measure, only quantifying similarity through a locally worst situation. The symmetric difference (sum of areas covered by exactly one of the shapes) is unable to accurately capture the similarity if the small-multiples layout covers significantly more or significantly less area than the underlying map.

Intuitively, we want the layout to overlap the underlying map as much as possible, while gaps overlap the map as little as possible; with "too much whitespace", we want the cells to spread out over the shape, to capture roughly its spatial extent, whereas with "too little white-space" we want cells that cannot overlap the underlying map to stay close to the map (and not pick an arbitrary cell). We formalize this as follows. First, we define a weight function $w\colon A \to [0,1]$, where $w(c)$ is the fraction of $c$ overlapped by regions in $M$. We also define a spread function $s\colon A \to \mathbb{R}$ as the minimal sum of weighted distances for distributing exactly one weight: $s(c) = \min_{\omega\in\Omega}\{\sum_{c'\in A} \phi(c')d(c,c')\}$, where $\Omega$ describes all ways of distributing one weight. More precisely, $\omega\colon A \to [0,1]$ is a function with $\sum_{c'\in A} \phi(c') = 1$ and $\phi(c') \leq w(c')$ for all $c' \in A$. As $w$ and $s$ do not depend on the layout, they can be precomputed. Moreover, $s(c)$ can be computed greedily by taking the nearest neighbors of $c$. Our measure of shape is now expressed as:

$$\text{SHAPE} = \frac{1}{|M|}\left( \sum_{c\in A} \min_{c'\in\mathscr{F}_{\mathscr{L}}(A)}\{w(c)\cdot d(c,c')\} + \sum_{c\in\mathscr{F}_{\mathscr{L}}(A)} s(c) \right)$$

The rationale of the first sum is that for a largely overlapped cell, we want to have a nearby assigned cell. The second term expresses that every assigned cell should have a total weight of 1 in its vicinity. The lower the value of SHAPE, the better the shape is preserved. Our metric can be seen as a bidirectional take on the earth-mover's distance [10] with the symmetric difference, though we allow any amount of "earth" to be piled up onto the target cell. It does not take topology into account nor the number of underlying data items (regions). As such, it is particularly suitable to measure a "first impression".

## 3 CREATING SMALL MULTIPLES WITH GAPS

**Existing examples.** Various small multiples with gaps have been created by hand and are being produced by design agencies, media outlets, researchers and data enthusiasts to represent a wide range of data sets. These include the *London Squared Map* produced by *After the Flood* and the *Future Cities Catapult* and in which 33 boroughs are represented with equally sized squares arranged spatially to help "*compare data across all of London*" [1]; Radburn's use of this layout for spatial small multiples [27]; Die Zeit's *regional distribution* maps in which the 16 German states are represented in a similar manner [7]; and various efforts to map the US including, Guo et al.'s "Map$^2$ Matrix" with small-multiple maps of the US in a 2D ordinal arrangement to reflect their geography [16] and Park's small multiple grid maps used in the New York Times to show the expansion of states authorizing gay marriage [25], Powell et al. of the Guardian US Interactive team [26] and Fong's implementation in Tableau [15]. Semi-automatic approaches also produce attractive outputs such as Krist Wongsuphasawat's grid map of Thailand [36].

**Algorithmic approaches.** Spatially ordered treemaps [12,39] produce ordinal 2D layouts. As with all treemap algorithms, they completely partition rectangular space into tessellating rectangles that may have different sizes and shapes. Specifying a fixed area for rectangles and introducing well-placed "dummy nodes" [29, 30, 41, Fig. 4] can add sufficient gaps, though getting the positions right involves some trial-and-error.

Grid maps are ordinal 2D layouts that can be considered as candidates for small multiples with gaps. Eppstein et al. [13] turn this into a point-matching problem, where a set of locations in continuous space (e.g. geographic space) needs to be matched to a regular array of cell locations. The approach involves a linear program that achieves the matching by minimizing DISPLACEMENT between the original and matched cell locations—one of the criteria that we aim to minimize. Eppstein et al. [13] also minimize changes in orthogonal directions (similar to COMPASS DIRECTIONS (ALL)) and other measures of displacement, concluding that minimizing total squared-distance displacement produces the best results. However, they also note that results are very dependent on the position and extent of the candidate locations with respect to those of the original locations. Our analysis partially extends these conclusions in the presence of gaps, on systematically varied maps and with different metrics.

**Optimizing the metrics.** We are investigating a multitude of metrics for optimizing small multiples with gaps. Foregoing in-depth algorithmic investigation for the time being, we use a heuristic optimization approach that lets us optimize layouts according to any of our metrics and even combinations of these. To this end, we implemented a simulated-annealing process [20]; below is a brief description of its crucial aspects. Full details and the implementation itself are in the online supplementary material.

The process starts with some layout and tries to make small adjustments to improve its optimization function. To escape local optima, the process sometimes accepts adjustments that lead to a worse solution, but the chances of doing so decreases in later iterations. Based on observations by Eppstein et al. [13], the initial layout is computed using their linear program optimizing DISPLACEMENT. If this is our optimization criterion, no further steps occur.

As the annealing process provides no guarantee of optimality, we perform an additional hillclimbing step afterwards, allowing any pair to be swapped. We do this until either no swaps are made or $t = 60$ seconds have passed. This time bound was sufficient for the process to end unconstrained in all considered cases.

When optimizing for metrics that are invariant under a transformation (e.g. translation for VECTOR (ALL)), we test whether any such transformation can be applied to the layout to improve DISPLACEMENT. This allows us to make reasonable assessments of the performance of other measures that are not invariant under the applied operations. Array characteristics and SHAPE are invariant under actual region assignment. For these, we optimize DISPLACEMENT using Eppstein et al.'s point matching approach [13].
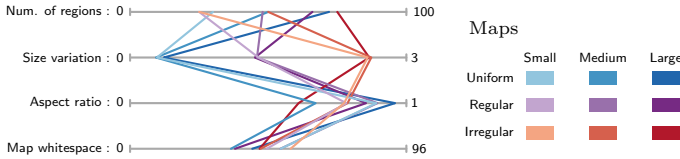
Fig. 4: A Parallel-Coordinates Plot of the map characteristics.

## 4 EXPERIMENTS

This process of generating small multiples with gaps allows us to produce myriad alternative layouts of varying size, optimizing according to various metrics and computing our metrics for each solution. To do so comprehensively and consistently, we structure our experiments to involve spatial data with a range of characteristics and develop visualization tools that enables us to explore effects in the design space.

**Map generation.** To ensure a range of map characteristics, we use maps generated by aggregating output areas from random locations across the UK [4]. We control for and measure the characteristics detailed below (see Fig. 4 and online material for details). Each map was fitted to a $500 \times 500$ box to ensure comparable distance measurements. Map colors are assigned using ColorBrewer sequential schemes with different hues [17] to reflect sequential and thematic differences.

We controlled for the number of multiples (regions in the map) to be positioned in the small-multiples array. Small maps have 25 to 30 regions, Medium 48 to 51 and Large 66 to 75.

Map regions are assigned to equally sized cells in the array through our processes. It is likely that the distortion is thus effected by size variations in the input. We measure this via the coefficient of variance: the standard deviation of region area divided by the mean area. A low value means that the input regions have roughly the same size; a high value implies large size differences. We controlled for this measure: Uniform maps have a coefficient of variation between 0.28 and 0.37, Regular from 1.37 to 1.4 and Irregular approximately 2.6.

The map's aspect ratio may affect the results, particularly when using square target arrays. To compute it, we use the width $w$ and height $h$ of the smallest axis-aligned bounding box containing the map. We take the minimum of $w/h$ and $h/w$ to obtain comparable numbers for vertically and horizontally elongated maps. Using the same bounding box, we also measure the *map whitespace*: the amount of whitespace inherent in the map. We define this as $100(1 - A/B)$, where $A$ is the total area of all map regions and $B$ is the bounding box area.

**Running trials.** For each of the above maps, we generate grids (target arrays) with various numbers of rows $R$ and columns $C$, to vary the whitespace in the eventual solution. In particular, we run two sequences: one in which the grid is required to be square ($R = C$) and one with flexibility, in which the aspect ratio of the grid is at least 80 percent of the aspect ratio $A$ of the map: $0.8A \leq C/R \leq 1$ if $A \leq 1$ and $1 \leq C/R \leq A/0.8$ otherwise. For each sequence we compute for 0% up to and including 80% whitespace at 5% intervals, the smallest grid that satisfies the constraint of the sequence and has at least the specified whitespace; we use the actual percentage in our analysis.

For each combination of map and array, we run the simulated-annealing algorithm to optimize each measure individually. The only exception being that we never optimize for FALSE NEIGHBORS: early experiments showed that optimizing only for this measure leads to inappropriate layouts that disperse the multiples in the layout.

The simulated annealing approach is not guaranteed to give us a global optimum, only a local optimum after the hillclimbing step. Therefore, we run 10 trials (including hillclimbing and postprocessing) for each condition (map $\times$ grid $\times$ optimization criterion). In our analysis, we consider only the best result of these trials.

**Visualizing the design space.** These trials produce a mass of data that quantifies design-space characteristics for small multiples with gaps. Appropriately, we use interactive visualization to reveal structure, trends and outliers as we explore these and provide software to support this activity. Designed iteratively as needs were established through our exploratory analysis, it provides interactive views that allow us to: compare characteristics of the input maps (Fig. 4); view and

configure representations of the layout with visual encodings of positional and topological error; and explore relationships between metrics as whitespace varies, both between metrics and existing layouts. A small-multiples matrix is the basis of three views designed to show the effects of whitespace. This allows us to make comparisons between the effects on the metrics for layouts optimized for each metric.

The *Metrics Matrix* is key here (see Fig. 5 with the details below), with columns representing the optimized criteria and rows indicating measured metrics. Each cell of the matrix is designed to show patterns as whitespace varies: it displays a line chart of the measured metric (row) as whitespace increases from left (0%) to right (100%), when we optimize for the metric indicated by the column. These values have been normalized according to the worst-case value occurring through the entire row; data values are comparable within one row, though usually not across rows due to the different metrics. Comparison across rows focuses trends rather than comparing actual values.

The *Trade-Off Matrix* uses a similar layout, but is designed to compare differences in the values of the metrics (row) under a particular optimization: it is designed to reveal how much we lose in one metric, if we optimize for another, visually representing the trade-off. Each cell contains a bidirectional bar chart; each bar shows the difference in the measured metric (row) between two solutions for the same map-array combination. One of the layouts, the "target", is always optimized for the metric of the column. The other layout, the "comparator", can be configured to be the layout optimized according to either: (i) the metric of the row (in online material only); or (ii) a specified fixed metric (Fig. 6). The bar's length indicates the magnitude of difference between the target and comparator. Its direction shows which performs better: a downwards bar indicates that the comparator performs better, upwards means the target (column-metric optimized) performs better. In setting (i) each small multiple involves two metrics and allows us to assess how much better we could have done for the row-metric, had we chosen to optimize for it instead of the column-metric. In other words, how much did we lose by optimizing for something else? All bars should be directed downward in this view, unless the annealing process did not compute an optimal solution. In setting (ii) each graphic involves three metrics, allowing us to assess a trade-off between the column-metric and the metric specified for the comparator, to answer questions such as "*how much do we gain or lose in terms of* FALSE NEIGHBORS *(row) if we choose to optimize for* SHAPE *(column) instead of* DISPLACEMENT *(comparator)*?" Bars are grouped by map, ordered by increasing whitespace in the target array (left to right); this is the same for both setting (i) and setting (ii).

The *Examples Matrix* operates nearly identically to the Trade-Off Matrix with setting (ii). The only difference is that the comparator is not set to another optimization, but to an existing layout instead. In addition to the standard graphics, this view also allows for an additional column, showing the absolute values for each metric (row) as a bar chart. The existing layout takes on the role of "target", whereas the metric of the column provides the "comparator"; a downward bar means that our optimization outperforms the existing layout. Through this representation, we may attempt to reverse-engineer the criteria that played an important role in the existing layout's construction.

The matrices support exploratory investigation with rich and rapid interaction, through: reorderable rows and columns, details on demand showing configurable sequences of layouts as whitespace varies, arrow keys to aid within graphic selection of points, lines and bars, and pattern matching to highlight sequences in which criteria change according to particular characteristics as whitespace increases. The metrics and maps used in a matrix can be selected or omitted instantly. Cells in which the same metric is optimized and measured (the "diagonal" of the matrix) are marked using a gray background. Our tools and data, which readers are invited to explore, are available online[3].

## 5 METRIC COMPARISON THROUGH OPTIMIZATION

We use our software to systematically compare the metrics, optimizing for one and measuring the other in the result, for increasing amounts

---

[3]Online material: http://www.gicentre.net/smwg

of whitespace under various conditions (map × array × optimization metric). In particular, we focus our exploratory analysis on the following four questions, considering the impact of whitespace:

Q1 How does capturing metrics (DISTANCE, VECTOR, DIRECTION and COMPASS DIRECTION) between neighboring regions compare to between all regions? Transitivity may imply that optimizing (NBR) would automatically yield a good score on (ALL).

Q2 What is the relation between SHAPE and DISPLACEMENT? E.g., does keeping regions near their locations preserve shape?

Q3 How do the three topological measures compare? Is there value from considering TOPOLOGY as a combination of FALSE NEIGHBORS and SPLIT NEIGHBORS?

Q4 What relations exist between the best metrics for optimization, if any, as uncovered with the previous questions?

Below, we present the main findings revealed using our interactive matrices. We focus on the use of square arrays and briefly touch upon non-square arrays at the end. We refer to the online supplementary material for an extensive description and additional, extended figures.

**Q1: All or neighbors.** Here, we compare DISTANCE, VECTOR, DIRECTION and COMPASS DIRECTION, each having two variants: (ALL) in which all pairs of input regions are considered and (NBR) in which relationships between neighboring regions are the sole focus. Fig. 5 shows that these metrics—when optimized, gray cells—monotonically decrease (improve) as we increase whitespace; this is to be expected as we simply allow for more flexibility. Two exceptions are DIRECTION (NBR) and COMPASS DIRECTION (NBR), although optimal results should not get worse when enlarging the grid. This suggests an inability of our simulated-annealing process to adequately compute an optimal solution. This calls for more specialized algorithms: can, for example, the algorithm by Eppstein et al. [13] for compass directions be modified to approximate DIRECTION (NBR)?

The off-diagonal (white) cells allow us to compare the (ALL) and (NBR) variants of the same metric. It is clear that optimizing for (ALL) also implies good performance in (NBR). Moreover, we observe two patterns: for DISTANCE and VECTOR, optimizing for (NBR) yields reasonable performance for (ALL), at least for low amounts of whitespace; for DIRECTION and COMPASS DIRECTION this does not seem to be the case as (NBR) yields poor performance for (ALL).

Performing similar comparisons between different characteristics (see online material) reveals that DISTANCE performs poorly in the other metrics, in spite of the postprocessing step to account for reflection and rotation. This may be caused by more structural flaws than global transformations (e.g. rotation may not suffice to resolve these issues), though we perform only rotations of 90 degrees: the perfect alignment may not be possible. Similarly, the DIRECTION and COMPASS DIRECTION metrics perform poorly on the other two metrics, as the former do not account for distances. We observe, however, that there is reasonable performance if there is very little whitespace.

Not surprisingly, the VECTOR metric seems to perform best overall as it combines both distance and angle. Particularly, it performs well for DISTANCE, but also has reasonable scores for the two directional metrics. Fig. 6 allows us to compare the performance of VECTOR (ALL) more easily and confirms that, barring some loss on preservation of (compass) directions, VECTOR (ALL) performs well (only short upward bars on the diagonal) compared to the other metrics (long downward bars in any column).

It is worth reflecting on the trade-off for considering measurements on neighbors (computational performance) and on all pairs of regions (metric performance). The choice need not be a dichotomy: rather than taking only direct neighbors, we can also include the neighbor's neighbors to compute relations, etc. This reduces the number of pairs drastically and improves computational performance compared to all pairs, and may improve the metric performance compared to considering only. Our visualization methodology would allow us to further explore such effects for a range of cases with geographic variations.

**Q2: Shape and displacement.** We postprocessed solutions optimized for SHAPE to assign regions to cells to minimize DISPLACEMENT—but with cells constrained to those occupied in the computed layout.
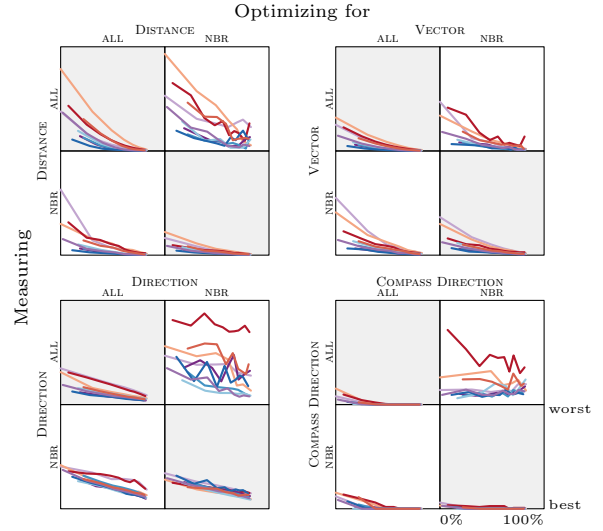


Fig. 5: Four *Metrics Matrices* to compare (ALL) and (NBR) variants of four different metrics. A cell shows how the layouts of a map (line) perform in a metric (row) when it is optimized for another (column), as whitespace increases (left to right). Lower values mean better performance. In a gray cell, the same metric is measured and optimized.
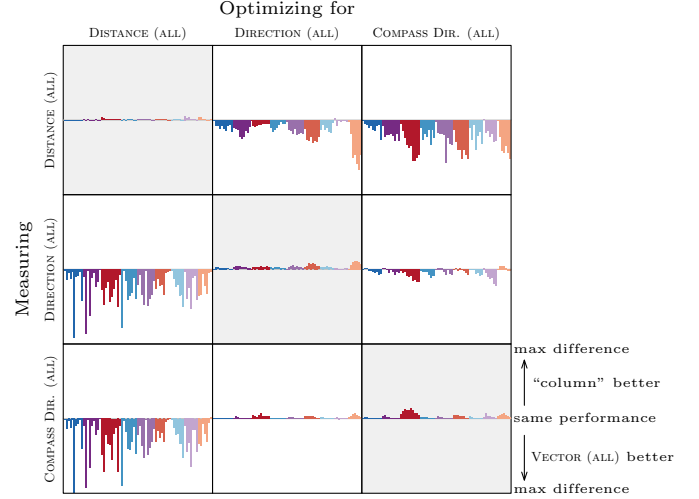


Fig. 6: *Trade-Off Matrix* to compare the performance per metric (row) between VECTOR (ALL)-optimized layouts and those optimized for other metrics (column). Bars are map-array combinations, ordered by increasing whitespace from left to right per map (color).

The assigned cells are also drawn towards the location of the regions in the map: we may expect DISPLACEMENT and SHAPE to perform similarly. Fig. 7 shows that the differences between the two optimized maps are very similar on regular maps, but the differences increase with irregularity; Fig. 8 shows the different effect the two metrics have on the layout for an irregular case. Interestingly, the difference seems low in comparison, for the Large Irregular input. Inspecting this further reveals that this input contains several clusters of small regions spread out through a number of large regions, whereas the others have one central cluster of smaller regions (see online material for the maps). Hence, the coefficient of variation of region sizes is not sufficient to capture map complexity for this task. Looking at the other measures, we see that displacement tends to outperform shape, in particular for measures involving direction (e.g. VECTOR and DIRECTION).

We conclude the following: if a map consists of regular regions or if the small and large regions are relatively well dispersed, optimizing for displacement is likely to provide a good shape, while also performing well on the other measures. On the other hand, if a map has a high coefficient and the small regions are strongly clustered in one area,
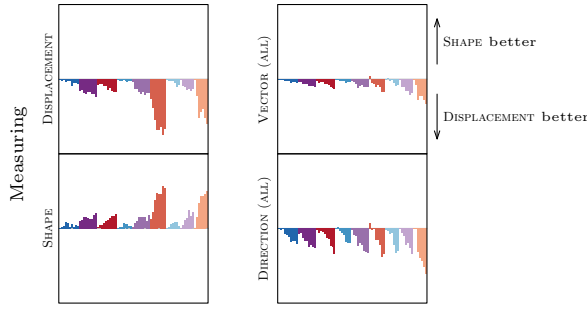
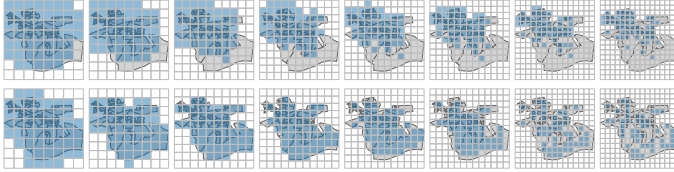Fig. 7: *Trade-Off Matrix* to compare DISPLACEMENT and SHAPE.



Fig. 8: Lineup of layouts for Medium Irregular, optimizing for DIS-PLACEMENT (top row) and for SHAPE (bottom row).
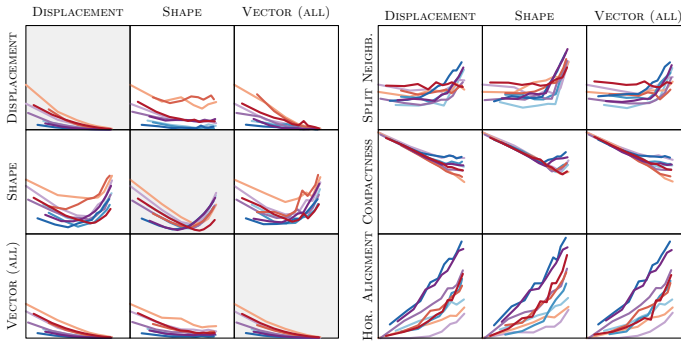


Fig. 9: *Metrics Matrices* to analyze the behavior of the best performing metrics with respect to each other (left) and other metrics (right).
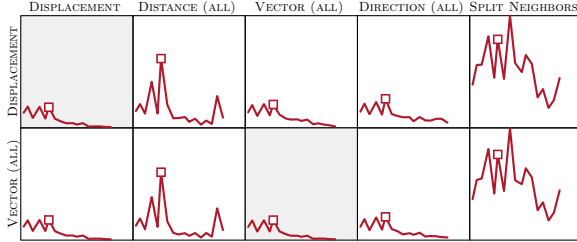


Fig. 10: *Metrics Matrix* displays common peaks (one highlighted) with flexible array dimensions.

some degree of shape should be explicitly considered at higher levels of whitespace. While this may negatively impact some other measures, it increases the first-glance resemblance and may thus lower the threshold of initial acceptance of the layout. For further analysis, we maintain comparisons with both SHAPE and DISPLACEMENT.

**Q3: The effect of false neighbors.** Early exploration quickly revealed that FALSE NEIGHBORS as an optimization criterion, resulted in inappropriate layouts, essentially dispersing the regions to avoid any adjacencies. Comparing the performance of the two other topological metrics, we find that the differences are minor (see online material for details). TOPOLOGY outperforms SPLIT NEIGHBORS in 17 out of 75 cases, when measuring SPLIT NEIGHBORS; conversely, SPLIT NEIGHBORS outperforms TOPOLOGY when measuring TOPOLOGY in 49 cases. As with the (NBR) variants above, this suggests an inability of the simulated-annealing process to properly optimize either metric. This calls for better algorithms to optimize topological characteristics, before making judgments between these.

Further comparison (online material) reveals that the topological measures do not perform well in nontopological metrics. We conclude that the topological metrics are likely unsuitable as the sole optimization criterion, at least for our simulated-annealing approach. Even when provably optimizing for topological measures, the discrepancy caused in other metrics is likely to be undesirable. If we combine these metrics with others (e.g. DISPLACEMENT), we may obtain outputs with desirable properties that ensure a good degree of topology.

**Q4: Bringing it all together.** With the results above, we now continue with comparisons between VECTOR (ALL), DISPLACEMENT and SHAPE. Let us consider the line charts in Fig. 9. Comparing DISPLACEMENT and VECTOR (ALL) shows that increasing whitespace improves the measured metric (row), irrespective of geometry and these patterns are consistent between these two optimizations.

The SPLIT NEIGHBORS row shows a deterioration pattern for all three optimized metrics: there is little to no improvement when adding the first gaps, but at at some point the result deteriorates, drastically decreasing performance in topology.

If we look at SHAPE, we see an interesting pattern: first, it starts off poorly as with only little whitespace we cannot accurately capture the shape of the input map. As whitespace increases, this improves as gaps are placed on the boundary of the array. At some point, the shape reaches its optimum (roughly at the percentage of the map whitespace; see online material). After this point, cells become too small to cover the original map, thus again losing on shape performance. This inflection point seems to roughly correspond to the percentage at which SPLIT NEIGHBORS starts to worsen rapidly, the internal gaps in the layout explaining this behavior of SPLIT NEIGHBORS.

Finally, we see that COMPACTNESS improves linearly with whitespace, until roughly this same percentage of whitespace, where some differentiation occurs between the various input maps. Those with more size variation (the red lines) then achieve better compactness than those with less. This is likely caused by the clusters of small regions that cause the assigned cells to remain closer together in the computed layouts: this effect is geometry dependent. However, more whitespace implies smaller multiples: we leave an investigation of the relationship between smaller distances and larger arrays to future work. Not surprisingly, alignment decreases as whitespace increases, the retention of geography preventing alignment in the layout.

Eppstein et al. [13] found that optimizing for DISPLACEMENT implies good performance in terms orthogonal directions and maintaining correct adjacencies (though both are measured slightly differently). Here, we wish to dive deeper into this result, seeing whether this finding generalizes. Fig. 11 illustrates the difference in performance between DISPLACEMENT and optimization for the other metrics. This mostly verifies their conclusion, at least at low values of whitespace. However, as discussed before, increasing whitespace tends to reduce the effectiveness of optimizing DISPLACEMENT when measuring SHAPE or SPLIT NEIGHBORS.

**Flexible arrays.** Turning briefly to the data collected for non-square arrays, we observe mostly the same patterns. However, there is quite a strong signal caused in nearly every metric as the aspect ratio of the array is no longer fixed (Fig. 10). Even the straightforward, provably optimized DISPLACEMENT shows such a pattern where adding some whitespace may have a *negative effect* on this metric. In part, this can be explained by the different alignment and scaling of the array with the underlying map. However, adding a row and column to a square array also causes a slightly different alignment of the centroids, but no such effect was observed in the previous analysis. Moreover, the peaks of this more erratic behavior seem to correspond across metrics. This suggests that some aspect ratios fit more naturally to a given map; choosing the right one thus is important. There is likely an association between the map's aspect ratio and the location of clusters with small region sizes. Though our methodology would be useful in exploring this, it calls for other aspects of the input to be controlled and is beyond the scope of this analysis.

**Summary of findings.** Our findings suggest that using more whitespace than inherent in the map is not advisable, unless done intention-

ally to provide extra space for labels or auxiliary information. Such whitespace is unlikely to be beneficial to the use of small multiples, resulting in smaller graphics that are further apart and thus impacting estimation and comparison tasks. We found that (NBR) variants do not readily imply a good performance for (ALL); VECTOR (ALL) provides a good combination of maintaining directions and distances. At little to no whitespace, we validated and extended Eppstein et al.'s result [13] that DISPLACEMENT yields good overall performance. However, with whitespace, we need more explicit consideration of shape and topology. The main question concerns how to effectively compute such layouts, without losing the benefits of having good DISPLACEMENT. These effects depend on the region-size differences in the input and how the small and large regions are distributed across the map.

## 6 ANALYSIS OF EXISTING LAYOUTS

**London.** We consider two existing layouts of London's boroughs, with different amounts of whitespace: AfterTheFlood's *London Squared* design [1] shows all 33 boroughs, whilst the space-filling grid map used by Wood et al. [38] as a BallotMap contains the 32 boroughs in which local elections occur. Fig. 12 shows their performance, compared to our optimization. The leftmost column shows that the AfterTheFlood solution outperforms the BallotMap in all projection characteristics.[4] However, the BallotMap performs better in HORIZONTAL ALIGNMENT and uses less graphical space. If we consider the other columns, we see how these existing layouts compare to our optimized solutions. The right-hand bar in each cell compares with the BallotMap and reveals that our DISPLACEMENT-optimized layout yields better results for all projection characteristics; there is no difference in array characteristics and SHAPE as there are no gaps. The left-side bar compares with AfterTheFlood. We see that we can do better (blue bars) on all metrics except SPLIT NEIGHBORS (red bar) if we optimize for DISPLACEMENT. If we compare to optimizing SPLIT NEIGHBORS instead, we find that we cannot perform better in that metric, but the optimized result suffers quite strongly in terms of the other measures. We conclude that AfterTheFlood has produced a strong layout in *London Squared*, where SPLIT NEIGHBORS—the need to retain topology between boroughs—has potentially been the predominant design criterion. This is achieved while compromising only a little on the other characteristics; our optimization can do slightly better in many metrics, but at the cost of losing topological features.

**The United States.** We study seven layouts of the US: FiveThirtyEight (538) [8]; Bloomberg [32]; Guardian [26]; NPR [11]; NYT [24]; WP [5]; and Map² Matrix by Guo et al. [16]. Fig. 13 shows their performance (left column) and compares it to our optimization approach. Following the London analysis, we observe many of the same patterns between our optimized solutions and the existing layouts, for both projections: SHAPE and DISPLACEMENT perform better on most metrics, except SPLIT NEIGHBORS. It is worth noting that measuring and optimizing these two metrics are now less related, caused by the cluster of small states in the northeast: as we know, this causes a discrepancy between SHAPE and DISPLACEMENT. Whereas our optimization considers only one of these metrics, the existing layouts seem to make a trade-off between the two. Again, the most important outlier in this pattern is topology. Although our optimized layouts are close in performance to the Map² Matrix layout by Guo et al. [16], their gaps are all on the bottom row allowing them to be used for map-level annotation.

We used two base maps in this analysis: Albers projection with Alaska and Hawaii moved to the bottom-left corner and a latitude-longitude projection. The patterns described above are independent of the projection. However, 538 and WP and the other layouts (except for Map² Matrix) change their relative performance, explained by these layouts placing Alaska (arbitrarily) similarly to the base map in Albers projection. Indeed, the definition of "the right direction" changes between the projections, since we measure directions in projected space.

In spite of defining metrics differently, our findings are in line with those of Wongsuphasawat [37], who concludes that NYT performs best among these layouts (excluding Map² Matrix). However, Bloomberg

---

[4]We adjusted SHAPE to make this a fair comparison: see online material.
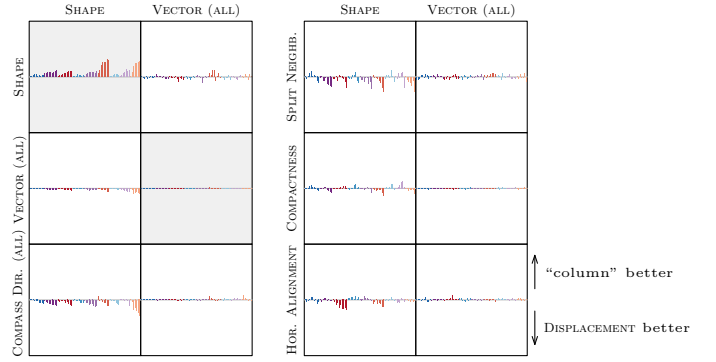


Fig. 11: *Trade-Off Matrix* comparing the performance of DISPLACEMENT to the other metrics.
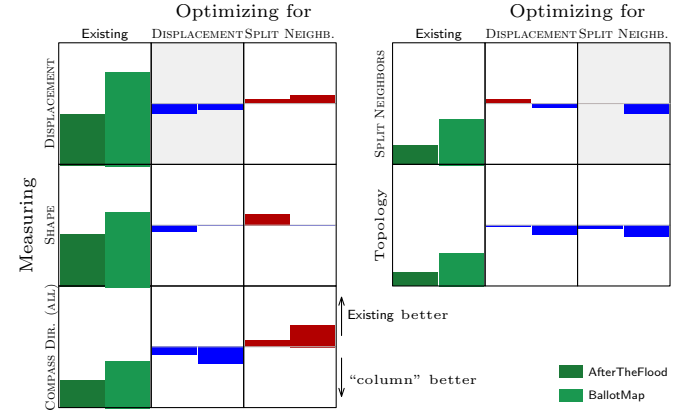


Fig. 12: *Examples Matrix* to compare the absolute performance of existing layouts of London (first column) and the relative performance to our metric-optimized layouts (second and third column).
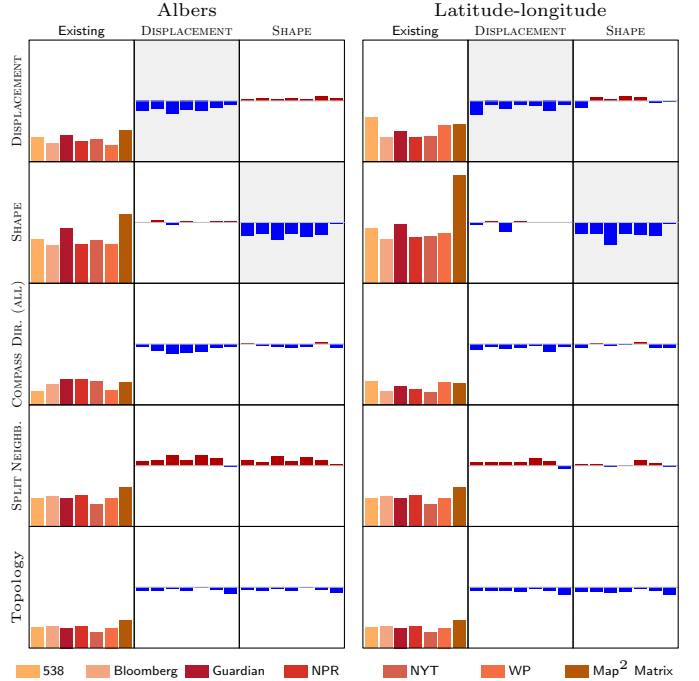


Fig. 13: *Examples Matrix* for the absolute performance (left column) of existing US layouts and their difference to our optimization (other columns). Downward bars (blue) indicate that our optimization is better, upward bars (red) indicate a better existing layout. Two variants of base map are used: one based on Albers projection (left) and one using a latitude-longitude projection (right).

and NPR have a (very) slight edge in terms of SHAPE (a subjective pass/fail test in [37]) and DISPLACEMENT (not factored into [37]).

# 7 FUTURE WORK

We provide the first systematic attempt to study design criteria for spatially ordered small multiples, exploring the role of whitespace. There are many opportunities to extend and build upon our results.

**Metrics.** We use a range of metrics to capture small-multiple layout characteristics, but the list is not exhaustive. For example, there are other ways of quantifying shape, each with different properties. Our experiments allowed us to explore the effect optimizing for different metrics has on the other metrics. Understanding this relationship better may lead to more rigorous algorithmic design to optimize single metrics (or combinations of these) in a more robust and efficient manner than our experimental simulated-annealing approach.

**Effects.** The amount of whitespace affects characteristics of layouts of small multiples in predictable and unpredictable ways. Some of these are desirable and some undesirable but this is likely to depend on the purpose of the graphic. This requires more investigation. We did not study the effect of aligning the array to the map—see Eppstein et al. [13] for algorithmic work—yet this has a strong effect on the metrics and we have only touched upon the impact of map projections. For example, does geographic data in a conical map projection (whose lines of latitude are curved) translate to an orthogonal grid and is it interpretable? We focused on square frames, but other shapes are possible e.g. hexagons, which improve our ability to show topology [11].

**Non-geographic layouts.** With Fig. 3 we argued that small multiples with gaps generalize to non-geographic layouts. By turning scatterplots into small multiples with gaps, other properties of data points can be shown in an aligned, non-occluding manner that facilitates comparison, in trade-off with precision on the scatterplot axes. Some of our metrics are transferable, but others such as SHAPE are not, calling for new metrics. Where there are too many points, one might consider aggregating or binning points based on proximity. This could be applied in a visual analytics context with various abstract projections of high-dimensional data (e.g. [21]), but needs to be investigated further.

**Interpretability.** We focused on quantifying aspects of small multiples with gaps from the perspective of a designer, but there are many open questions on how the various characteristics affect human interpretation. How well can observers compare and detect trends in data with different amounts of whitespace? Do gaps help or hinder to identify large-scale spatial trends? This is likely to depend on the nature of the graphic used within the small multiples, the data distribution and task. The strength of small multiples lies in the structured way of presenting data, but we cannot capture the full extent of the underlying spatial arrangement. We need to understand how these distortions affect interpretation. We could design and evaluate methods of revealing such distortions, which may include annotation, various graphical encodings and animated transitions. These questions call for user studies involving audiences with different expertise and expectations, possibly both through lab studies and crowd-sourced evaluation systems.

**Design through optimization.** We took a metric-based approach to exploring the design space of small multiples with gaps. We first constructed various measures to capture aspects of a small-multiples layout. Subsequently, we optimized and measured these metrics in the context of a wide variety of inputs, as we varied whitespace. We then used visualization to explore the design space. We believe that this approach of metrics, measurements and visual analysis to exploring the design space has been relatively successful here in supporting our attempts at characterizing good small multiples with gaps. The approach has potential for other design spaces where one faces many potentially conflicting criteria and seems worthy of further investigation.

# 8 CONCLUSION

Small-multiple layouts usually reflect ordinal aspects of the conditioning variable; e.g. the temporal sequence of months. We focused on 2D ordinal small-multiple layouts that are conditioned by spatial distributions and ordered by location, but emphasize that this generalizes

to non-geographic spaces (e.g. Fig. 3). We build on existing work to develop practice and to establish the effects of adding gaps to small-multiple layouts to capture aspects of the spatial distribution of the conditioning variable. We have shown that adding gaps to small multiples has beneficial effects, in retaining important characteristics of the original maps such as "shape" which may help readers relate abstract graphics to the more familiar locations that they represent, and in supporting comparison tasks, for example "compactness" which captures the distance between small multiples to facilitate comparisons. But this is at the cost of other characteristics: most importantly the size of each small multiple, but also topological relationships that tend to get more distorted as levels of whitespace increase. These effects are strong and are not always readily predictable.

The design of small multiples with gaps is a complex task: there is no perfect or optimal layout, similar to the myriad of map projections [31] each with its own benefits and drawbacks. Our metrics and exploration of the design space provide reference points to steer good (manual) design and inform future algorithms. The relations between characteristics depend on the nature of the original geometry and the grid onto which is it projected. Our exploration of the effects of region-size variation revealed some complex solutions that perform badly, but size variation alone does not explain this fully. The spatial structure of size variance is a factor as much as size variance itself. We also found that some existing layouts perform well in terms of projection metrics and characteristics important for the comparison of multiples. *AfterTheFlood*, for example, have generated an effective compact layout of the London boroughs, skillfully capturing topology at only marginal costs in terms of other projection characteristics. In line with existing layouts, our analysis suggests that topology is a prominent metric, after which a trade-off between shape and displacement is made—perhaps providing some insights into how designers produced these layouts. We have shown that optima exist (e.g. for shape), but at different levels of whitespace for different maps. Even a few gaps—perhaps less than used in existing layouts—may drastically improve some metrics.

We advocate the approach of quantifying a complex design space through metrics and optimization, using interactive visualization software to explore the effects as we have done throughout this paper. This allowed us to systematically investigate this design space in the face of many related criteria, establishing relations and conflicts between these and gaining insight into a designer's approach to resolving them. It enabled us to generate evidence upon which to hypothesize and make recommendations. Our exploration of the small-multiples-with-gaps design space identified great variation in the effects of adding gaps; each metric is affected differently and no single metric (that we considered) consistently performs well on all fronts, irrespective of the underlying map. This suggests that algorithms for computing layouts may need to take multiple metrics into account, weighing them depending on the spatial aspects of the input data. Moreover, the intent of the graphic may factor into its design, something likely to be beyond the scope of any single algorithm. This challenges the assumption of a single-map solution: solutions may need to be developed or algorithms be applied on a case-by-case basis, selecting appropriate characteristics to inform their design. The legacy of static maps may still pervade our thinking too strongly here [14]. Interactive graphics that transform between layouts to emphasize different aspects may work well, especially if visual encodings are used to convey inconsistencies in the array. This will help us find candidate layouts and, through linking with algorithms, may allow us to approach visual analytics for design.

We call for further work to establish the effectiveness of such dynamic solutions, layouts with different emphases and efforts to represent their deficiencies in small multiples, as we enhance one of Tufte's prominent devices to support those seeking spatial outliers, dependencies and inconsistencies in faceted multivariate information.

## REFERENCES

[1] AfterTheFlood. London squared map. http://aftertheflood.co/projects/london-squared-map. Accessed March 2016.

[2] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 5(1):75–91, 1995.

[3] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–216, 1991.

[4] 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*, in this issue, 2016.

[5] B. Berkowitz and L. Gamio. What you need to know about the measles outbreak. https://www.washingtonpost.com/graphics/health/how-fast-does-measles-spread/, February 2015. Accessed March 2016.

[6] J. Bertin. *Graphics and graphic information processing*. Walter de Gruyter, 1981.

[7] P. Blickle and S. Venohr. Dürfen wir vorstellen: Deutschlands Muslime. http://www.zeit.de/gesellschaft/2015-01/islam-muslime-in-deutschland, January 2015. Accessed March 2016.

[8] B. Casselman and A. McCann. Where your state gets its money. http://fivethirtyeight.com/features/where-your-state-gets-its-money/, April 2015. Accessed March 2016.

[9] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[10] S. Cohen and L. Guibas. The Earth Mover's Distance under transformation sets. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 1076 – 1083, 1999.

[11] D. DeBelius. Let's tessellate: Hexagons for tile grid maps. http://blog.apps.npr.org/2015/05/11/hex-tile-maps.html, May 2015. Accessed March 2016.

[12] F. S. Duarte, F. Sikansi, F. M. Fatore, S. G. Fadel, and F. V. Paulovich. Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2063–2071, 2014.

[13] D. Eppstein, M. van Kreveld, B. Speckmann, and F. Staals. Improved grid map layout by point set matching. In *Proceedings of the IEEE Pacific Visualization Symposium*, pages 25–32. IEEE, 2013.

[14] P. F. Fisher. Is gis hidebound by the legacy of cartography? *The Cartographic Journal*, 35(1):5–9, 1998.

[15] B. Fong. How to make tile grid maps in Tableau. https://public.tableau.com/s/blog/2016/01/how-make-tile-grid-maps-tableau, January 2016. Accessed March 2016.

[16] D. Guo, J. Chen, A. M. MacEachren, and K. Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474, 2006.

[17] M. Harrower and C. A. Brewer. ColorBrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, June 2003.

[18] F. Hausdorff. *Grundzüge der Mengenlehre*. Verlag Von Veit & Comp., Leipzig, Germany, 1914.

[19] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 203–212. ACM, 2010.

[20] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[21] X. Liu, Y. Hu, S. North, and H.-W. Shen. CorrelatedMultiples: Spatially coherent small multiples with constrained multi-dimensional scaling. *Computer Graphics Forum*, 2015.

[22] A. Marzal and V. Palazón. Dynamic time warping of cyclic strings for shape matching. In *Pattern Recognition and Image Analysis*, LNCS 3687, pages 644–652, 2005.

[23] G. J. McRae, W. R. Goodin, and J. H. Seinfeld. Development of a second-generation mathematical model for urban air pollutioni. model formulation. *Atmospheric Environment (1967)*, 16(4):679–696, 1982.

[24] New York Times. How the rulings affect gay couples. http://www.nytimes.com/interactive/2013/06/26/us/scotus-gay-marriage.html, June 2013. Accessed March 2016.

[25] H. Park. Gay marriage state by state: From a few states to the whole nation. http://www.nytimes.com/interactive/2015/03/04/us/gay-marriage-state-by-state.html, March 2015. Accessed March 2016.

[26] K. Powell, R. Harris, and F. Cage. How voter-friendly is your state? http://www.theguardian.com/us-news/ng-interactive/2014/oct/22/-sp-voting-rights-identification-how-friendly-is-your-state, October 2014. Accessed March 2016.

[27] R. Radburn. Go with the flow: Commuting & migration flows within London. https://public.tableau.com/profile/robradburn#!/vizhome/ODMpasLondonAftertheFlood/GowiththeFlow, March 2016. Accessed March 2016.

[28] A. Reimer, A. Unger, W. Meulemans, and D. Dransch. Schematized small multiples for the visual comparison of geospatial data. In *Abstracts of IEEE Information Visualization Conference*, 2011.

[29] A. Slingsby, J. Dykes, and J. Wood. Rectangular hierarchical cartograms for socio-economic data. *Journal of Maps*, pages 330 – 345, 2010.

[30] A. Slingsby, M. Kelly, J. Dykes, et al. OD maps for showing changes in Irish female migration between 1851 and 1911. *Environment and Planning A*, 46(12):2795–2797, 2014.

[31] J. P. Snyder and P. M. Voxland. An album of map projections, 1989.

[32] A. Tribou and K. Collins. This is how fast America changes its mind. http://www.bloomberg.com/graphics/2015-pace-of-social-change/, June 2015. Accessed March 2016.

[33] E. R. Tufte. *The Visual Display of Quantitative Information*, volume 2. Graphics Press Cheshire, CT, 1983.

[34] Wikipedia. Iris flower data set. https://en.wikipedia.org/wiki/Iris_flower_data_set. Accessed March 2016.

[35] N. Willems, W. R. van Hage, G. de Vries, J. H. Janssens, and V. Malaisé. An integrated approach for visual analysis of a multisource moving objects knowledge base. *International Journal of Geographical Information Science*, 24(10):1543–1558, 2010.

[36] K. Wongsuphasawat. Create your own grid map, a semi-automatic way. https://medium.com/kristw/creating-grid-map-for-thailand-397b53a4ecf#.fms4i4ee6, January 2016. Accessed March 2016.

[37] K. Wongsuphasawat. Whose grid map is better? quality metrics for grid map layouts. https://medium.com/kristw/whose-grid-map-is-better-quality-metrics-for-grid-map-layouts-e3d6075d9e80#.rt72gpv5v, January 2016. Accessed March 2016.

[38] J. Wood, D. Badawood, J. Dykes, and A. Slingsby. BallotMaps: Detecting name bias in alphabetically ordered ballot papers. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2384–2391, 2011.

[39] J. Wood and J. Dykes. Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1348–1355, 2008.

[40] J. Wood, J. Dykes, and A. Slingsby. Visualisation of origins, destinations and flows with OD maps. *The Cartographic Journal*, 47(2):117–129, 2010.

[41] J. Wood, A. Slingsby, and J. Dykes. Visualizing the dynamics of London's bicycle-hire scheme. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 46(4):239–251, 2011.